A Model for Inebriation Recognition in Humans Using Computer Vision

by

Zibusiso Bhango

DISSERTATION

submitted in fulfilment of the requirements for the degree

MASTER OF SCIENCE

in

Information Technology

in the

FACULTY OF SCIENCE

at the

UNIVERSITY OF JOHANNESBURG

SUPERVISOR: Prof. D. T. Van der Haar

November 2020

Abstract

Inebriation is a situational impairment caused by the consumption of alcohol affecting the consumer's interaction with the environment around them. Inebriation leads to lower inhibitions, loss of fine motor coordination and the inability to control a motor vehicle. Driving while inebriated is a criminal offence in many countries involving driving a motor vehicle after consuming alcohol to a level that renders the driver incapable of safely operating a motor vehicle.

Short term effects of inebriation include weakened balance, slow reaction times, lack of fine motor coordination, slow pupil response and slow heart rate. Alcohol consumption also results in slurred speech, balance, slow reaction times, pupil dilation, double vision, heavy sweating, slowed heart rate and reduced blood pressure can also result from drinking alcohol.

These short-term effects can be used to recognise inebriation in humans. Many systems have been used for inebriation recognition. These include using biomarkers such as blood tests, urine tests or hair tests. Behavioural traits emanating from inebriation can also be used to recognise inebriation. Heart-based signals have been used because the heart rate slows down when inebriated. Thermal imaging can be used to recognise inebriation because inebriated people have more active blood vessels. Gait can be used to recognise inebriation because of the loss of balance that results from inebriation. Although these methods have been used to recognise inebriation with varying performance, they have their shortcomings, such as being too invasive (heart-based), requiring expensive equipment (thermal imaging), being too inconvenient (biomarkers) and not yielding enough inter-class and intra-class variability (gait).

An ideal inebriation recognition method operates in real-time, is less intrusive, more convenient, and efficient. Computer vision has been used to solve object recognition, image segmentation, object recognition, object classification, object tracking, along with context and scene understanding problems. In this research, we propose a model for inebriation recognition in humans using computer vision.

In our study, we first described the research methodology that we adhered to. We used the design-oriented research science methodology because of its ease of use and compatibility with

the study of IT artefacts. We performed a literature review to understand our environment and inebriation problem, which is inebriation, our potential solution computer vision and investigated how similar work in computer vision and inebriation recognition is implemented in literature.

We then created our model using the knowledge from our literature review to choose pipelines and methods for our model. We then created a benchmark to quantify our research against. We developed both functional and non-functional requirements for our study that meets our research aim. We created a dataset using publicly available face images of sober and inebriated individuals on the Internet. We made sure our dataset consisted of images that are free to use and have enough features to objectively sample our model and quantify our results.

We then developed our research prototype based on our model. Our prototype consisted of various pipelines containing both traditional and deep learning methods to recognise inebriation in humans. After our prototype is implemented, we critically analysed our pipelines statistically using our benchmark. We then compared the performance of our pipelines and picked the best performing pipeline. Our best performing pipeline was YOLOv5, which exhibited state-of-the-art results with a 97.5% classification accuracy.

Acknowledgements

Special thanks to my wife Ngalula Patricia Bhango for being my rock, and Prof. D.T. van der Haar for all the support, guidance, and patience. I would also like to thank Mr Khulekani Mathe for making all of this possible…

Writing Style

In the dissertation, the use of the pronoun "we", may refer to the author, reader, and/or any collective group, depending on the context in which it resides. We use this royal "we" to reduce the subjective tone and to increase readability throughout the dissertation, thereby improving the reader's experience.

Table of Contents

List of Figures

List of Tables

List of Abbreviation

ANN	Artificial Neural Networks
BAC	Blood-Alcohol Content
CLAHE	Contrast-Limited Adaptive Histogram Equalisation
CNN	Convolutional Neural Networks
CV	Computer Vision
DUI	Driving Under the Influence
DSR	Design Science Research
ECG	electrocardiogram
EtG	Ethyl glucuronide test
GBT	Gradient Boosted Trees
HRV	Heart Rate Variation
ML	Machine Learning
ICA	Independent Component Analysis
IS	Information Systems
LDA	Linear Discriminant Analysis
LoG	Laplacian of Gaussian
PCA	Principal Component Analysis
PEth	Phasphatidylethanol
PPG	photoplethysmography
ROI	Region of Interest
SVMs	Support Vector Machines
YOLO	You Only Look Once

UNIVERSITY
OF
JOHANNESBURG

Chapter 1 Introduction

1.1 Introduction

Excessive consumption of alcohol leads to inebriation. After consuming alcohol, physical and physiological changes begin to take place. Inhibition, emotional ability, and self-awareness are affected [1]. Driving while inebriated is a criminal offence in many countries that involve driving a motor vehicle while inebriated beyond the legal limit. Every 33 minutes, a person in the world is dying in a road accident caused by inebriated driving [2].

Excessive alcohol consumption introduces long-term impacts on our health. Heavy alcohol consumption meddles with the delicate balance of neurotransmitters. Short-term effects of alcohol consumption can be used to recognise inebriation. Short-term effects of alcohol consumption include lower inhibitions and caution, loss of fine motor coordination and inability to perform critical hand-coordinated tasks such as controlling a motor vehicle [3]. Existing inebriation recognition methods include using biomarkers such as a breath test, urine test or hair test. These methods are inconvenient and invasive, introducing privacy, ethical and legal issues.

An ideal inebriation recognition method in a real-time environment is fast, convenient, and non-invasive. In this research study, we propose a model for inebriation recognition using computer vision.

In this chapter, we will introduce our research. We begin by introducing our research problem in section 2. In section 3 we will discuss our research hypothesis. In section 4 we will provide our research aims and objectives that we intend to achieve. In section 5 we will briefly mention our research methodology, which is the blueprint of our research. In section 6 and 7 we will discuss our research constraints and assumptions, respectively. We will then briefly discuss the research articles we have written during our study and have been published in academic journals and the role they played in our research in section 8. We will close off our introduction chapter by providing our research roadmap in section 9.

1.2 Research Problem

Substance abuse is a social issue that has plagued societies for centuries. Alcohol abuse is a social issue that has resulted in fatal vehicle accidents and long-term health effects.

Depending on the type of alcohol and its effectiveness, it is often difficult to identify drinkers, especially in their early stages. Physical signs of detecting alcohol abuse include rapid heart rate, high blood pressure, poor muscle coordination, and dilated pupils. However, with excessive drinking, many behavioural traits become more apparent, such as depression, introversion, and poor personal hygiene.

Substances intercept and alter the messages going to the nervous system, resulting in altered perception. Usually, alcohol induces euphoria, relaxation, or hyperventilation, thereby changing the mood of the drinker considerably. The euphoria is the feeling the user is after, and the user will continue consuming alcohol to keep getting the same "high". However, tolerance builds up swiftly; increased doses are required to satisfy the same level of effects, leading to dependence. When unattended, this can lead to fatal alcohol poisoning.

More Americans die from substance overdose than they do in car accidents [4]. To tackle this social issue, there is a need for novel methods to gain more insight and combat substance abuse and addiction. The most famous way of recognising alcohol abuse is using a breathalyzer. This method, although useful, is quite invasive and inconvenient. Different methods of inebriation recognition are required which are non-invasive, fast, and convenient.

The side effects of alcohol consumption make it possible to detect inebriation in an image or video. In this research, we propose a model that uses computer vision for inebriation recognition in an individual. This system will be used to detect alcohol abuse in a non-invasive manner. We will discuss our research hypothesis in the next section.

1.3 Hypothesis

Our research hypothesis is that computer vision can be used to detect inebriation in humans. An image or video of an individual provides enough information about the individual to detect if

they are sober or inebriated. A model can thus be developed that can use images or videos to classify individuals as either inebriated or sober.

For our research outcome, we are expecting a model that can recognise inebriation in humans. The model will use computer vision to classify images and videos of individuals as either inebriated or sober. Classification algorithms will be used to recognise different features in images and videos of inebriated and sober individuals.

We expect the model to be useful in societies by being able to recognise inebriation in individuals who are struggling with alcohol addiction and provide help before it is too late. The model can also be used to reduce inebriated driving by capturing images and videos of drivers, thereby reducing fatalities and injuries on the road caused by drivers driving while inebriated.

1.4 Aims and Objectives

Our research aim is to build a model that recognises inebriation in humans using computer vision by detecting two classes of people – inebriated individuals and sober individuals. Our model will comprise of various pipelines and we will compare the pipelines' performances to find the pipeline with the best results.

Computer vision will be used to detect whether the individual is inebriated or sober. Alcohol consumption increases the heart rate, lowers inhibition, slurs speech, lowers reasoning ability, weakens balance, and slows reaction times. Most of these symptoms become apparent in a video or image and can potentially be used to detect inebriation in an individual.

Our research objectives are listed below:

1. Do a literature review on inebriation, computer vision and similar methods for inebriation recognition using computer vision.
2. Create a model to recognise inebriation in humans using computer vision.
3. Create a benchmark to measure our model's performance against.
4. Create a dataset using publicly available face images of inebriated and sober individuals on the Internet.

5. Create our research prototype, provide research results by statistically analysing the prototype against the benchmark and compare different pipelines' results to find the best performing pipeline for our research.

Potential users for our research are public workers who work in environments where inebriation can potentially harm people, such as truck drivers. Our research can also be used to monitor alcohol consumption at bars, with the bartender monitoring inebriation levels before selling more alcoholic beverages to individuals. Our research can also be used by traffic officers instead of breathalyzers to detect inebriation in drivers.

1.5 Constraints

Our study aims to recognise inebriation in humans using computer vision. However, it is not meant to be used alone to recognise inebriation. It must be used to recognise potential inebriation in humans, with more in-depth measures such as blood tests taken to ascertain inebriation on individuals that are recognised as inebriated by our system.

Our research aims to recognise inebriation only in humans. Our methods will be designed to classify inebriation or sobriety as a binary classification problem. Our approach is based on computer vision methods. We use human face images or videos to recognise inebriation. These human faces are of either sober or inebriated people and are randomly chosen.

Using our system in these environments comes with certain ethical and legal issues. These issues are addressed in the research methodology chapter. Before working on our research, we obtained ethical clearance approval during our research proposal phase.

1.6 Assumptions

Our research uses a dataset with subjects that were not scientifically proven to be inebriated and the ground truth labels depend on metadata of the image sources. The dataset was collated from publicly available images of sober and inebriated individuals on the Internet. The level of inebriation or sobriety cannot be proven, however, the face images we classified as sober are

assumed to remain sober throughout the research. Also, the images that are classified as inebriated are assumed to remain inebriated throughout our research.

1.7 Research Methodology

We used the design science research method instead of the scientific research method for our research paper. We chose the design science approach because of its ease of use and compatibility with developing information systems. We believe scientific research works well with objects or social phenomena and design science is best suited for engineering and IT artefacts. Design science research is rooted in engineering and has been generally accepted by IT practitioners as an important part of information systems research since information system's inception [5]. We discuss our research methodology approach in detail in chapter 2.

1.8 Published Work (Contributions)

During our research, we published a research paper based on our literature review, our model, and the results we gathered. We also presented our research findings at the Business Information Systems 2019 conference in Sevilla, Spain in 2019. The research paper is titled "A Model for Inebriation Recognition in Humans Using Computer Vision." In this research paper, we discussed inebriation, computer vision and similar systems in literature using computer vision and inebriation recognition. We discussed our model and developed a prototype using computer vision methods. After statistically analysing our prototype, we provided our very good model results.

Publishing this research paper and getting the model and results feedback gave us pivotal help in the research. Please find below the details about the research paper publication, which is also included in appendix A:

Research paper title: A Model for Inebriation Recognition in Humans Using Computer Vision.

Conference: 22nd International Conference on Business Information Systems

Publication Name: Lecture Notes in Business Information Systems.

Publisher Name: Springer

ISSN: 978-3-030-20484-6

We have also written another research paper titled “A Comparison of Deep Learning Methods for Inebriation Recognition in Humans” which has been submitted for the 33rd International Conference on Advanced Information Systems Engineering. The research paper focuses on comparisons of results gathered from various pipelines we implemented in our model for inebriation recognition in humans using computer vision. The research paper is included in appendix B.

1.9 Outline of the Dissertation

Our research roadmap aims to meet our research aims and objectives. We will start with the research methodology to provide a blueprint for our research and how we aim to achieve scientific robustness in our research. We will then discuss our environment and research problem in the inebriation chapter.

We then discuss our solution domain, which is computer vision, and provide similar systems in literature that use computer vision and inebriation recognition. Our similar systems chapters make up our literature review phase of the research. After the literature review, we will discuss our research model in detail.

A research benchmark is then created based on both the functional and non-functional requirements of our research. A prototype based on the model is developed, including various diverse pipelines that aim to solve the research problem. After developing our prototype, we will get our research results by statistically analysing the prototype using our benchmark requirements. We will then compare our pipelines’ results, provide our findings, and choose our best performing pipeline for our research problem. A fitting conclusion is then provided to close off our research.

1.10 Conclusion

In this chapter we introduced our research. We introduced our research problem, discussed our research hypothesis, aims and objectives, research methodology, constraints, assumptions, published work and provided a roadmap for our research.

In the next chapter, we will discuss our research methodology and provide reasons why we chose that specific research methodology. We will also discuss the ethical and legal implications of our research and how we address both issues.

Chapter 2 Research Methodology

2.1 Introduction

Research design is the conceptual blueprint used for conducting research [6]. It is used to formulate research problems and objectives and present results gained from the study. Research design focuses on the research process and the tools and procedures used. Choosing a research approach is crucial because it determines how relevant information for a study will be obtained [7].

In this chapter we will objectively discuss two research methodologies in detail: scientific research and design science. In section 2, we will discuss the scientific research methodology and the components that make up the methodology, namely research design, research paradigm and research methods. In section 3 we will discuss the design science methodology in detail, mentioning two types of design science: design-oriented and design-based. In section 4 we make our methodology selection and provide a justification. We close off our chapter with a brief look on how we handle our research's validity and reliability, the ethical issues and risks introduced by our research.

2.2 Scientific Research Methodology

The scientific research methodology is a common research methodology in academia. It is made up of three pillars, namely research design, research paradigm and research methods.

The research design provides an appropriate framework for research [7]. It is the glue that keeps research elements intact [6]. Research design is a masterplan specifying the methods and procedures for collecting and analysing information to help answer a research question [6]. It is driven by the research problem and focuses on the logic of the research [8]. A research design contains a clear research problem, procedures and techniques for information-gathering, the studied population and the methods used for data processing and analysis [6]. A research conducted haphazardly without a robust research design may draw up incorrect conclusions.

There are various research design approaches used in research. We discuss the most popular ones in the below sections.

2.2.1. Qualitative

Qualitative research is a systematic scientific enquiry that builds a holistic description to inform the researcher's understanding of a social or cultural phenomenon [9]. It is used by researchers to study behaviour, opinions, themes, and motivations by exploring depth, richness, and complexity inherent in the phenomenon [10]. Qualitative methods are used to understand how participants understand the reality of a situation and interact with it from a subjective point of view [11]. Qualitative research design is flexible and is made up of generally accepted methods and structures [9].

Qualitative research yields richer and more insightful information about a phenomenon. In qualitative research, the researcher collects the data, examines why events occur and what these events mean to the studied subjects [12]. Primary data sources used in qualitative research include field observation, interviews, informal discussions, focus groups and case studies [9]. Qualitative research does not depend on sample sizes [10].

Qualitative research has proven to be time consuming and costly, requiring the researcher to handle hour-long interviews, among other things. Qualitative data gathered cannot be quantified mathematically due to intentional sampling [11]. Researchers also need to be aware of the ethical issues, bias and philosophical underpinnings of their research questions [10]. Qualitative data is prone to personal bias and judgment, and results cannot be re-examined or generalised [11]. The common qualitative research design types include biography, phenomenology, grounded theory, ethnography and case study [13].

2.2.2. Quantitative

Quantitative research design is a formal, objective, systematic process to obtain information about the world. It is a common research design method used in most scientific disciplines [14]. Quantitative research is used to construct a generic principle to explain a phenomenon in a

certain situation aiming to extrapolate the possible consequences of various diverse situations [11].

Quantitative research requires developing a hypothesis that must be proved or disproved [14]. This hypothesis must be mathematically and statistically proven and is the pillar on which the research is designed and conducted. Mathematical and statistical analysis is used on randomly chosen research data to evaluate the hypothesis [11]. It is encouraged to only manipulate a single variable in research to avoid complicating statistical analysis [14]. A qualitative research must be conducted in a way that other researchers can recreate the experiment and get similar results.

Quantitative data is data that can be statistically analysed, revealing relationships between variables and is used to generalize concepts more widely and predict future results. The goal of quantitative research is determining the relationship between an independent variable and a dependent variable within a population. After data analysis, the results can be used to answer a hypothesis that can be legitimately discussed and published.

2.2.3. Mixed Methods

Historically, there is antagonism between qualitative and quantitative research [11]. Mixed methods research is made up of both qualitative and quantitative research approaches for a deeper understanding and corroboration [15]. It is used to strengthen a study's conclusions while increase knowledge and validity.

Mixed methods research design becomes necessary when neither qualitative nor quantitative research design can fully address the research. It incorporates a qualitative component into a quantitative study or vice versa. It can also be used to build from one phase to another or explore a problem qualitatively before building an artefact and quantitatively analysing it. Determining the point of integration is an important part of the research. Data collection can either be done in parallel or sequential phases.

Mixed methods research design uses interviews and questionnaires, performance tests and observations. The common mixed methods research design forms are the convergent design, the explanatory design, the exploratory design and the confirmatory design.

The research paradigm and its methods are discussed in the sections below.

2.2.4. Positivism

A research paradigm provides the process to execute research [16]. It is made up of assumptions and beliefs in a research community about ontological, epistemological, and methodological concerns [17]. There are several research paradigm methods used in research. We briefly discuss the most popular ones in these sections.

Positivism assumes that reality exists independently of humans [18]. It believes the same cause-effect relationship in nature exists in the social world [19]. Due to the reality being context-free, different researchers working independently of each other will converge to the same conclusions about a given phenomenon.

Evidence is gathered and critically analysed to explain causal effect between the dependent variable and the independent variable [18]. A hypothesis is proposed and either confirmed or rejected based on the results of statistical analysis on the data.

2.2.5. Interpretivism

Interpretivism rejects that a single reality exists independent of our senses, believing multiple socially constructed realities exist [18]. It argues that reality is created, not discovered, and cannot be known as it is governed by our senses. Due to different worldviews, well-argued varying interpretations about a phenomenon are accepted [18]. Interpretivism research's goal is to understand how subjects interpret the phenomena they interact with.

In interpretivism methodology, social phenomenon is understood through the subject's point of view as opposed to the researcher [18]. Qualitative data such as audio or video is collected from participants over a lengthy period and analysed to extract hidden patterns and generate a theory. Differing subjects' viewpoints provide researchers with a deeper understanding of phenomena in social context [20].

2.2.6. Pragmatism

Pragmatism was birthed from the argument that a single scientific method cannot be used to exclusively know the world and its realities [19]. Pragmatism argues that more practical approaches are necessary to learn the truths about the world. Pragmatism deals with facts [16]. It uses mixed methods from positivism and interpretivism as a pragmatic way to understand human behaviour [19].

Research methods are discussed in the below sections.

2.2.7. Literature Review

Research methods are the tools used by researchers to do research. These tools can be qualitative, quantitative, or mixed. Research methods include literature review, model, prototype, data sampling, data analysis and population. These methods are described in the sections below.

Literature review is a comprehensive overview of all available knowledge on a specific topic to date. It looks at scholarly articles, books and scientific research and experiments relevant to the area of research being conducted. Literature review aims to enumerate, summarize, objectively evaluate, and provide clarity to the research and provide a theoretical base for the research.

Literature review looks at previous researchers' work. It provides a full understanding of the developments that have taken place in the specific field, informing the reader of the author's diligence in studying and understanding the significant works in the field of research.

A literature review must provide context for the research and justify the research. It must identify important works and scholars in the field, highlight flaws and gaps in previous research, acknowledge existing theories, illustrate points of view and misconceptions in the field of research. A thorough literature review can be used to ascertain the research has not been done before, educate the researcher on the subject and gaps in literature. Literature review helps refine, refocus, or evolve a research field.

2.2.8. Model

A model is a physical, mathematical, or conceptual representation of a system of ideas, events, or processes. It is used for presenting a hypothesis. It is central for building knowledge and demonstrating the tentativeness of scientific knowledge. Testing models can lead to redesigned models and improved predictions and experiment output.

2.2.9. Prototype

A prototype is a functional unit used to evaluate the design, performance, and production potential of a phenomena. It is derived directly from the research model. A model provides theoretical methods and/or processes that can be used to prove or disprove a research hypothesis. A prototype is a representation of a design idea that can be in any form [21]. The statistical analysis is then used to prove or disprove the research hypothesis.

Building a prototype and evaluating it is an iterative process. A researcher develops a model, builds a prototype based on the model and evaluates the prototype using statistical analysis and metrics. A researcher can make changes on the prototype to yield better results. This iterative process can go on until the researcher has found the best results possible. These results are then used to statistically analyse the model performance and prove/disprove the research hypothesis.

We will discuss design science in the below sections.

2.3 Design Science

Design science research (DSR) is an information systems (IS) paradigm that has been generally accepted by IT practitioners [5]. Information systems is a discipline that uses information technology-related artefacts in human-machine systems. Design science caters for the building and evaluation of artefacts [22]. Design science research creates knowledge through using design, analysis, reflection, and abstraction. Design science is knowing through building [23].

There are different forms of design science research, such as the Peffers Approach, the design-oriented and the design-based IS research. A general design science research comprises of four

consecutive phases, namely analysis, design, implementation, and dissemination [24]. In the sections below we will discuss the common types of the design science methodology.

2.3.1. Design-Oriented Information Systems Research

Design-oriented information system (IS) research is an alternative way to conduct design science research. It aims to develop and provide an artefact as a research contribution or output [24]. This artefact should address a real-world problem having various stakeholders. Design-oriented IS research values stakeholders because of the essential role they play in the creation of the artefact. The artefact can be guidelines, frameworks, business models or methods [24].

Research must comply with the four principles to be classified as a design-oriented approach: abstraction, originality, justification and benefit [24]. For an artefact to meet the abstraction principle, it must be generalised enough to solve a certain domain and not focussed on one single solution. To meet the originality principle, the artefact must be novel to cater for gaps in literature. To meet the justification rule, the building of an artefact must be justifiable and must be able to be validated. An artefact must be beneficial to its stakeholders. These four principles provide a basis on which design-oriented IS research is built.

Design-oriented IS research is a proponent for academic freedom, offering researchers freedom to decide on research objectives and methods. This academic freedom is supported if the research abides by the four principles mentioned above [24]. Design-oriented IS research follows an iterative research process consisting of four consecutive phases grounded on the principles. These phases do not prescribe, dictate, or propose comprehensive guidance to be followed but instead allow for academic freedom to be practiced [24]. The four phases are discussed in the next section.

a. Analysis

During analysis, the research problem is identified. The artefact is developed based on the research problem to provide a solution [25]. The proposed solution must be justified highlighting the knowledge of the problem and how the solution solves the problem.

After identifying the research problem, the research objectives are formulated. The research objectives, either qualitative or quantitative, are extracted from the research [25]. Quantitative objectives compare the working solution against existing ones while qualitative objectives are more focused on developing an artefact that can provide solutions to previously unaddressed problems [25].

b. Initial Design

At this phase, the artefact is designed according to the generally accepted methods. The chosen design of the artefact must be justified by existing solutions. Design in this context is both a process and an artefact [22]. Design describes the world as acted upon (processes) and the world as sensed (artefacts).

c. Evaluate

During the evaluate phase, the artefact is produced against the objectives specified in the analysis phase. Continuous refinement of the artefact is implemented until an acceptable artefact is produced. The evaluation of the artefact is a continuous process throughout its development. Evaluating an artefact provides an opportunity for feedback, which can be used to build a more efficient artefact and process. Building and evaluating an artefact is an iterative process that aims to improve the artefact. Due to the continuous evaluation, this approach is best suited for an environment where the artefact must be developed according to stakeholders' specifications.

d. Validate & Diffuse

Validation of the artefact takes place at this phase. During this phase, the artefact is observed and measured against the research problem. Research objectives are compared against the built artefact. The validation process requires the use of metrics for analysing data [25]. The methods of validating the artefact will vary depending on the artefact. After the validation process, the artefact is then released to the stakeholders.

2.3.2. Design-Based Information Systems Research

Design-based research are approaches that foster new theories, artefacts, and practices. The design-based research approach includes comprehensive guidance on how to conduct design science research in the form of elements that should be completed within each phase. The four phases of the design-based research include comprehensive guidance on how to complete the individual phase. These elements act as guidance and must be completed to complete the research. The elements are discussed in the four phases of this method below.

a. Analysis

In the analysis phase, researchers and stakeholders collaborate to analyse practical problems. The research problem is designed at this phase in collaboration with researchers and stakeholders. The research objectives and literature review elements are executed [26].

b. Development

In this phase, the solution is designed and developed based on the existing core aspects and technological innovations [26]. A theoretical framework, development of draft core aspects to guide the design of the intervention and the description of the proposed intervention are the elements that need to be completed at this phase.

c. Test & Refine

In this phase, the initial artefact is developed according to stakeholders' specifications. Data is collected and analysed to evaluate the performance of the artefact. The artefact is refined based on the feedback until the final artefact is produced. The artefact is refined amid iterative cycles of testing.

d. Reflect

In this phase, a reflection is performed to produce core aspects on which the artefact is based on and enhance the solution implementation. Design principles, a designed artefact and professional development are the elements that need to be completed at this stage [26].

2.3.3. Guidelines-Based Information Systems Research

The guidelines-based research is a problem-solving process built on seven guidelines [22]. These guidelines are design as an artefact, problem relevance, design evaluation, research contributions, research rigor, design as a search process and communication of research. Below is a table describing what each guideline represents:

Guideline	Description
Guideline 1: Design as an Artifact	Design-science research must produce a viable artifact in the form of a construct, a model, a method, or an instantiation.
Guideline 2: Problem Relevance	The objective of design-science research is to develop technology-based solutions to important and relevant business problems.
Guideline 3: Design Evaluation	The utility, quality, and efficacy of a design artifact must be rigorously demonstrated via well-executed evaluation methods.
Guideline 4: Research Contributions	Effective design-science research must provide clear and verifiable contributions in the areas of the design artifact, design foundations, and/or design methodologies.
Guideline 5: Research Rigor	Design-science research relies upon the application of rigorous methods in both the construction and evaluation of the design artifact.
Guideline 6: Design as a Search Process	The search for an effective artifact requires utilizing available means to reach desired ends while satisfying laws in the problem environment.
Guideline 7: Communication of Research	Design-science research must be presented effectively both to technology-oriented as well as management-oriented audiences.

Figure 1: Design-Science Research Guidelines [22].

The guidelines-based research approach is based on the principle that understanding the design problem and the solution to the design problem is done while building the artefact [22].

In the next section, we will provide our research methodology and a justification for why we chose it.

2.4 Methodology Selection

Our research uses the design science research instead of the scientific research method for our research study. Our research implements the design-oriented design science approach because of its ease of use and compatibility with developing information systems. Design science is knowing through building [23]. We believe scientific research works well with objects or social phenomena and design science is best suited for engineering and IT artefacts. Design science research is rooted in engineering and has been generally accepted by IT practitioners as an important part of IS research since IS's inception [5].

We chose the design-oriented information system research over the design-based information system research because the design-oriented approach aims to develop and provide an artefact as a research contribution or output [24]. In contrast, the design-based research intends to create new theories [27]. The design-based research emanates from the learning sciences, and not information systems. We believe our research meets the four design-oriented IS research principles: abstraction, originality, justification and benefit. The design-oriented approach also allows for "academic freedom", allowing us the freedom to select the most appropriate methods at hand [24].

The design-oriented method is made up of four phases: analysis, initial design, evaluate and validate & diffuse. During analysis, the problem is identified, objectives are formulated and used to develop an artefact to produce a solution [25]. In our research, we will use the quantitative research design because our research is best suited for numerical data that can be statistically analysed to prove or disprove the research question and objectives.

During initial design, our research is designed according to generally accepted methods, resulting in a research model. In the evaluate phase, a prototype is developed based on the model against the objectives specified in the analysis phase. Continuous refinement of the prototype takes place until our prototype shows acceptable results. During validate & diffuse, the prototype is observed, and its results are measured against our research problem and objectives.

For data sampling, we used a primary dataset. We collated face images of sober and inebriated individuals from publicly available images on the Internet. Since we are not using a scientifically

proven dataset, we are testing our model in the wild. We selected face images with a large distinction between sober cases and inebriated cases. Our dataset is made up of different types of people, from sex, race and age. Our population is made up of sober and inebriated humans.

In the next section we will briefly discuss population and data sampling for our research.

2.5 Data Sampling and Population

Data sampling is a technique that uses a subset of data to infer information about a specific population. Data sampling eliminates the need to investigate every individual. To maintain data quality, the sampled data must be balanced and representative of all population.

Research problems speak to a specific group of people called a research population. A research population shares specific traits or behaviour, and research is done for the population's benefit. Due to large sizes of populations, researchers rely on sampling to avoid testing every individual.

There are two ways to obtain data for sampling: using primary data and secondary data. Primary data is data created as part of the research while secondary data is existing data produced by others. In our research, we created a primary dataset and used it to evaluate our model. The quantity, subjects and population of our data sampling is discussed in below.

For data sampling, we could not find a dataset in literature for inebriated and sober individuals, therefore, in our research we are testing in the wild. We tried to the best of our abilities to collate images that can be classified as sober or inebriated. Also, there are other issues such as lighting, change in sensor and change in environment that might affect our model. We created a primary data set consisting of 230 images made up of randomly selected people from different walks of life. Figure 2 shows a sample of our dataset. Some of the images have more than one face image as shown in Figure 3.

Figure 2: A sample of our data set. The top row consists of inebriated individuals and the bottom row consists of sober individuals.

Figure 3: An image consisting of more than one face image.

Our dataset offers diversity in terms of race, age, and gender. It also consists of images captured in various image resolutions, poses, backgrounds and lightings. We collated our data from publicly accessible images on the internet with free usage rights. After collating our dataset, we

performed data augmentation. For each image, we obtained three more images by rotating the image to both the left and the right within the 30-degree range and flipping the original image horizontally as shown in Figure 4.

Figure 4: An augmented sample of a sober individual. Far left is the original image. Middle left is the horizontal flip of the original image. Middle right is the right rotation of the original image and far right is the left rotation of the original image.

We performed data augmentation to increase our dataset size and to help our model detect objects even when the camera or subject are not perfectly aligned. This makes our model more robust. After performing data augmentation on our 230 face images, we get 920 images. From the 920 images, we get 1000 face images. 392 face images are of sober individuals and 608 face images are of inebriated individuals. This means we have a relatively imbalanced dataset. 726 images were used for model training and validation whilst 184 images were used to test our model. From the 726 training images, 799 face images (308 sober and 491 inebriated) were used. From the 184 test images, 201 face images (84 sober and 117 inebriated) were used.

By creating our dataset, we have met our research objective 4. In the next sections, we will discuss the validity and reliability of our research and the risks and ethical issues introduced by our research. We also provide an explanation on how our research attempts to mitigate/minimise these risks and ethical issues.

2.6 Validity and Reliability

In our research, we have put measures in place to ascertain validity and reliability. Our research follows the design science research methodology to maintain scientific validity and reliability. We come up with our research question and objectives during the analysis phase, the research model in the initial design phase, build our prototype in the evaluate phase and observe results and measure them against our research problem and objectives in the validate & diffuse phase. By implementing this process, we are making sure our research follows a tried and tested scientific process, and its output is both valid and reliable.

Our research provides sufficient details for other researchers to replicate our experiments and yield similar results. We provide in-depth literature review to provide a background to our research, similar systems in literature to our research and the gap in literature our research covers. We provide a research model with methods and algorithms that can be used to potentially solve our research problem. We also provide a prototype with different pipelines showcasing different ways to solve the research problem and compare the different outputs using statistical tools.

Our sampled data is randomly chosen and publicly collected for variability. It is made up of individuals from diverse race, sex, and age for model generalisation. Our dataset is made up of face images of individuals in various situations such as bars, roads, home, and workplaces, providing a unique problem of recognising inebriation in various scenarios, leading to better generalisation and validity. We use a relatively large dataset to avoid bias. We also perform data augmentation on our dataset to provide more robustness. A uniform face image size is used as input for our methods. Our research uses methods that have been tried and tested in literature with great results.

2.7 Ethical Considerations

As a human biometrics system, our research has ethically contentious connotations. Privacy sits at the heart of ethical issues in biometrics [28]. Biometric information is collected through observing individuals. These individuals might not be aware they are being observed. Although our research does not involve live capture of human biometric traits, our system can be extended

by organisations or governments to observe individuals without their consent or knowledge. The observance of individuals, whether they are aware or not, takes away their privacy.

Our research stores and processes sensitive information about humans. The knowledge of whether an individual is inebriated or sober is very sensitive and might be damaging if it falls in the wrong hands. Extra care will need to be practised to keep human sensitive information private. We believe our research can save lives by proactively recognising inebriation in humans before they execute tasks that might lead to injury or death.

We believe the biometric trait we are using (a human face) is public knowledge and highly acceptable in the public domain. Many individuals take profile images of themselves and post them on social media for public viewing. However, we believe using primary data for research purposes also comes with its own ethical issues as inebriated people cannot give ethically acceptable consent. In our research, we use a dataset collated using publicly available face images on the Internet, therefore, our dataset does not take away human privacy. Our dataset consists of publicly available images that are free to use. In this way, we do not infringe any copyright laws.

2.8 Risks

There is no biometric system that yields 100% accurate results. Every biometric system comes with errors such as false identification or failure to identify a correct biometric trait. It is discouraged to operate a motor vehicle while inebriated, and it is a crime to do so in many countries. If our research is used to flag inebriated drivers, there are legal consequences for drivers recognised as inebriated.

The main risk for our research is when a sober individual is recognised as inebriated, and when an inebriated individual is recognised as sober. A sober individual recognised as inebriated will face criminal charges unfairly, and this might have a negative impact on their life and livelihood. An inebriated individual recognised as sober might cause a vehicular accident, losing their lives and taking innocent victims with them.

We believe it is very important to ensure our research performs very accurately in recognising both sober and inebriated individuals before it is deployed into an organisation or government system. Also, we believe our system is best suited to recognise inebriation, but further tests such as blood or urine tests must be taken to ascertain inebriation for individuals that were recognised as inebriated by the system.

2.9 Conclusion

Research design is a conceptual blueprint for conducting research, providing guidance from problem and objectives formulation to analysis and presentation of research results [6]. Research methodology focuses on the research processes, tools and procedures used to conduct research.

There are two common forms of research methodology: scientific research and design science. Scientific research methodology is one of the most used method in academia. It consists of the research design (quantitative, qualitative, and mixed methods), research paradigm (positivism, interpretivism, critical theory and pragmatism) and research methods (literature review, model, prototype, data sampling and population). Design science is a generally accepted research method by information systems practitioners [5]. It is rooted in engineering and is fundamentally a problem-solving paradigm [22]. There are two common design approaches: design-oriented and design-based.

In our research, we chose the design science research method over the scientific research because of its ease of use and compatibility with IT artefacts. We chose the design-oriented research method because it suits our research problem and artefact better.

In the next chapter, we will cover our research problem, which is inebriation. We will discuss inebriation, our research environment, the effects of inebriation and a potential solution.

Chapter 3 Inebriation Recognition

3.1 Introduction

The consumption of intoxicating substances, such as alcohol, leads to inebriation. Blood Alcohol Concentration (BAC) is one form of measurement used to recognise the level of inebriation in individuals. It is used to quantify one's level of intoxication for both medical and legal purposes. After consuming alcohol, one's BAC can be measured in blood, saliva, urine, breathe or respiration [1].

After consuming alcohol, psychological and physiological changes begin to take place and inhibition, emotional ability, self-awareness, judgment, and balance are affected [1]. As more alcohol is consumed, the changes intensify and become more apparent. An individual will also feel relaxation, higher body temperature and an altered mood. These changes also vary depending on the level of inebriation.

BAC testing can be done at a hospital, or during an autopsy for fatal cases. Law enforcement officers also conduct random breath testing on drivers on the side of the road to determine a drivers' alcohol impairment levels. Failing a breath test may have legal implications, such as prosecution, fine and/or imprisonment.

An ideal method to detect inebriation is one that operates in real-time, is less intrusive, more convenient, and efficient. Many methods exist in literature to recognise inebriation. These methods are discussed in detail in this chapter. The definition of inebriation is provided in Section 2. Section 3 discusses the effects of inebriation. In Section 4 we discuss the existing methods to detect inebriation in detail. We finish off with a fitting conclusion.

3.2 Definition of Inebriation

A human body handles adversity well, such as dealing with the injection of toxins and poisons. The human consumption of alcohol affects physical and cognitive functions and has legal consequences in many countries such as inebriated driving and underage drinking. Inebriation is

a temporary “situational impairment” that affects the subject’s interaction with their environment [3]. 20% of consumed alcohol goes into the bloodstream, while 80% goes into the intestines [29].

Blood-Alcohol Concentration (BAC) is the most common metric used to measure the amount of alcohol in the human body at a given time, expressed in grams of alcohol per litre of body fluid. In the USA, the National Advisory Council on Alcohol Abuse and Alcoholism defines inebriation at above 0.04% BAC [3]. According to Wu et al., having 0.02% BAC level in one’s system will lead to loss of judgement [30]. Feelings of relaxation, body warmth and altered mood are also reported at this BAC level. As BAC gradually rises to 0.08%, muscle coordination ability, reasoning ability and inhibition decreases [29]. When BAC levels reach 0.10% - 0.15%, there is definite deterioration in reaction time, slurred speech, poor reasoning capability, vomiting and major balance loss [29]. A BAC of 4 grams per litre can result in a coma while a BAC of 4.5 – 5.0 grams lead to death [31].

Driving under the influence (DUI) of alcohol is a criminal offense involving operating a motor vehicle after consuming alcohol beyond the legal limit [32]. In countries such as Australia, Austria, France, Italy and Israel, it is illegal to drive with a BAC level of more than 0.05% [33]. Norway has a BAC limit of 0.02%. In South Africa, you are breaking the law if you are operating a vehicle with a BAC level of above 0.05%.

Alcohol consumed on an empty stomach is absorbed faster, while alcohol consumed on a relatively full stomach is absorbed more slowly [29]. This results in differing BAC peak time, which can play a role on ability to function and/or drive a motor vehicle. The type of alcohol consumed also plays a big role on determining BAC peak time. It takes a distilled spirit an average of 36 minutes to reach BAC peak, 54 minutes for wine and 62 minutes for beer [29]. This means an individual may peak their BAC level long after they finished their last drink.

According to a study by the WHO, vehicle accidents will become the 5th highest cause of death if inebriated driving is not mitigated [30]. Inversely, 70% of the world population can be protected by handling inebriated driving effectively. Alcohol consumption leads to approximately 5.9% of all global death (3.3 million) each year [34]. Excessive alcohol use is the third leading lifestyle-related cause of death in the United States and about 1.24 million people die on the road annually

[35]. If this trend does not change, the death toll on the road is expected to reach 2.4 million by 2030, mostly caused by driving while inebriated [36].

Every 33 minutes, a person in the world is dying in a road accident caused by drunk driving [2]. In 2014, about 27 people died daily due to inebriated drivers in the United States of America (USA) [3]. Alcohol consumption leads to physical and mental harm [37]. Accidents on the road cost USD 500 billion a year, which is between 1% to 3% of the world's GDP [38]. Inebriated driving puts both the driver and other potentially sober road users at risk.

3.3 Effects of Inebriation

About 10 minutes after consuming alcohol, the effects begin to set in. The heart rate increases to filter the toxins from the bloodstream through kidneys. A few minutes later, alcohol penetrates the blood-brain barrier, affecting the cognitive and neuromotor functions. This leads to loss of balance and inability to walk properly [35].

Alcohol is a big part of human social culture – it helps humans socialize and it enhances our religious ceremonies [1]. Initially, alcohol acts as a stimulant, producing intense feelings of warmth, well-being, and relaxation. This is usually the feeling the alcohol consumers are after. However, inhibition and judgement are soon affected. Fine motor and reaction times also begin to suffer at this stage with an exponential impact as more alcohol is consumed. Females are affected by alcohol more than males and heavier people are affected less than slender ones [39].

Keall et al. [40] randomly selected roadside sites to collect BAC measurements from motor vehicle drivers. This data was used to build a logistic model to estimate alcohol's effects on drivers, the driver's age, and passengers' influence on drivers on the risk of driver fatality in New Zealand. Their results showed that estimated risks increase rapidly with increasing BAC levels. They also found that drivers under the age of 20 were five times riskier than older.

Excessive alcohol consumption heavily impacts our health. It meddles with the delicate balance of neurotransmitters, facilitating the brain's functionality [1]. For long-term drinkers, change in brain cells size can occur, which can potentially shrink the brain's size. Such a change has an impact on one's motor coordination, temperature regulation, sleep, learning and memory.

Eventually, the brain builds up tolerance for alcohol dependence, leading one to experience withdrawal symptoms.

Consistent excessive alcohol consumption weakens the heart. A weakened heart may fail to pump adequate blood to provide to the body's organs [1]. Alcohol can also lead to irregular and rapid heartbeats or elevated blood pressure, resulting in hypertension.

In a nutshell, some of the effects of alcohol include lower inhibitions, lower caution, loss of fine motor coordination or the inability to operate a motor vehicle [3]. Alcohol consumption also results in slurred speech, poor balance, pupil dilation, slowed heart rate and reduced blood pressure. In some instances, nausea, vomiting or loss of consciousness can also occur.

In the next section, we will discuss the existing methods of identifying inebriation in humans. We discuss manual methods and the more automated methods, their use cases and their advantages and disadvantages.

3.4 Existing Methods to Identify Inebriation

The effects of alcohol consumption, both long-term and short-term, enable inebriation to be recognised using direct and indirect biomarkers, automated biometric systems, and several manual methods. Direct biomarkers measure ethyl alcohol. Biomarkers include blood, urine, saliva, and hair analysis. Indirect biomarkers include measuring the effects of alcohol on the body. These methods come with varying performances, advantages, and disadvantages, depending on their environment and problems they are solving.

Direct biomarkers are very effective and have high accuracies, but they are very intrusive and cannot be used practically in a real-time inebriation recognition environment. They are impractical in a real-time setting because extracting the biometric trait such as blood samples is a slow process, and results may only be produced after a few hours to days. Due to this, we do not provide a detailed analysis on them. We cover generally more acceptable methods that have been implemented to recognise inebriation.

3.4.1. Surveys

A survey is a questionnaire created to extract information from a specific population of interest. It is one of the oldest, least convenient, and least effective methods of inebriation. Surveys require the individual to be engaging and truthful, are prone to recall bias and inaccuracies of up to 20% [35]. They require manual input from the user. Due to these reasons, surveys cannot be used to effectively recognise inebriation in real-time.

3.4.2. Field Sobriety Tests

Field Sobriety Tests are administered by law enforcement officers to determine if a driver is inebriated or sober [41]. There are 8 indicators used by law enforcement officers for this test, namely: if the individual can keep their balance, starts walking too soon, stops mid-walk, does not touch heel-to-toe, deviates from the line, request external balance (such as arms), turns incorrectly, and takes incorrect number of steps.

These methods are not reliable as law enforcement officers can administer the tests incorrectly. Due to lack of conclusive test results, the one administering the tests must make their own conclusion, which brings about inaccuracies in recognising inebriation. Walking in a straight line, turning, and repeating the feat can be challenging for sober individuals as well, falsely identifying them as inebriated.

3.4.3. Blood Tests

Phosphatidylethanol (PEth) is formed on the surface of the red blood cell after alcohol consumption [42]. Varga et al. [43] experiment indicated that a substantial amount of alcohol needs to be consumed to elevate PEth. Due to individual differences and different alcohol effects, it is not possible to determine the benchmark for inebriation recognition. However, a PEth test (blood test) can be used to positively recognise inebriation within an hour of alcohol consumption with the maximum PEth concentrations recorded between day 3 and day 6 of consumption [44]. Alcohol can still be recognised days after consumption because once the PEth is formed around the red blood cell, it degrades very slowly. Moreover, PEth can be recognised weeks after alcohol consumption [42].

3.4.4. Urine Test

The ethyl glucuronide test (EtG) is a urine test used to recognise inebriation [42]. After a few alcoholic drinks, EtG can be recognised in one's urine for up to 2 days. With heavy alcohol consumption, EtG can be detected for a period of 10 days. Urine is sensitive to small amounts of ethanol. Urine tests can be effectively conducted in environments where sobriety is required, such as rehabilitation centers and workplaces.

The EtG can, however, incorrectly detect inebriation when an individual is continuously exposed to hand sanitizers or mouthwashes with ethanol. To tackle this problem, a higher EtG threshold is used when conducting urine tests.

3.4.5. Hair Test

EtG can also be measured in one's hair to monitor alcohol consumption use over time. EtG is incorporated into the hair follicle surface as it grows, providing a record of alcohol consumption over time [42]. Using this measurement, alcohol consumption can be detected for as much as 30 days. However, if hair is treated using chemicals with alcohol content, it can be falsely classified as alcohol consumption.

3.4.6. Breath Test

Breath Alcohol Concentration (BAC) measures pure ethanol in the blood, as opposed to ethanol metabolite [42]. The breath test can only be actioned within a few hours of alcohol consumption before it metabolizes. The BAC gradually declines as the alcohol is metabolized.

The most common device used to perform a breath test is a breathalyser. Portable breathalysers were invented in 1931. Their purpose was to allow for law enforcement officers to enforce the driving under the influence (DUI) law in real-time. Taking a blood or urine sample for later analysis in a laboratory is not an efficient and practical way to detain drivers who are driving while inebriated beyond the legal limit.

Alcohol consumed is prevalent in one's breath because it gets absorbed into the blood stream. Alcohol is not digested upon consumption, and it does not change in the blood stream. The concentration of alcohol found in the lungs is proportional to the alcohol found in the bloodstream. As the inebriated individual exhales, the alcohol from their lungs can be recognised and measured by a breathalyser in real-time. A law enforcement officer can then detain the inebriated driver and charge them with a criminal offense. The breathalyser measures one's BAC levels. It uses a chemical reaction that involves changing color when alcohol is detected.

3.4.7. Mobile Applications

A great deal of publicly available mobile applications exist, such as BAC Calculator and IntelliDrink PRO, which enable individuals to record their drinking behaviour. Using data such as height, age, size and weight and information on the drinks (frequency, quantity, and type), these applications attempt to estimate the user's BAC level. These applications are prone to error since they heavily depend on the user's manual input. Studies have since shown that 98 mobile applications underperformed against a breathalyser baseline [3].

Weaver et al. [45] used available BAC calculator iOS and Android apps to test against individuals with a known BAC from a previous study. In their research, they used 384 apps, of which 50% (192) were entertainment apps, 39% (148) were BAC calculator apps and 11% (44) were health apps discouraging the consumption of alcohol. Their results showed a huge variation in the applications' BAC results, with apps tending to overestimate BAC scores. There is also an overall concern over the use of these apps, as many of them are used to encourage the consumption of alcohol, as opposed to discouraging their drinking behaviour.

These applications are not regulated, and their calculations are not always accurate. For individuals looking to improve their alcohol dependence, these applications do not provide the correct information to offer enough help. Also, they require human input to provide BAC results, and manually entered information is not always accurate. Individuals without smart phones or access to one and technology-illiterate individuals cannot use these applications to improve their drinking behaviour. Due to these issues, such applications cannot be used to recognise inebriation effectively.

3.4.8. DUI (Drunk User Interfaces)

Mariakakis et al. [3] developed Drunk User Interfaces (DUIs), a mobile phone-based set of tasks that test the user's motor coordination and cognition. While using DUI, it measures how well one performs the required tasks using human performance metrics. These metrics include measuring the ability to type a sentence or phrase by counting typing errors, measuring the striking of keys using accelerometer and touchscreen information.

The DUI app is made up of 5 interfaces: typing, swiping, balancing + heart rate, simple reaction, and choice reaction. The typing task measures fine motor coordination using repeated target selection. The swiping task measures fine motor coordination using gesturing. The balance + heart rate measures the heart rate using photoplethysmography (PPG) by having the user hold their finger completely still on their mobile camera, with the expectation that the heart rate changes considerably after consuming alcohol. The simple reaction task measures the user's alertness and motor speed. Each of these tasks generates metrics forming the feature set for training a machine learning model to estimate blood alcohol level.

This method measures behavioural effects of alcohol use, but not the amount of alcohol in one's system. 14 participants were used for this study. We believe this is not an adequate dataset for such a study, and more data samples are required to train and test a model to get more conclusive results. They were able to measure the Blood Alcohol Level (BAL) with a mean absolute error of 0.005% against the breathalyser baseline [3]. They used 0.04% BAL as a threshold and had a sensitivity score of 93.9% and a specificity score of 82.3%.

This method of inebriation recognition requires the drinker to own a mobile phone and to use the application when they are intoxicated. We believe these are shortcomings because each user requires an expensive sensor (smartphone). The method requires user participation, and many users find it very inconvenient to go through the entire 5 challenges to calculate their BAL. Also, users might get better at tasks as they become too familiar with them, resulting in failure to accurately measure their BAL. Some of the features they classify as inebriation can be from fatigue, such as the inability to type or swipe well to pass the challenge(s).

These shortcomings make DUI difficult to use for inebriation recognition in real-time. This method cannot be used practically to enforce inebriation recognition in real-time scenarios.

3.4.9. Gait Recognition

Arnold et al. [35] explored inebriation recognition using a smartphone-based gait recognition method. They wanted to identify the number of drinks consumed using the drinker's gait as they walk with their smartphone in their person (pocket, hand, or bag). They believe neuromotor testing, such as gait analysis, is a reliable way to recognise inebriation in humans. This approach infers inebriation using machine learning classifiers on a mobile application, which analyses data obtained from the accelerometer. The classifier was made up of 3 classes: 0-2 drinks (sober), 3-6 drinks (tipsy) and more than 6 drinks (drunk). Their method did not perform well due to insufficient and imbalanced data.

A BAC of 0.04% produced noticeable gait unsteadiness while the walking stride increased in length [46]. Kao et al. [46] proposed a gait anomalies recognition system for inebriated individuals. They argue that such a system can recognise walking patterns influenced by the consumption of alcohol as individuals walk carrying their smart phones in their pockets. Their method only requires the user to place their phone in their trouser pocket looking upwards. This makes their method less robust as not everyone walks with their phone in their pockets as some clothing requires one's smartphone to be in their hand or bag.

They collected accelerometer data from a smart phone and analysed it for both sober and inebriated gaits. 3 individuals (2 males and a female) were used for data sampling. The experiment's results show that inebriated individuals' gait have larger variance than that of sober individuals. They classified inebriation by using gait's step time variance and gait stretch variance.

Gait recognition has uncertainty issues and difficulties that are slowly being addressed by scholars. The placement of the mobile phone (hand, pocket, or bag) has a strong influence on the accelerometer data sent to the machine learning classifier. Gender, weight, fatigue, mood, ground conditions, injury, urgency of the situation, clothing and walking behaviour strongly alter one's

gait, and becomes difficult to generalise the features extracted for classification. These shortcomings make gait recognition difficult to use for inebriation recognition in real-time.

3.4.10. Thermal Infrared

Inebriation is a stimulating physiological condition which can be inspected using infrared images. This is because physical characteristics of arteries and vessels of an inebriated individual change after consuming intoxicating substances [47]. A thermal gradient appears at the region of the vessels, and intoxicated individuals have more active blood vessels. It has been proven that a person's bloodstream changes in temperature after consuming alcohol, thereby separating an inebriated person and a sober person [48]. The human sclera and iris maintain the same temperature for a sober individual, but the temperature increases for intoxicated individuals [47].

Neagoe and Carata [36] used thermal infrared facial images to recognise inebriation. Their method consisted of the following pipelines: thermal infrared image acquisition; Pulse-Coupled Neural Network (PCNN) image segmentation; feature selection using Principal Component Analysis (PCA) algorithm cascaded with the Linear Discriminant Analysis (LDA) algorithm and a Support Vector Machines (SVM) classifier. They used 10 subjects (6 males and 4 females) in both sober and inebriated conditions for data sampling. Their neural network uses a genetic algorithm to optimize its parameters, which leads to more exploration before convergence. Their method achieved a 97.5% accuracy rate [36].

Koukiou et al. [48] also used thermal infrared images to detect inebriation. They used pixel values on specific 2D points on the face to separate inebriated individuals from sober ones, making up the feature space. The circulatory system of an inebriated person's face increases its flow as they consume alcohol [36]. They used a Fisher Linear Discriminant approach for feature dimension reduction. They used 4 individuals in both sober and inebriation states to test their method. Euclidean distance was used to separate inebriated image's pixels from the sober ones. Their research paper does not provide results of their experiment.

Koukiou and Anasassopoulos [39] used neural networks on infrared thermal facial images to recognise inebriation. 20 people (12 males and 8 females) were used for data sampling. Due to the small size of their sample, their experiment only provides indication on parts of the face that

are likely to show significant changes after inebriation, rendering them possibilities for use in inebriation recognition. Their experiment showed that the forehead and nose change thermal behaviour with alcohol consumption. Only locations in which blood vessels are present in a dense manner are good candidates for inebriation recognition. Their conclusion show that large neural structures on the forehead and nasal area can be used to recognise inebriation [39].

Bhuyan et al. [47] implemented a multimodal biometrics system to recognise inebriation. Their system was made up of gait, face, and iris recognition. They used thermal images to study activities of facial blood veins and temperature distributions and variations on the eye socket of inebriated individuals. For a sober person, vessels around the nose and eyes and nearer to the forehead region remain inactive and smooth, but they become more active for an inebriated individual. They used Curvelet Transform to capture a face's edges to identify an intoxicated curvelet, and SURF (Speeded Up Robust Features) for temperature change recognition in the iris and sclera. Optical flow was implemented to determine walking behaviour of an intoxicated and sober individual. They classified inebriation using Random Forest and SVMs. 40 healthy individuals (30 males and 10 females) were used for data sampling. Their method obtained 89.23% accuracy using face biometrics, 100% using eye biometrics and 100% using gait.

These methods had varying levels of success, but they all require expensive equipment such as infrared cameras to implement. They used a very small dataset to experiment on their models, which may influence their results. Also, although more robust, the use of multimodal system can impact speed, rendering systems impractical to use in real-time scenarios.

3.4.11. Real-Time Inebriation Monitoring

Chatterjee et al. [49] developed a system to recognise driver drowsiness and loss of vehicle control due to fatigue or inebriation. They implemented a multimodal biometrics system which uses computer vision techniques for facial landmark recognition and motion recognition using a smartphone camera for data input.

Their system continuously analyses the driver to detect drowsiness by checking the frequency of eye blinks using Eye Aspect Ratio (EAR). The system also checks for the driver's head and body orientation with respect to the steering wheel to detect drowsiness and/or inebriation.

The sensor used in this study can be expensive since users will be required to acquire a smartphone with a good camera for their method to work effectively.

3.4.12. Heart-Based Inebriation Recognition

Individuals who are dependent on alcohol show greater heart rate (HR) and heart rate variations (HRV) variations and are at a higher risk of having cardiovascular diseases [32]. Chronic alcoholics have a decreased heart rate [50]. It is possible to recognise inebriation using bio-signals such as heart-based signals. The most common ways to obtain heart-based signals are using ECG and PPG sensors.

Electrocardiogram (ECG) signals are produced by the changes in voltage during heartbeats and are useful for observing cardiac systolic and diastolic activities [32]. Although these signals correspond to the heart's current condition, they can also be used as an indicator of physiological or psychological change. The ECG signal, which is mainly characterized by P, Q, R, S and T fiducial markers, is a proven signal to indicate intrinsic human status since it measures the electrical activities of the heart [38]. The ECG signal is significantly affected by alcohol consumption [32].

Photoplethysmography (PPG) is an optical technique used to detect volumetric changes in microvascular blood flow in the fingers or skin [32]. It detects r peaks and is mainly used for HRV. The voltage variations are like HR and ECG readings [32].

Koskinen et al. [50] studied acute effects of alcohol consumption on heart rate and blood pressure variability and baroreflex sensitivity. 12 male subjects were used for data sampling. Finger blood pressure and ECG signals were used for the study. The research showed that sharp alcohol consumption leads to a significant decrease in HRV due to reduced vagal modulation of the heart rate [50].

Wang et al. [32] developed a biometrics system to identify inebriation using ECG and PPG sensors. They implemented a fast and accurate inebriation recognition system using SVMs for classification. They achieved an average accuracy score of 95%. Due to similar results produced

by both ECG and PPG, they opted for PPG features because they are more convenient and easier to acquire.

Wu et al. [38] developed a Support Vector Machines (SVM) classifier for Drunk Driver Recognition (DDD) based on ECG signals. They studied and analysed ECG signals from inebriated drivers to identify drunk syndromes. 50 volunteers were used for data sampling. Using weighted feature vectors of ECG signals improved their classifier accuracy by 11%. Their method improved accuracy by 8% to 18% compared to prevailing methods [38]. The experiment achieved accuracy, sensitivity, and specificity of 88%, 88% and 87%, respectively.

Sensors for heart-based signals are expensive to acquire. User participation is required to get heart bio-signals, which is very inconvenient and intrusive for the user. Also, heart data contains information about the user including their health information and gaining access to this information is considered an invasion of privacy. These shortcomings make it less desirable to implement heart biometric systems for inebriation recognition purposes.

In the next section we will provide justification for our research.

3.5 Research Problem Justification

Alcohol consumption renders an individual incapable of safely operating a motor vehicle. Alcohol consumption leads to 5.9% of all global deaths annually, which is 3.3 million people. Every 33 minutes, a person in the world dies on the road due to inebriated driving. According to a study by the WHO, vehicle accidents will become the 5th highest cause of death if inebriated driving is not mitigated [30]. The first step to fixing problems that come with alcohol consumption is recognising inebriation.

Inebriation can be recognised using direct or indirect biomarkers, using mobile applications or more automated biometric systems. As shown in section 4, direct biomarkers, namely blood, hair, breathe and saliva have been used to recognise inebriation. These biomarkers are very intrusive as they contain sensitive biometric information about individuals.

Direct biomarkers are also very inconvenient as they require user participation. Some biomarkers are not accurate. Drinking a mouth wash with alcohol contents can be recognised as inebriation

during a breath test. Treating hair with chemicals containing alcohol content can result in inebriation recognition during hair test. The sensors to get biomarker samples are expensive. Due to these reasons, biomarkers cannot be used for a real-time inebriation recognition system effectively.

A real-time inebriation recognition system is important because it automates capturing biometric traits and produces inebriation recognition timeously. This enables the inebriation recognition system to be deployed in environments that require fast inebriation recognition such as roadblocks implemented by traffic officers and access control points where only sober individuals can enter a building or a vehicle. A slow inebriation recognition system such as blood tests will cause traffic or long queues, considerably inconveniencing the users.

Field Sobriety Tests generate inconclusive results and require a great level of participation for both parties involved. An officer can administer the test incorrectly and a subject can fail the test due to fatigue and be flagged as inebriated. This leads to a high misclassification rate, rendering the process ineffective in recognising inebriation.

Surveys are the oldest, least convenient, and least effective way to manually recognise inebriation. They suffer from recall bias and reach inaccurate rates of 20%. Surveys require an unrealistic level of honesty from the subject to be of any value. Manual input is required from users, which is time consuming and inconvenient. Survey analysis is a slow process and cannot be used to build a real-time inebriation recognition system.

Mobile apps such as BAC Calculator and IntelliDrink Pro use information such as height, age, size and weight of individuals and the frequency, quantity and type of drink consumed to detect inebriation. These apps require user participation and a high level of honesty and recall from users, which is very inconvenient to the user. The apps require sensitive behavioural information which is very intrusive. These apps have a huge variation on detecting inebriation using the same information. Their calculations are not regulated, and their results are not always accurate.

Gait recognition has been used for real-time inebriation recognition, which comes with a lot of uncertainties as a lot of factors can influence the way a subject walks, such as gender, physical

orientation, fatigue, mood, clothing, injury and urgency. It is difficult to generalise the features extracted for inebriation classification, leading to low classification accuracy rate.

Thermal infrared images have been used to recognise inebriation. Although effective, this method requires a special equipment which is expensive. Many systems existing in literature use a small sample to test their methods, which can lead to inaccurate and inconclusive results.

Heart-based methods have also been used to recognise inebriation. Inebriation leads to a higher heart rate and heart rate variations than sober individuals. ECG and PPG sensors are used to extract heart information. These sensors are expensive, and the method is very invasive as it contains health information of subjects. This method is also inconvenient as it requires subject participation to extract heart information, although PPG and ECG sensors are now prevalent in wearable devices.

Our research aims to find a real-time inebriation recognition system that uses an affordable sensor, is less intrusive, automated for convenience and exhibits good accuracy. Such a method can be effectively and efficiently deployed in environments where real-time inebriation recognition is required. These environments include on the road where subjects potentially drive while inebriated, at workplaces where sobriety is required, at sporting events for professionals and at bars to minimise excessive intoxication. Such a method can save lives by minimising inebriated driving. A real-time inebriation recognition system on the road will reduce traffic and inconvenience to the subjects and can be used on more subjects, thus more effective in pruning out inebriated drivers.

3.6 Conclusion

Inebriation is measured using Breath Alcohol Content (BAC), and the more alcohol is consumed, the higher the BAC score. Alcohol causes psychological and physiological effects soon after consumption, altering perception, weakening fine motor coordination, inhibition, and the ability to solve problems. These effects have made alcohol consumption one of the leading causes of social imbalance, lifestyle diseases and accidental deaths on the road.

Chapter 3: Inebriation Recognition

Due to its effects, it is possible to detect alcohol consumption in humans. This can be done manually using direct and indirect biomarkers. This includes doing a hair, urine, blood, or breath tests. These methods have high false positive scores, and are considered invasive, inconvenient, and slow.

Assisted methods can be implemented where user participation is required to detect inebriation. Mobile applications can be used to help calculate inebriation but are not accurate. The required user input can be wrong, which will result in wrong output.

Automated methods exist which do not require user participation and have better accuracy. Gait recognition, thermal images and heart-based biometrics can be used with varying performances. Gait recognition has its performance issues linked with where the phone will be placed, the age, gender, height and physical orientation of the subject, the type of clothing, agency of the situation and walking surface of the subject. Using infrared is expensive as it requires using special cameras. Heart signals are very invasive as they contain sensitive medical information, and they may require effort from the user to extract features.

In this chapter, we provided our research problem and research environment. We discussed the existing methods in our problem environment and the problems related to these methods. By completing this chapter, we have met the first part of objective 2 of our research. In next chapter we will discuss computer vision as a potential solution to our research problem and environment.

Chapter 4 Computer Vision

4.1 Introduction

The advancement in artificial intelligence and machine learning have facilitated rapid growth in computer vision (CV). CV is a subfield of artificial intelligence focusing on engineering the complexity of the human visual system to enable computers to see, identify and process images and videos. Computer vision has provided solutions to object localization [51], image segmentation [52], object recognition [53], object classification [54], object tracking [55], and context and scene understanding [56] problems.

The purpose of this chapter is to discuss computer vision as a potential solution to recognising inebriation in humans. An overview of computer vision concepts, environment, and usage as a solution to various problems is covered in detail. The methods that are briefly discussed in this chapter are covered in detail in the model chapter. We briefly discuss the pipeline that makes up a general computer vision solution and the different methods that form part of each module in the pipeline.

In section 2 we define computer vision. Section 3 discusses the history of computer vision and its progress over the years. Section 4 focuses on the computer vision process, namely image capturing, preprocessing methods, the Region of Interest (ROI) methods, feature extraction methods, deep learning methods and classification methods. In Section 5 we provide examples of existing computer vision applications. Section 6 and 7 we provide advantages and disadvantages of computer vision, respectively.

4.2 Definition of Computer Vision

The advances in artificial intelligence, deep learning and neural networks techniques have facilitated rapid growth in computer vision in recent years. Computer vision (CV) is the automated extraction of information from images [57]. CV aims to impart a mixture of human intelligence and instincts to a machine. Computer vision's goal is to extract image data and use it to infer something about the world [58]. CV retrieves symbolic information from an image using

models constructed from scientific disciplines such as statistics. It is study that covers multiple domains and involves the use of specialized methods and learning methods.

Computer vision uses digital images to mimic human vision, achieved through image acquisition, processing, analysis and understanding [59]. Image acquisition is a process that converts an optical image into a numerical data array that can be further manipulated by a computer [60].

The information acquired at this stage must be preprocessed or denoised for better exploitation in the latter stages. Image processing is made up of advanced applied mathematics algorithms and techniques applied to the digital image to infer low-level information contained in the image [61]. These algorithms and techniques include edge recognition, segmentation, classification, and feature recognition.

Image analysis and understanding analyses the data given, paving the way for effective decision-making. High-level algorithms are used together with image data and the low-level information for image processing. Image analysis includes segmentations, measurements, classification, and statistical evaluation [62] and is used in scene mapping [63], object recognition [64], and object tracking [55].

The next section will discuss the history of computer vision and how it has developed into an important subfield of artificial intelligence.

4.3 History of Computer Vision

Before computer vision came, image analysis, such as x-rays, MRIs and high-resolution space photography were done manually. Computer vision was introduced into research in the 1960s, with the intention of mimicking the human visual system. Researchers wanted computer vision to automate the image analysis process by extracting meaning from visual data. Seymour Papert, a professor at MIT, launched a Summer Vision Project in the 1960s aimed at solving the computer vision problem, aimed at developing a system that performs background and foreground segmentation and extract non-overlapping objects from images [65]. Although it was

meant to be a summer project, 54 years later we are nowhere near a general-purpose computer vision solution.

Computer vision evolved over the decades, paying more emphasis on geometrical algorithms in the 1980s. This resulted in an overall improvement in computer vision performance. In 1998, Yann LeCun et al. released LeNet-5, the first modern Convolutional Neural network (CNN) algorithm [66]. The algorithm performed extremely well in classifying handwritten digits, and the MNIST dataset of handwritten digits was developed. However, resources were a problem.

In the late 1990s, computer vision researchers shifted their focus to using feature-based object recognition. David Lowe's SIFT algorithm was developed in 1999 to be invariant to rotation, location, and illumination [67]. In 2001, Paul Viola and Michael Jones developed a real-time face recognition framework that learned features that could help localize a face [68].

The ImageNet Large Scale Visual Recognition Competition (ILSVRC) project was launched in 2010 to provide a standardised dataset for object classification [69]. An annual competition was also run that allowed evaluation of different methods for object recognition. ILSVRC contains over a million images with more than a thousand classes. Since inception, the ImageNet challenge has become a benchmark in object classification and recognition.

In the early 2010s, due to GPU-based processing schemes, there was an increase in focus on artificial intelligence and deep learning techniques. Apple's Siri and Amazon's Alexa are natural language processing applications that were developed using artificial intelligence to answer questions, make recommendations and exercise actions. Mobile smartphones were developed with face recognition-enabled cameras and face recognition-based biometrics systems for access control. Faster, powerful, and easily accessible machines have facilitated the growth of artificial intelligence and deep learning. Deep learning allows algorithms to train themselves, improve over time and provide solutions to varying environment problems.

In 2012, a computer vision breakthrough happened when AlexNet won the ImageNet challenge [70]. Before AlexNet, the error rate in object classification was 26%. AlexNet achieved a 16% error rate in object classification. Since 2012, the error rate in object classification methods has

gradually decreased, and variations of CNNs such as GoogLeNet [71] continue to win the ImageNet challenge.

According to the inventors of the ImageNet challenge, computer vision algorithms require vast amounts of data to learn efficiently [69]. The rise of social networks, the availability of mobile smartphones and open-source projects such as ImageNet driven at improving computer vision algorithms have helped create large-scale image databases that researchers can use to train and develop their algorithms. This has resulted in an overall improvement in computer vision solutions.

Making a computer "see" has proven to be harder than first assumed. Computers are still unable to be an all-purpose general "seeing machine" [58]. For a computer vision solution to perform efficiently, it requires context and constraints to a specific domain such as face recognition or animal classification. The challenge in computer vision is the need for human intervention to tag and classify training images for supervised learning algorithms. Deep learning algorithms then use this information to create benchmarks for future image classification.

The next chapter focuses on the high-level computer vision process. It covers the state-of-the-art methods that have been created in computer vision and their intended purpose. These methods are discussed in detail in chapter 5.

4.4 The High-Level Computer Vision Process

Many state-of-the-art computer vision methods exist in literature with varying levels of accuracy. These methods have been implemented for different problems that exist in different domains and environments such as the object recognition functionality in self-driving cars and face recognition in access control environments. A general computer vision solution is made up of a pipeline consisting of an image capturing sensor, the preprocessing module, the region of interest module, the feature extraction module, and the classification module such as in Bai et. al. [53]. Each of these modules is made up of varying algorithms suitable for specific environments. This section aims to briefly discuss the common algorithms in a general computer vision pipeline. A more in-depth discussion on the pipeline and the methods used in the study can be found in the Model chapter.

The first step in a system is image acquisition. In this stage, the input is captured using an appropriate sensor for the scene in question. Poor input capturing may lead to a poor performing system. Excellent input capturing makes the job easier for modules later in the pipeline. Computer vision uses a general sensor (camera) to capture an image or video as input**.**

Preprocessing or image processing is an integral part of computer vision, following input capturing in the pipeline. Preprocessing is used for correction of problems that might have arisen during input capturing. These problems might be sensor or lighting related, such as dead pixels, shadows obscuring local structure or uneven lighting [72]. Histogram equalisation and Laplacian of Gaussian are examples of preprocessing methods.

In localisation, we try to draw a bounding box around our object of interest. If there are more objects of interest in the image, the aim is to draw bounding boxes on all of them. Although effective, localisation is a complicated method due to environmental issues such as image position, orientation, lighting conditions and resolution [73]. Viola-Jones and local binary patterns are examples of localisation methods.

Feature extraction is a very important step in computer vision. After getting the region of interest, we need to extract the necessary features needed for object classification or recognition. Feature extraction is used for dimensionality reduction, enabling the classification algorithm to only work with the data that is of interest, while discarding the rest. Histogram of Gradients (HOG) and Principal Component Analysis (PCA) are examples of feature extraction algorithms.

Classification is an important function in computer vision. It is used to predict class labels from unknown data [74]. Using distance or a supervised machine learning model, classification is usually the end goal in a computer vision system. Classification is used to put image data into classes they belong. Various computer vision classification methods exist such as support vector machines (SVMs), random forests and gradient boosted trees.

Deep learning is an information processing paradigm inspired by biological nervous systems [75]. A traditional deep learning algorithm has connections that go forward from the input layer to the output layer through one or more hidden layers. Given input, its value is propagated along

these connections, reaching neurons in the next layer. Deep learning algorithms are described in depth in the Model chapter.

4.5 Applications of Computer Vision

Since inception, computer vision has been used in various environments to solve problems. Computer vision has been successfully deployed in retail, automotive, healthcare, agriculture, banking and industrial environments. Computers have proven to be efficient in recognizing images, which has resulted in extensive investments by big technical giants like Google, Microsoft, Facebook, and Amazon computer vision research.

Amazon recently opened Amazon Go, a futuristic store that does not require shoppers to queue to pay for their purchases. The store is fitted with cameras on the ceiling above the aisles with visibility on the entire shop. These cameras use CV to determine when an object is taken from the shelf, creating a virtual basket for the shopper. Shoppers walk out of the store with their products, with the application sending an invoice and charging the cost of products to their Amazon account.

StopLift [76] developed a computer vision system that helps reduce theft at store chains by detecting checkout errors or cashiers who maliciously avoid scanning a product at checkout. The system uses video cameras and point-of-sale systems to monitor as cashiers scan products at the check-out counter. A product not scanned is regarded as a “loss” by the system, and the incident is reported to management for further action.

Waymo [77] is a computer vision driven project which uses self-driving cars to improve transportation. The self-driving cars are equipped with sensors that can detect 360-degree movements of other users of the road up to a distance equivalent to 3 football fields. Their system can safely operate through daily traffic.

Tesla [78] has also developed self-driving cars fitted with eight cameras for 360-degree visibility around the car up to 250 metres. The car has 12 ultrasonic sensors that can detect hard and soft objects, and can see through the fog, heavy rain, and dust. The camera, called Tesla Vision, is

built on deep neural networks, and can deconstruct the environment to enable the car to navigate complex roads.

Gauss Surgical [79] developed a blood monitoring system that estimates real-time blood loss during medical situations such as surgery and giving birth. Their system recognizes haemorrhages better than the human eye. The system captures images of blood on surgical sponges and suction canisters, which are then processed by cloud-based computer vision algorithms to estimate blood loss.

4.6 Advantages of Computer Vision

Computers have been trained to effectively analyse both real and virtual worlds through images and videos. They have been successfully deployed in various domains to solve practical real-world problems [80]. Computer vision has made it possible for machines to gather, analyse and understand image data. This has resulted in a faster, simpler way of handling monotonous, repetitive tasks and has propelled the automation of manual tasks, resulting in faster productivity and increased profitability.

Computer vision is less erroneous than its human counterparts in repetitive tasks. Employing computers for repetitive work minimises errors and the cost of fixing errors. Most computer vision algorithms use general cameras that are relatively cheap and do not require special hardware. Many computer vision systems do not require user participation, making them very convenient for users. These reasons make the implementation of computer vision very cost effective.

Computer vision has propelled research innovation in search of ways to improve the quality of life. It has solved new problems that could not be solved before, such as improving transportation using self-driving cars [77] and using anomaly recognition in patient diagnosis of cancer [81].

Computer vision has opened new avenues of security that are safer and more robust. The use of biometric traits such as a human face strengthens security. In this domain, computer vision can

also be used to detect terrorist acts, common crime, and anomalies. Computer vision can also be implemented in hostile environments for humans such as outer space and provide useful insights.

Computer vision is a well-researched discipline, and many state-of-the-art algorithms exist in literature to handle different problem domains with high accuracy performances. Projects such as ImageNet challenge facilitate growth in computer vision, opening windows for more practical implementations of computer vision solutions to real-life problems. The LeNet algorithm's performance in 1998 resulted in its deployment in the banking sector for character recognition [66]. Today, GoogLeNet's superior performance has resulted in the introduction of self-driving cars [71].

4.7 Disadvantages of Computer Vision

After decades of computer vision research, building a computer vision system has proved to be a lot harder than first imagined. Making a computer "see" was something that leading experts in Artificial Intelligence thought would be very simple, but we are still far from building a general-purpose "seeing machine" [58]. Computer vision has proven to be difficult due to the complexities inherent in the visual world. An object may be seen from any orientation, in any lighting conditions, under any form of occlusions, and a computer must be able to see in any form of setting and be able to extract useful information about the image.

Computer vision algorithms such as VGGNet require high computational time to perform efficiently [63]. This can directly impact the computational resources and render computer vision incapable of real-time object recognition/classification for certain domains and problems.

Computer vision can be considered to infringe on an individual's privacy. Computer vision can be implemented to identify and recognize faces, track individuals, and learn their behavioural habits. This information about individuals can be very sensitive. Privacy in computer vision is an ongoing contentious issue that questions the ethics of implementing such a system.

Today, computer vision is implemented in many domains such as gaming, entertainment, government, security, and hospitals. In gaming, misclassification of an object is an inconvenience at best but a misclassification in hospitals might result in patient misdiagnosis

with fatal consequences. Although computer vision methods have high accuracy, they are not 100% accurate and are systems that come with their own limitations.

Computer vision can be implemented in a pipeline, and if a single module in the pipeline fails to play its role, the entire pipeline might be compromised. If wrong features are extracted from an image in the feature extraction module, the rest of the modules in the pipeline may not operate efficiently.

A computer vision system requires training data to achieve the required accuracy. The training data required for a robust performance can range in millions of images. Each image for training the computer vision system might require manual tagging and classification, which can be cumbersome and inconvenient.

4.8 Conclusion

Computers can capture images at high resolutions containing more detail than a human vision system. They can measure differences between colours with high accuracy, but they struggle to find context in the images. Computer vision is a subdiscipline of artificial intelligence and machine learning birthed from the human visual system which aims to provide context to images.

Computer vision was intended to be a summer project but has remained an unsolved problem until today. The advent of faster, powerful, and easily accessible machines has facilitated the rapid growth of computer vision as a discipline. Convolutional Neural Networks (CNNs) have made major breakthroughs in character recognition, face recognition, pedestrian recognition, robot navigation, image restoration, object recognition and semantic segmentation.

Since inception, computer vision systems have been deployed in retail, automotive, healthcare, agriculture, banking, and industry. Big corporations such as Amazon, Facebook, Google, and Tesla have invested billions in computer vision research with astounding results in terms of innovation and return profits. Today, computer vision can be used for self-driving cars, diagnosing patients, monitoring crops, detecting anomalies such as terrorist threats, and automating repetitive tasks with outstanding results.

Chapter 4: Computer Vision

The performance of computer vision on various domains gives us confidence that computer vision can be implemented to recognise inebriation in humans. By completing this chapter, we have met the second part of research objective 1. The next chapter focuses on similar systems in literature that attempt to solve the inebriation recognition problem in humans using computer vision.

Chapter 5 Inebriation Recognition Using Computer Vision

5.1 Introduction

Physical traits of alcohol consumption are slurred speech, speaking too loud or too fast, loss of balance and slow reaction time. Many systems have been built to leverage the effects of alcohol to recognise inebriation in humans such as breath, blood, hair or urine tests and field sobriety tests as shown in the Inebriation Recognition chapter. Other more advanced systems include recognising inebriation using gait [35], heart-based biometrics [32] and thermal infrared [48].

Due to the advancement of artificial intelligence, deep learning and hardware technology, computer vision research has flourished. Computer vision has been deployed in shopping malls, airports, security, law enforcement, entertainment and more. It has been used to solve problems such as pose estimation [82], vehicle recognition [83] and video surveillance [84] because of its high performance, non-invasive and non-intrusive nature. Computer vision does not require user participation and can be used in real-time.

This chapter aims to discuss inebriation recognition in humans using computer vision. In section 2 we will discuss existing localisation methods in computer vision. Section 3 focuses on existing methods for inebriation recognition using computer vision.

5.2 Existing Methods in Computer Vision for Localisation

We have discussed both the traditional and deep learning computer vision pipelines in the Computer Vision chapter. In this section we will discuss similar work in computer vision found in literature.

5.2.1. Object Recognition for Computer Vision Using Image Segmentation

Barik and Mondal [85] developed an object recognition system using image segmentation and graph partitioning. They attempted to determine an object from a background where similarly shaped objects are present. If the contrast difference between the foreground and the background

is high, object recognition is simple. The smaller the contrast difference, the harder it becomes to detect edges from the background [85].

Their method uses image segmentation and graph partitioning to detect objects in chaotic backgrounds. They first used the original object being detected to build a feature set, then trained the system using the feature set.

Recent object recognition techniques include the top-down and bottom-up approaches [85]. The top-down approach uses a training stage to obtain class-specific model features. The bottom-up approach builds hypotheses from features, extends them through construction rules and evaluates them using cost functions. A combination of both methods can be used to avoid exhaustive search and grouping. Barik and Mondal used a combination of a bottom-up approach with a graph-based co-segmentation of the image to achieve object recognition. They first built groups of image classes for a training set, then used the test image to search for the object after partitioning the test image using a graph partitioning algorithm.

For image class preparation, for each object class, they prepared a set of characteristics from sample images where objects are manually segmented. Each object is defined by the histogram of the image, the edge matrix of the image and the mean contrast variation inside the edge boundary of the image. For object segmentation, two images were used: one with the object used in class preparation and the other containing overlapping objects. Maximum Ownership Labeling was used to mark the desired object and the weight matrix difference was stored as a parameter for graph partitioning.

They used data augmentation for data sampling. Data augmentation is the process of adding data by adding slightly modified copies of existing data. Four classes of objects were used to test their method: red apples, green apples, brown button, and a white button with holes. They used an image segmentation algorithm to partition different classes of objects in an image and store the resulting feature vector.

Their objection recognition system successfully detected the object amidst similar objects. The performance of their method varied depending on the object being detected. For example, they achieved a 79% recognition rate on flowers and a 100% recognition rate on buttons, which are

both relatively high performances. Their system is very robust in object recognition as it can successfully detect objects in chaotic backgrounds.

To the best of our knowledge, the researchers did not mention their data sampling size. The size of the data sampled on an algorithm in object detection may influence the results as classification algorithms may require a considerable amount of training and testing images to reach high levels of robustness and accuracy. However, the researchers admit their object recognition algorithm struggles to detect multipart objects such as humans.

5.2.2. Computer Vision -Based Recognition and Localisation of Road Potholes

Azhar et al. [86] developed a computer vision-based system to detect pavement potholes on asphalt roads. They proposed a technique to detect potholes in asphalt pavement images and accurately localise them. Visual features of the road surface are classified as either pothole or non-pothole images.

A pothole is a shallow or deep hole in the pavement surface found in different shapes and sizes. Potholes are usually elliptical-shaped but can be of any shape depending on the wear and tear of the road surface. Different types of pothole and non-pothole images are used for data sampling. They used an existing dataset containing 120 images. 50 images were used for training, with the remaining 70 images used for testing.

The Histograms of Oriented Gradients (HOG) algorithm was used for feature extraction. The HOG feature descriptor algorithm is used to represent the shape of the pothole. Firstly, both horizontal and vertical gradients are computed on the image. These gradients are then used to calculate the magnitude and orientation of the gradient. The HOG algorithm results in a feature vector, with its visual representation giving a glimpse about the regions that are different from the surroundings in terms of appearance and shape.

The Naïve Bayes classification algorithm is used for classifying images as pothole or non-pothole images. If an image is classified as a pothole image, it is forwarded to the pothole localisation module which localises the pothole region on the image.

A normalized graph cut segmentation algorithm was used to locate potholes on detected pavement images. They achieved pothole localisations using graph-based segmentation using normalised cuts. Each pothole image is segmented into 12 regions using normalised cuts-based segmentation algorithm. From these segmented regions, the region falling under a certain threshold is selected as the localised pothole.

Their experiment achieved a high accuracy rate of 90% for detecting pothole images. They also had a high precision and recall rate of 86% and 94%, respectively. Pothole localisation had a 100% performance.

They used 120 images which contained both pothole and non-pothole images. We believe this a small dataset to test a method to be very accurate and robust. A small data sample can affect the performance of the algorithm.

5.2.3. Thai Fast-Food Image Classification Using Deep Learning

Hnoohom and Yuenyong [87] developed a system that classifies Thai fast-food images. They implemented a deep learning process trained on natural images using the ImageNet dataset and fine-tuned to generate a predictive Thai fast-food model.

The researchers created a dataset for data sampling using smartphones. Their dataset contains 3960 images. They divided the dataset into 11 groups comprising of different types of Thai dishes. They collected their dataset using a smartphone in varying dishes, backgrounds, and locations. 300 images per group were used for training, and 60 images per group were randomly selected for testing their method.

During preprocessing, they resized the images to improve processing time and used a histogram equalisation algorithm for contrast enhancement. Their method had a high classification rate in the range of 70% and 100% for varying food groups, with an average accuracy rate of 88%. They used a state-of-the-art method and a large dataset which resulted in high accuracy rates and a robust method.

5.3 Existing Methods in Inebriation Recognition Using Computer Vision

In this section, we discuss computer vision methods that have been used to recognise inebriation and their performances.

5.3.1. Detection of Driver Impairment Using Pupillary Light Reflex

Amodio et al. [88] developed a system to detect inebriation using the dynamic analysis of an individual's Pupillary Light Reflex (PLR). The pupil is a hole found at the iris' centre that allows light to reach the retina. It is responsible for sending light to the retina by constricting when the amount of light increases and dilating when it decreases.

Amodio et al. used the pupillary light reflex to reveal an impairment condition caused by alcohol consumption. They investigated the effects of a high BAC on the pupillary light reflex by applying stepwise light stimuli to one eye and recording the pupillary response of both eyes using video cameras.

They proposed a method to analyse the dynamic behaviour of the human pupil to recognise inebriation. They used a two-step method to extract the pupillary light reflex information: first detecting the eye and measuring the iris size. The ROI was then cropped to only contain the pupil. A database of 3 subjects containing their pupillary responses was created consisting of both inebriated and sober subjects. 27 sober and 63 inebriated observations were collected. They used an 8-vector feature space containing the pupil constriction data.

A Circular Hough Transform algorithm was used to detect the iris, the pupil and the center of each detected object. The SVMs classifier was used with its different kernel functions to classify subjects as sober or inebriated. A Decision Tree was also implemented as a binary classifier. To the best of our knowledge, Amodio et al. do not mention which type of decision tree they used for classification.

Their method showed a delay in Pupillary Light Reflection after alcohol consumption, showing a slowing in pupil response. The SVMs method with the polynomial kernel outperformed the other kernels with a maximum misclassification rate of 9.52%. To the best of the authors' knowledge,

this was the first paper to detect inebriation in humans using the pupillary light reflex, which makes it very novel research.

Drowsiness, fatigue, and sleepiness can result in slowed pupillary light reflex, with similar effects to inebriation. An optic nerve injury or an eye defect can also cause abnormal pupillary light reflex. Hippus, a phenomenon that induces a pupillary light reflex without any stimulation can also show effects like that of alcohol consumption. These external factors make it difficult for the classifier to be sure the slowed pupillary light reflection is solely due to inebriation.

An inebriation test requires a subject to pose in front of the sensor, which requires user participation and is relatively inconvenient. To negate the undesired effects of Accommodation Reflex, their method requires subjects to focus on an object at a far distance. This affects their method's performance as there is no way of truly knowing if the subject complied with looking at an object at a far distance or not. Three subjects were used in their research, which we believe is not enough data subjects for method generalisation.

5.3.2. DrunkSelfie: Intoxication Recognition from Smartphone Facial Images

Willoughby et. al. [89] developed a system that classifies intoxication levels using self-portrait images (selfies). They explored the facial changes that occur after alcohol consumption and whether they can be differentiated from changes caused by other factors such as fatigue and drowsiness. Facial images of 53 subjects after drinking 0 to 3 glasses of wine were used to test the performance of their method. They also blurred, rotated, and altered lighting to capture more realistic drinking scenarios.

Their method involved a preprocessing step to detect faces, locate facial landmarks and align faces. To detect faces, they used the HOG method. After face recognition, they detected face landmarks that are likely to change due to alcohol consumption. These landmarks are the mouth, eyes, and nose.

They extracted vectors of facial landmarks, the distance between landmarks and angle of vectors in aligned images as features. These lines were used to detect inebriation through the changing of shapes of the eyes, nose, and cheeks. Alcohol relaxes facial muscles and can potentially change

wrinkles and lines on the face. The Canny edge recognition algorithm was used to extract facial lines and wrinkles as features. Willoughby et al. argue the face reddens after alcohol consumption, especially the cheeks and forehead [89]. We believe this might not be applicable to all ethnicities. They used a segmentation algorithm to identify foreheads that were covered by hair or hats and estimated the forehead's colour and texture using the available face area.

The Gradient Boost Machines algorithm was used to classify subjects as either sober or intoxicated. Other classification algorithms such as SVMs, Polynomial SVMs, Random Forests and Decision Trees were also implemented with inferior results compared to the Gradient Boost Machines algorithm. Their system showed that facial lines of subjects changed significantly after alcohol consumption and facial landmark vectors were predictive features. A drinker's face reddens and relaxes after alcohol consumption.

Their method achieved a high classification rate of 81% classifying subjects as sober for those who consumed 0 to 1 glass of wine and inebriated for those who consumed 2 or 3 glasses. Although the images used for data sampling were captured in an ideal studio lighting and not in real-world representations of drinking scenarios, they mimicked the drinking environment people find themselves in by rotating, brightening, blurring, changing perspective, changing contrast, and adding tint, making their system robust. They also normalised redness classification per individual, that is, differences in redness were measured per person to cater for redness varying among individuals.

53 subjects were used for data sampling, which we believe is relatively small a sample. Researchers admit to not capturing data scientifically, which can have an impact on how the performance of their algorithm is measured. Their system used face recognition, landmark recognition, edge recognition and semantic segmentation algorithm for the forehead redness, lips, and eyes. This makes the system a multimodal system, which can be very slow as it requires capturing various biometric traits, which may affect real-time inebriation recognition. It can also be very inconvenient for the users of the system.

The authors admitted that smiling faces had a high misclassification rate as their method could not detect accurately if the individuals were inebriated or sober. Their method requires an

individual to have a phone with a good camera and Internet access, which we believe is relatively expensive equipment. Only wine was used for the experiment, which we believe is not enough alcohol representation for a general inebriation recognition system. The authors used the number of glasses consumed to detect inebriation. This may lead to inaccuracy as different individuals are affected by alcoholic beverages differently [39].

5.4 Conclusion

Inebriation is caused by alcohol consumption and it affects the consumer's interaction with their environment. Alcohol consumption leads to lower inhibition, lower caution, loss of fine motor coordination and inability to handle tasks that require hand coordination.

Computer vision has been implemented to detect an object from a background with similarly-shaped objects [85], road potholes on asphalt pavements [86] and to classify Thai fast food images [87]. Computer vision has also been used in systems that detect inebriation in humans. Amodio et al. [88] developed a system to detect inebriation using the dynamic analysis of an individual's Pupillary Light Reflex (PLR). Sooraj et. al. [90] developed a computer vision system for inebriation recognition in human drivers using thermal images. Willoughby et. al. [89] developed a computer vision system that classifies intoxication levels using self-portrait images (selfies).

Driving while inebriated has caused many deaths globally. According to a study by the WHO, vehicle accidents will become the 5th highest cause of death if inebriated driving is not mitigated [30]. A system that detects inebriation can be used to minimise the damage of inebriated driving. By completing this chapter, we have completed the literature review part of our research and have met our research objective 1. The next chapter discusses our research model. Our model is made up of methods that can be used to solve our research problem.

Chapter 6 Model

6.1 Introduction

A model is a representation of a phenomenon showing interrelationships of an action and reaction [91]. It must represent the reality being investigated. A model helps researchers relate more accurately to reality, aids them in describing, predicting, testing, and understanding complex artefacts. It is made up of methods and algorithms used to provide a solution to the research problem. It is the blueprint of the research, providing details on how to answer the research questions. A model must relate to other models, show transparency in terms of interpretation, robustness or sensitivity to assumptions made, fertility or richness in deductive possibility and ease of enrichment or ability to modify and expand [91].

In this model section, we will discuss the methods and algorithms that have been used to solve similar research problems in the past and can be used to potentially solve our research problem. These methods and algorithms will be used to develop a prototype for our research, which will be statistically analysed to answer our research problem.

In section 2, we discuss the 5 modules that make up the biometric pipeline which will potentially answer our research questions. These modules are input capturing, preprocessing, localisation, deep learning, feature extraction and classification. In each of these modules, we provide popular methods and algorithms that have been used in the past to solve similar research problems. The methods and algorithms discussed below make up our model. We then provide a conclusion in section 3.

6.2 A Model for Inebriation Recognition in Humans Using Computer Vision

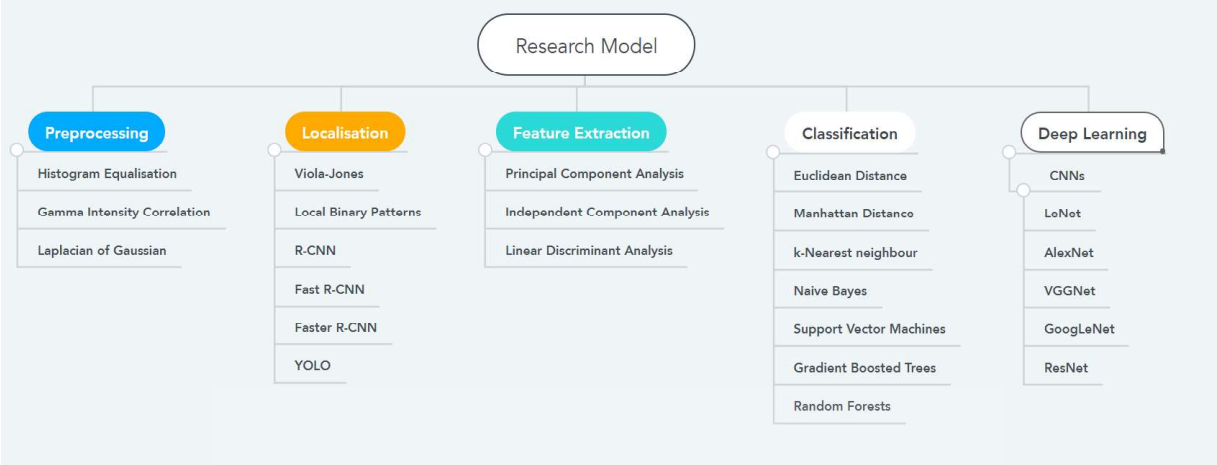

Figure 5: Research model modules and their methods.

Many state-of-the-art computer vision methods exist in literature, each with varying levels of accuracy. These methods have been implemented to solve different problems that exist in different domains and environments. A general computer vision model is made up of a pipeline consisting of an image capturing sensor, the preprocessing module, the region of interest module, the feature extraction module, and the classification module. Each of these modules is made up of varying algorithms suitable for specific environments. This section aims to discuss our research model by providing common algorithms in a general computer vision pipeline. We briefly discussed our model in the computer vision chapter, but in this chapter, we will discuss our methods more in-depth.

6.2.1. Capturing

The first step in our research is capturing input. In this stage, the input is captured using an appropriate sensor for the biometric trait in question. The input capturing stage is important because the information collected here may determine how the remaining processes in the pipeline perform. Poor input capturing can result in noisy data, leading to a poor performing biometrics system. Efficient input capturing will make the job easier for modules later in the pipeline.

Chapter 6: Model

Image formats and their quality have evolved over the years. Raster images use pixels to define an image as a matrix consisting of rows and columns of pixels [92]. Raster images with higher resolution contain more pixels and detail. Today, Graphical Interchange Format (GIF), Joint Photographic Experts Group (JPEG), Tagged Image File Format (TIFF) and Portable Networks Graphics (PNG) formats are the most used encoding schemes.

The GIF format was created in 1987 and was one of the first solutions to electronic image storage [93]. It is the oldest and most popular web-based graphics file format. Its strength lies in its lossless compression algorithm and how it displays the preliminary version of an image before the entire image is transmitted. However, GIF is limited to 256 colours in an image, and complex images may lose some detail when reformatted into GIF, resulting in an image not meeting its full-colour range [92].

The JPEG format was invented in the 1990s specifically for the storage and transmission of photographic images. Unlike GIF, JPEG does not index colour. It allows more colour and contrast resolution than GIF (16.7 million colours as opposed to 256). JPEG can compress larger image files to as little as 20% of the original size but has a lossy compression technique which results in loss of data.

The TIFF format was developed by Microsoft in 1986 specifically for compatibility with image processing devices. It supports the full range of image sizes, resolutions, and colour depths. TIFF uses multiple compression techniques, including lossless compression. TIFF's lossless compression techniques and its use of tags usually result in large file size.

The PNG format was created in 1995 to allow lossless data compression, gamma correction for cross-platform brightness consistency and variable transparency [92]. PNG format supports Palette-mapped, grey-scale and true-colour (RGB) images. True colour comes with 48-bit colour images, which is superior to JPEG's 24-bit colour. Each pixel has four bytes (red, green, blue, and alpha), making PNG powerful for usage in transitions and shadow effects [92]. These features simply make the PNG format one of the best for image processing. We will discuss the preprocessing module in the next section.

6.2.2. Preprocessing

Preprocessing or image processing is an integral part of computer vision, which comes after capturing in the pipeline. Preprocessing is used to correct problems that might have arisen during input capturing. These problems include dead pixels, shadows obscuring local structure or uneven lighting [72]. Noise correction is also performed at this stage to mitigate the effects of noisy input. If the image has been rotated or is taken from the wrong perspective, it can be rotated at this stage. It can also be important to redistribute colour saturation or correct image illumination.

Preprocessing is also used for image enhancement before feature extraction. Enhancement methods are used to optimise feature measurement methods [72]. Enhancement methods alter an image to reduce the lowest grey values to black and the highest to white [94]. Enhancement might also include sharpening and colour balancing. The objective for image enhancement is not to solve problems, but to enable easier extraction of features. Preprocessing is implemented to readily captured input for feature extraction. Common preprocessing methods are discussed below.

a. Histogram Equalisation

The expectation of higher image quality has prompted researchers to develop cutting-edge techniques for image enhancement. Histogram Equalization is a popular image enhancement technique. It is one of the most used techniques due to its effectiveness and simplicity in contrast enhancement [95].

The histogram equalisation method enhances the contrast of an image by mapping the pixel values in a way that the histogram of the resulting image has uniform intensity [96]. An image's histogram is a graphical presentation based on the probability of occurrences of the intensities versus the intensity values in the image. Each bin of a histogram represents the number of pixels with the same value. Many adaptations of Histogram Equalisation exist such as global histogram equalisation (GHE), local histogram equalisation (LHE), fast quadratic dynamic histogram equalisation (FQDHE) and contrast limited adaptive histogram equalization (CLAHE).

b. Laplacian of Gaussian

Laplacian of Gaussian (LoG) is a method used for edge recognition. An edge is the boundary connecting two different regions, a discontinuity or abrupt change in pixel density [97]. The edge of an image removes unnecessary details while retaining important marks on the shape of an object [98]. The LoG method convolves two distinct methods: the Gaussian function and the Laplacian function.

The Gaussian filter is used for smoothing filters and plays a crucial role in edge recognition, acting as a low pass filter. The Gaussian blurs an image and reduces the noise [98]. Laplacian is applied after an image has been smoothened using Gaussian. Laplacian pays close attention to the grey level discontinuities in an image and focuses on the region with slowly varying grey level. The Gaussian function is used for filtering and the Laplacian function for differentiation [97]. The Laplace operator performs well at detecting edges and noise, but to improve its performance, it is desirable to smooth the image first by a convolution with a Gaussian kernel to suppress the noise before using Laplace for edge recognition [99]. The Gaussian in LoG smoothens the image to reduce noise and offsets the influence of the increasing noise caused by the second derivative of Laplacian. In the next section we will discuss localisation.

6.2.3. Region of Interest (ROI) Segmentation

Our research uses object detection for ROI detection. Object detection is a substantial part of the object recognition process and is one of the most promising applications in the field of computer vision [73]. Object detection applications can be found in shopping malls, airports, security, law enforcement and entertainment. Object detection helps in pose estimation, vehicle recognition and surveillance.

In object detection, we draw a bounding box around our object of interest. If there are more objects of interest in the image, the aim is to draw bounding boxes on all of them. Although effective, object detection is a complicated method due to environmental issues such as position, orientation, lighting conditions and image resolution [73].

There are many state-of-the-art methods in the literature for ROI detection. In the following sections we will aim to discuss some of the most common methods.

a. Viola-Jones

The Viola-Jones algorithm is used to detect various objects, especially the human face [100]. Paul Viola and Michael Jones developed this algorithm in 2001, providing a framework for real-time object recognition.

The Viola-Jones algorithm has four stages: Haar-like feature extraction, integral image creation, Adaboost training and cascading classifiers. Haar-like features are used for feature extraction. After extracting features, an integral image is created. This integral representation of the original image frame is used to minimise the amount and time of the necessary calculations [101]. A machine learning technique called Adaboost is used for object recognition. In this technique, the classifiers are built of basic classifiers using any one of the four boosting techniques and are used to select the most suitable features [102]. Classification is reached using cascade classifiers, which can combine many features. The cascades are used to reduce the amount of time for finding the object by quickly eliminating windows without the required object [101].

When an image is inputted, the algorithm selects the Haar features and then scans the image from the top left corner and checks if any object feature is present until it reaches the bottom right corner [100]. To detect the object efficiently, the algorithm scans the image several times using different Haar-like features. Once the strong classifiers have been cascaded, the object and non-object regions can be separated. This object detector can efficiently detect the nose, upper body, lips, eyes, and pupils and has been widely used in computer vision problems.

b. Local Binary Patterns (LBP)

The texture is an important spatial feature used for identifying objects or region of interest in an image [103]. Local Binary Patterns (LBP), introduced by Timo Ojala, is a non-parametric algorithm used to describe the local spatial structure of an image. It is a very good texture descriptors algorithm widely used in numerous applications, achieving very good results in object detection [104]. It is a simple and very efficient method for texture analysis that is robust to monotonic grey change [105].

LBP characterises the local image texture by local binary patterns. It labels image pixels by thresholding the neighbourhood of each pixel and considers the result as a binary number. LBP can filter out image noises using the uniform pattern concept.

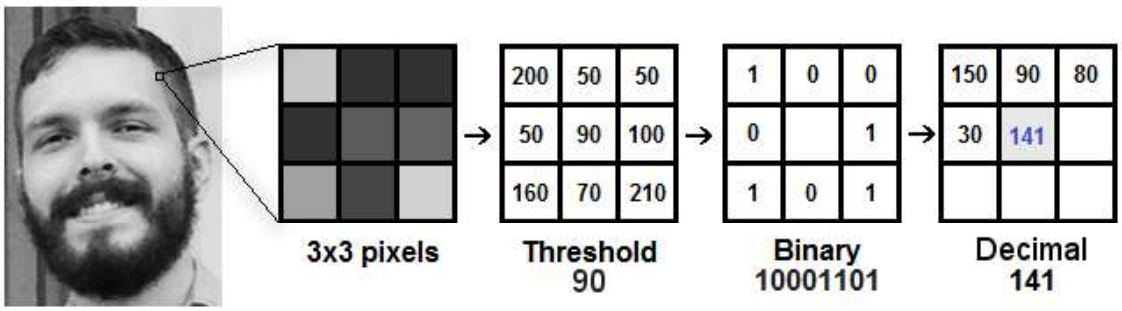

Figure 6: The standard LBP process [106].

In the standard version of the LBP algorithm, a 3x3 neighbourhood is used as shown in the figure above. A 1 is assigned when the neighbouring pixel's value is greater than the centre's, and a 0 is assigned if the neighbouring pixel's value is equal or less than the centre's. The resulting binary string is converted to a decimal, becoming the LBP feature.

LBP is sensitive to image rotation and its texture descriptive power dwindles significantly when an image is rotated [105]. Variations of the LBP method have been implemented to address the image rotation problem, such as Local Binary Pattern Rotation Invariant (LBPROT) [107] and Uniform Local Binary Pattern (ULBP) [108].

In our research, we implemented Uniform LBP because it greatly reduces the size of the feature vector when dealing with higher number of sampling points. Also, research has shown that uniform patterns occur more frequently in texture images compared to non-uniform patterns [109]. This makes uniform LBP computationally more efficient without losing its performance accuracy. The LBP algorithm is used for both localisation and feature extraction.

c. Histogram of Gradients

The histogram of oriented gradients (HOG) algorithm was originally used as a feature extraction technique for pedestrian recognition [110]. HOG is one of the best descriptors in various

domains such as object detection and facial recognition [111]. Descriptors are an effective tool capable of representing feature sets of images [112].

HOG features exploit the distributions of pixel-wise gradients to extract feature vectors describing relative changes in regions of an image [110]. To calculate the gradient, you calculate the gradient of all points of images by applying the horizontal and vertical masks of derivation to the image [111]. For each point of the image, an estimate of the horizontal and the vertical component gradient denoted are obtained. The absolute value of the gradient is usually used because what matters is the contrast between two objects, that is, a black object on white background has the same response as a white object on a black background [111]. After that, the orientation is calculated. The image is divided into several cells, and for each cell a histogram of gradients is constructed by counting the occurrences of the gradient in a bar corresponding to a specific orientation interval [111]. This histogram becomes the HOG descriptor used for classification.

The HOG descriptor is built using a combination of these separate histograms. The HOG descriptor captures edge or gradient structure that is very characteristic of local shape. HOG descriptors are incapable of capturing spatial information [112]. The HOG algorithm is used for both localisation and feature extraction.

d. R-CNN

R-CNN is birthed from combining a regional proposal with Convolutional Neural Networks (CNNs) [113]. CNNs are discussed in the Deep Learning section. R-CNN uses a selective search algorithm to extract 2000 regions from the image, called region proposals. 2000 category-independent region proposals are generated from the input image. Each proposal is used to extract a fixed-length feature vector using a CNN which is then classified with category-specific SVMs [114].

The CNN is used for feature extraction, and its output is fed into an SVM algorithm to classify the presence of the object within the region proposal. The R-CNN algorithm is also used to predict the four points making up the corners of the bounding box marking the region of interest.

R-CNN consists of three modules: the category-independent region proposals, a large CNN that extracts a fixed-length feature vector from each region, and a class-specific linear SVMs [114].

It takes long to train the network as you would require classifying 2000 regions per image. R-CNN may not be implemented in real-time since it takes longer for each test image. Also, the selective search algorithm R-CNN uses is a fixed algorithm with no learning happening at this stage. It uses the exhaustive search approach, which is computationally expensive as we need to search for an object in thousands of windows even for small image size. Training is a multi-stage pipeline, making it expensive in space and time. Object recognition is also slow [115].

e. Fast R-CNN

Fast R-CNN is a popular method for objection recognition using deep CNN [113]. This approach was written by the same author of R-CNN, and is built on the drawbacks of R-CNN to improve the speed of object detection.

Instead of feeding the region proposals to the CNN, Fast R-CNN feeds the input image to the CNN to generate a convolutional feature map. Regions of proposals are identified from the convolutional feature image, and for each object proposal, a region of interest pooling layer extracts a fixed-length feature vector from the feature map [115]. Each feature vector is fed into a sequence of fully connected CNN layers that branch into two output layers: one producing softmax probability estimates over object classes and the other outputting real-valued numbers for each of the object classes containing the four corners of the bounding boxes for the detected object. The ROI pooling layer uses max pooling to convert the features inside any valid region of interest into a small feature [115].

The Fast R-CNN method is faster than R-CNN because it eliminates feeding 2000 region proposals to the CNN. However, it still uses a selective search algorithm to find the region of interest, which is slow and time-consuming.

f. Faster R-CNN

Both R-CNN and Fast R-CNN are predecessors of Faster R-CNN. Faster R-CNN eliminates the use of the selective search algorithm and lets the network learn the region proposals on its own [116].

Faster R-CNN consists of two modules: a deep fully connected CNN that propose regions and the fast R-CNN detector [117]. Like R-CNN, the image is provided as an input to a CNN, which provides a convolutional feature map. Instead of using the selective algorithm on the feature map to identify the region proposals, a separate network, called Region Proposal Network (RPN) is used to predict the region proposals [116].

The RPN is a fully connected CNN that predicts both object bounds and object probability scores at each position. By minimising the classification loss and regression loss for learning region proposals, the RPN is trained to generate high-quality proposals [118]. The object recognition network takes the generated proposals as input and performs elaborate classification and positioning for each proposal by calculating softmax probabilities and bounding-box regression offsets for each proposal.

During training, in the first step, RPN and Fast R-CNN are trained independently. In the second step, both networks are trained alternatively to share convolutional layers. This process also allows the methods to fine-tune their respective fully connected layers for the ultimate model [118].

Faster R-CNN is considerably faster than its predecessors, and can be used for real-time object detection. However, the algorithm requires many passes through a single image to extract all the objects. Due to many modules working consecutively, the performance of modules down the pipeline largely depends on how the previous modules performed.

g. You Only Look Once (YOLO)

You Only Look Once (YOLO) is an object detection algorithm which uses a single CNN to directly predict the confidence of the bounding boxes and the class probability for these boxes

from the entire image [119]. It treats object detection as a regression problem for a spatially separate target box and its category confidence. The method excels at object detection in real-time.

YOLO uses a single CNN trained on full images. It simultaneously predicts multiple bounding boxes containing objects for all defined classes as well as probabilities combined with those boxes [120]. During training, an image is divided into grids of NxN cells, with each cell responsible for the prediction of A possible bounding boxes, which are rectangles surrounding the detected object. These bounding boxes have 5 values, namely x, y, w, h and cs. x and y are coordinates of the centre of the bounding box relative to the grid. W and h are width (w) and height (h) of the bounding box relative to image dimension, and cs is the confidence value which determines how confident the network is about the presence of the object inside the bounding box [120]. The confidence value does not contain object class information.

YOLO is faster than many object detection algorithms and has a low computation overhead [121]. However, due to the spatial constraints in the algorithm, it struggles with smaller objects in an image.

We will discuss the feature extraction module in the next section.

6.2.4. Feature Extraction

Feature extraction is a very important step in computer vision. After getting the ROI, we need to extract the necessary features needed for object classification or recognition. Feature extraction is used for dimensionality reduction, where only the important features are selected, and the unimportant features are discarded. Dimensionality reduction ensures we do not waste computing resources on unimportant features. There are many algorithms used for feature extraction in computer vision, and some of the most widely used methods are discussed in the following sections.

a. Principal Component Analysis (PCA)

Principal Component Analysis (PCA) is a classical dimensionality reduction method which guarantees the minimum square error and gains linearly independent vectors as the basis of

subspace [122]. PCA projects data in an orthogonal subspace generated by the eigenvectors of the data covariance matrix [81]. It is a linear data transformation technique used to represent high-dimensional data [123]. The PCA algorithm removes redundant data while reserving useful features for future analysis [122].

Dimensionality reduction is achieved by transforming a large dataset into a smaller one while preserving the important information in the large dataset. Reducing the number of variables in the dataset affects accuracy. The idea is to find a perfect tradeoff between a good accuracy rate and simplicity because smaller datasets are easier and faster to explore and visualise.

The PCA algorithm consists of four steps: normalisation, covariance matrix computation, computing covariance matrix's eigenvalues and eigenvectors to identify principal components and getting the feature vector. Before extracting the principal components, it is important to normalise the range of the continuous initial variables to contribute equally to the analysis. Normalisation is important because the PCA algorithm is very sensitive to the variances of variables. If the variables' range is too big, the variables with larger ranges will dominate those with smaller ranges, leading to an unwanted bias. Normalisation solves this issue by transforming variables to the same scale.

The covariance matrix is computed to identify the correlation between variables. The sign of the covariance defines the type of correlation between variables. A positive sign means the two variables are correlated, which means they increase or decrease together. A negative sign means the two variables are inversely correlated, meaning one increases when the other decreases.

To identify the principal components, the eigenvectors and eigenvalues need to be computed. Eigenvectors and eigenvalues are linear algebra concepts computed from the covariance matrix to determine the principal components of the data. Principal components are variables constructed as linear combinations or mixtures of the initial variables. The principal component variables are uncorrelated but are made up of the initial variables, resulting in a dimensional reduction without loss of much information by discarding only the components with low information.

Principal components are structured in a way that the first principal component has the largest possible variance in the dataset, with the second principal component accounting for next highest variance and so on. This maximum variation direction-based approach guarantees the least information loss for feature selection [81]. The feature vector is derived from the eigenvectors of these principal components.

Eigenfaces is a popular object recognition technique based on PCA first used by Turk and Pentland [124]. The Eigenface algorithm projects face images onto a low dimensional feature space using PCA [125]. The eigenfaces algorithm uses PCA to generate a set of eigenvectors, known as eigenfaces, that form a basis in a reduced dimensionality feature space [124]. The eigenfaces algorithm is very sensitive to outliers.

The PCA algorithm requires images to be transformed into vectors first, a procedure that can lead to the curse of dimensionality [122]. Since large errors will dominate the mean square error, PCA is prone to the presence of outliers that are significantly far from the rest of the data points [123]. PCA is a global feature selection which misses the features contributing to the local characteristics of data, preventing subtle data local latent structure discovery [81]. Principal components may prove to be difficult to interpret since they are the linear combination of original samples [125]. PCA also underperforms when exposed to environments with angular lighting or when using a large database [126]. Modifications of the classical PCA have been developed to cater to these problems, such as the Two-Dimensional PCA (2DPCA) [122].

b. Independent Component Analysis (ICA)

Independent Component Analysis (ICA) is a famous dimensionality reduction algorithm. It is a data analysis tool based on high order characters [127]. ICA can estimate latent random vectors from observed data and the components of the vectors are independent. It is a statistical technique used for revealing hidden detail that underlies sets of random variables. PCA is usually run for preprocessing to decrease computation complexity before running ICA for feature extraction [122].

ICA is a blind source separation technique used to find the independent source signal from a non-Gaussian noisy signal [128]. It deals with maximisation of non-gaussian source signal using

higher-order parameters. ICA defines a generative model that finds underlying components from multivariate data with unknown mixing coefficient. The algorithm looks for components that are non-Gaussian and statistically independent [128]. ICA uses two assumptions: the observed signals are a mixture of independent source signals and the independent signals are statistically independent non-Gaussian, meaning their joint probability density function is the product of their individual probability density function [128].

ICA is more effective than PCA since the independent components provide more detailed local relationships than principal components. Variations of the ICA method exists, such as JADE, Fast ICA, and Constrained ICA (cICA) [129]. Like PCA, ICA also suffers from the curse of dimensionality [128].

c. Linear Discriminant Analysis (LDA)

Linear Discriminant Analysis (LDA) is a classical feature extraction and dimensionality reduction method widely used in classification problems [126]. LDA has been used in pattern recognition for subspace feature extraction and dimensionality reduction [130]. LDA performs better than the PCA algorithm in pattern classification by optimising the low-dimensional representation of the objects focusing on the most discriminant feature extraction [53].

The LDA algorithm introduces the concept of classes and distance between these classes [126]. The classes concept brings about the concept of intra-class and inter-class variance. The LDA algorithm aims to maximise the determinant of inter-class scatter matrix, minimising the determinant of the intra-class matrix [53]. The LDA algorithm finds the linear combination of features that best separates data from two or more classes [131]. The resulting composition gives a more condensed representation of the data, providing discriminative information that can be used for classification. The LDA algorithm projects data in a way that the projection best separates the data with the minimum least-squares error [130].

The Fisherface method, an improved method based on Fisher Linear Discriminant Analysis (FLDA), is an LDA method that performs dimensionality reduction using linear projection while preserving linear separability [126]. Fisherface is a class-specific method that aims to maximise the ratio of inter-class scatter and intra-class scatter matrix.

In classification problems, if the available class-specific data samples are limited and the dimensionality of each sample is large, the LDA suffers from the singularity problem [130]. The class separability criteria of the LDA method does not maximise the classification accuracy, which can lead to neighbour classes overlapping [126]. LDA methods require many training images taken from various viewpoints and lighting conditions to perform well, which is not practical in a real-life environment [53]. Many LDA variations such as Orthogonal LDA and regularized LDA have been developed to handle environment-specific problems encountered with the classic LDA.

In the next section we will discuss the classification module.

6.2.5. Classification

Classification is a very important technique in computer vision used to predict class labels from unknown data [74]. This is usually the end goal in computer vision, classifying an object. There are many algorithms in literature for object classification. Some of the most common classification methods are discussed below.

a. Euclidean Distance

Euclidean distance is a distance metric method that has been widely used in image classification. It is also called L2-norm [132]. Euclidean distance is the distance between two points. These two points can be in a different dimensional space. In one-dimensional space, the two points are on a straight line. In a two-dimensional space, each of the two points is made up of two coordinates. As the dimension increases, so do the coordinates on each point.

Mathematically, Euclidean distance is the square root of the sum of the squares of the differences between the points in each dimension [132]. Therefore, the Euclidean distance between two points cannot be negative since squares of real numbers are nonnegative.

Euclidean distance has had success in computer vision implementations and performs better than Manhattan distance [133]. However, the algorithm is not very robust and is sensitive to outliers [134].

b. Manhattan Distance

Manhattan distance, also called taxi-cab distance or city block distance, is a distance metric widely used in environments where using Euclidean distance is impractical. Such environments include two points in a city map, where calculating a straight line between two points may not be possible.

The Manhattan distance between two points is the sum of the absolute differences of their Cartesian coordinates [132]. The algorithm performs better than Euclidean distance in terms of speed and precision [135].

c. k-Nearest Neighbour

k-Nearest Neighbor (kNN) is a supervised machine learning algorithm used to solve classification, regression and searching problems. It was proposed by Cover and Hart in 1967 [74]. The kNN algorithm is a nonparametric lazy algorithm because it does not build a classification model like SVMs. It stores all the training samples until all the test samples are classified [136]. kNN is widely used in text classification and pattern recognition because of its simplicity and operation.

The k represents the number of nearest neighbours that are used to predict the class of the test sample [74]. It assumes that similar data points exist in proximity. Proximity is measured by a distance metric. This distance metric depends on the problem being solved but Euclidean distance is often the easiest and most preferred choice.

The kNN algorithm computes the most common class of k nearest neighbours to estimate the class of the test instance of the test set [74]. When a test sample is provided, the algorithm searches for the n-dimensional pattern space of the training data, finds the k training samples closest to the sample to be sorted by a certain distance measure, and the category is judged by the class that has the most nearest neighbours [136].

For the algorithm to achieve better classification results, it is important to select the right value of k [137]. When the value of k is small, there's small neighbors' inference and predictions are

less accurate. Inversely, as we increase the value of k, predictions become more and more stable and accurate, but the neighbours' interference is very large [137]. The majority voting principle is used to determine the class labels considering weighting the distances [138]. However, as the value of k increases, we will begin to see more errors.

The kNN algorithm is simple, efficient, and effective and is among the top ten classification algorithms of data mining [74]. The algorithm is only useful if the assumption that similar data points exist in proximity is true. The selection of k and the distance formula and the uneven distribution of samples have a strong impact on the classification accuracy of the algorithm [136].

d. Naïve Bayes

A Naïve Bayes classifier is a probabilistic machine learning model used for classification. Naïve Bayes is a classical statistical algorithm based on perfect Bayesian theory [139]. It is a very useful algorithm in data mining and machine learning [140]. The Naïve Bayes algorithm is widely used in text categorization, document judgment (such as spam filtering) and data stream classification. Naïve Bayes uses the weak assumption of conditional independence between features, meaning that the presence of one feature does not affect the other [140]. This is the reason this classifier is called naïve.

There are three common adaptations of the Naïve Bayes, namely Multinomial Naïve Bayes (MNB), Bernoulli Naïve Bayes (BNB) and Gaussian Naïve Bayes (GNB). Multinomial Naïve Bayes works on the concept of term frequency, that is, the number of times a word occurs in a document [141]. The classifier provides whether the word appears in a document or not and its frequency.

Bernoulli Naïve Bayes uses Boolean variables as features. In this case, the classifier checks whether a word occurs in a document or not but does not provide word frequency [141]. Gaussian Naïve Bayes uses continuous values that are not discrete as features and assumes the values are sampled from a Gaussian sample.

The Naïve Bayes algorithm is simple, efficient and has a very good performance on various domains [139]. It is a generative model-based classifier with a fast learning and testing process [140]. The Naïve Bayes method performs very poorly on datasets with a strong correlation between features due to the conditional independence assumption which is not always true in a real-world scenario [139].

e. Support Vector machines (SVMs)

Support Vector Machines (SVMs) is a machine learning algorithm originally developed by Vapnik for binary classification [142]. SVMs are widely used for solving both regression and classification problems as a quadratic optimization problem [143]. The algorithm was developed from statistics theory, which is based on the structural risk minimization principle [144]. The SVMs algorithm looks for the perfect tradeoff between model complexity and learning ability and has proven to be superior to the traditional methods in time series forecasting [145].

SVMs find a hyperplane in an N-dimensional space, where N is the number of features, that distinctly classifies the data points. SVMs' hyperplane separates data points of one class from data points of another class with a maximum margin [145]. Getting the maximum margin distances enables future data points to be classified efficiently.

A hyperplane is a decision boundary that helps classify the data points and a data point falling on a specific side of the hyperplane is attributed to that class. The dimension of the hyperplane is related to the number of features. If there are two features, then the hyperplane is a line, but if there are three features, the hyperplane becomes a two-dimensional plane. For n features, the hyperplane is a n-1 dimensional plane.

The SVMs algorithm is made up of support vectors which play a crucial role in the classification process of the algorithm. Support vectors are the data points that are closer to the hyperplane, which then influence the position and orientation of the hyperplane. The margin is defined by the distance of the hyperplane to the nearest data points of both classes [145]. A maximum margin among these support vectors defines our hyperplane and deleting a support vector will change the position of the hyperplane.

SVMs performs efficiently with high accuracies achieved using less computational resources. The algorithm performs effectively with small samples [144]. SVMs has high classification accuracy, robustness and shows indifference towards the input data type [146]. The algorithm overcomes the inherent drawback of a neural network, such as local minimum points, over-learning, and architecture [145]. SVMs can have different kernel functions such as linear, nonlinear, polynomial, Gaussian kernel, radial basis function and sigmoid.

f. Gradient Boosted Trees

Gradient boosting is a machine learning technique used in regression and classification problems. It produces a prediction model in the form of an ensemble of weak prediction models, typically decision trees. The gradient boosted trees algorithm is a combination of decision trees and the boosting technique as opposed to bagging [147].

Boosting is a method used to convert weak learners into strong learners. It is an ensemble technique to train basic classifiers in an iterative manner [147]. The boosting technique increases the sample weight of the previous decision tree misclassified, minimising the chances of the next decision tree classifying wrongly as well [148]. Because new predictors are learning from mistakes committed by previous predictors, it takes less time/iterations to reach close to actual predictions. With more trees built, less and less samples are misclassified, but the stopping criteria must be chosen carefully, or it could lead to overfitting on training data [148].

In the gradient boosted trees algorithm, the weak learners are decision trees. When a new tree is added, it fits on a modified version of the initial dataset. Since trees are added sequentially, boosting algorithms learn slowly. The accuracy of gradient boosting is improved iteratively, contributing to the anti-interference and generalisation abilities of the algorithm [147].

The gradient boosted trees algorithm is highly efficient on both classification and regression tasks. It can handle a mixed type of features and no pre-processing is needed. However, the algorithm require careful tuning of hyperparameters to perform efficiently and may overfit if too many trees are used. Also, the algorithm can be sensitive to outliers.

In our research, we used an advanced version of gradient boosting called XGBoost (eXtreme Gradient Boosting). XGBoost is faster, highly scalable, and more accurate than the gradient boosted trees algorithm [149]. XGBoost runs trees in parallel and uses regularisation to significantly reduce complexity [149].

g. Random Forests

Random forest is an ensemble algorithm based on decision trees and bagging [147]. It is a classification and regression algorithm that applies bagging and random feature selection methods [150]. If used for a classification problem, the result is based on majority vote of the results received from each decision tree. For regression, the prediction of a leaf node is the mean value of the target values in that leaf.

Random forests combines many trees in training data to produce high accuracy [150]. As the number of trees in a forest increase, the results get better. However, after some point, adding trees does not improve the model, but simply adds time complexity.

Random forests uses bagging to combine many decision trees to create an ensemble. Bagging, meaning combining in parallel, is a simple ensembling technique where many independent learners are built and combined using a model averaging technique. A random sub-sample of data is used per model to ensure models are dissimilar. Each model will have different observations based on the sampling process. The use of the bagging method and the random sub-sampling method ensures the overfitting problem of a single decision tree is avoided and the generalisation ability of the algorithm is improved [147].

Random forests reduces the risk of overfitting and accuracy is much higher than a single decision tree. The success of random forests highly depends on using uncorrelated decision trees. If similar trees are used, the overall result will not be much different than that of a single decision tree.

Like decision trees, random forests require neither normalization nor scaling and can handle different feature types together. They have high accuracy, robustness to noise and outliers and can be implemented in many problem domains [151].

In the next section we will discuss Deep Learning.

6.2.6. Deep Learning

Deep learning is a type of an Artificial Neural Network (ANN) consisting of multiple layers of neurons that are interconnected with varying weights and activation functions to learn the hidden relationship between input and output [152]. Deep learning has rapidly become one of the most popular research areas [153]. Several hierarchical layers in a deep learning model enable the extraction of important features from large datasets. Deep learning architectures such as deep neural networks, deep belief networks and recurrent neural networks have achieved success in pattern recognition, speech recognition, natural language processing, facial expression analysis, audio recognition, machine translation and object recognition [80]. Common deep learning methods in computer vision are described below.

A Convolutional Neural Network (CNN) is a multi-layered neural network developed to recognize visual patterns directly from pixel images without a considerable amount of processing. CNNs are a feed-forward multilayer perceptron trained in supervised mode using the gradient descent back-propagation algorithm that allows for the automatic extraction of features [154]. The algorithm is inspired by the hierarchical processing mechanism of information in the biological visual cortex channel where cells are only sensitive to the local regions of receptive fields [155].

CNNs apply the convolution operation on the local regions of an image to take advantage of the spatial relevance of the local pixels [154]. Less pre-processing is needed for CNNs compared to other classification algorithms [156]. CNNs show high degrees of invariance to displacement, scale, and deformation [157]. They have a high recognition rate and fast implementation speed and continue to make breakthroughs in numerous domains such as character recognition, face recognition, pedestrian recognition, robot navigation, sound recognition, image retrieval, object recognition and semantic segmentation [158].

CNNs are made up of four layers: convolution operation, activation function, pooling layer and the fully connected neural network layer [155]. The convolution layer is used for feature extraction and each convolution kernel can be regarded as a feature extractor [159]. Different

convolution kernels extract different features [155]. Each layer has numerous convolutional kernels used to convolve the image input [159]. The activation function is used to map the output of the convolutional layer to a non-linear function [155].

The pooling layer is used to perform feature selection process to reduce dimensionality while retaining important features [159]. Max, mean, and random pooling are the most commonly used approaches. The max-pooling method extracts the point with the largest value. The mean pooling function extracts the mean value, and the random pooling function extracts a value randomly from the given domain.

The fully-connected layer maps the features generated by the convolution layer into a fixed-length [155]. The general architecture of a CNN is made up of an input layer, numerous convolution layers, numerous pooling layers, numerous fully-connected layers and a single output layer [154].

CNNs have a local connection and weight sharing. The local connection is the non-fully connection of neurons between adjacent layers. This means that the input of each node in the convolution layer is only a small area of the upper layer of the neural network [155]. Weight sharing refers to how all parameters in each convolution kernel are shared by the entire graph without changing the weight coefficient in the convolution kernel due to the different positions in the image. In the following sections, we will discuss the different types of CNN algorithms.

a. LeNet

LeNet-5 is a 7-layer feed-forward convolutional neural network designed by LeCun et al. in 1998 to classify digits in the MNIST dataset [160]. Since then, it has been successfully implemented in handwriting recognition, face and speech recognition [158]. It is the first successful CNN and is based on the convolution operation [153]. LeNet is known for working well on digit classification tasks, such as recognizing handwritten digits [161]. It was applied in classifying digits and was used intensively by banks to recognize hand-written numbers on cheques.

LeNet is made up of 7 layers: an input layer, two convolution layers, two pooling layers, two fully connected layers and an output layer [161]. In the LeNet-5 architecture, sigmoid units have been implemented as activation function on the convolution layers, with sigmoid or linear units implemented in the output layer for classification or regression, respectively [154]. The softmax activation function is usually preferred in the output layer. The back-propagation algorithm is used to calculate the gradient used to calculate the weights of each layer. The back-propagation method is based on the gradient descent method [160]. The number of convolutional kernels is a key parameter in a LeNet architecture [159].

The LeNet algorithm is considered shallow compared to other CNN architectures [153]. To process high-resolution images, more convolution layers than are provided by LeNet are required. LeNet architecture is constrained by the availability of computing resources. Since features extracted by shallow CNNs are not rich, CNNs exhibit a low performance on some complex databases, rendering them undesirable [155].

b. AlexNet

AlexNet is one of the most studied CNN architectures and has achieved popularity due to its suitable tradeoffs between speed and accuracy [64]. AlexNet was developed in 2012 by Krizhevsky et al. with a very similar architecture to LeNet but deeper, bigger and more featured [80]. The major difference between AlexNet and LeNet architectures is the implementation of ReLU and dropout methods [162]. The AlexNet algorithm won the 2012 ImageNet Large Scale Visual Recognition Challenge (ILSVRC), thereby accelerating deep learning using CNNs [80]. Over the years, the AlexNet architecture has been improved upon to build state-of-the-art CNN architectures such as VGGNet, GoogLeNet and ResNet, with most of them having varying architectures and more convolution layers to extract more image details.

AlexNet is deeper and has more filters per layer and stacked convolutional layers. It consists of convolutions, ReLU, max pooling, cross channel normalization, dropout and ReLU activations. The ReLU activation strategy is used to handle the overfitting problem and the dropout strategy is used to speed up convergence [162].

AlexNet contains eight layers with weights and has 60 million training parameters. The first five layers consist of convolution operations, with the remaining three layers consisting of fully-connected neural networks and the last one being an output layer [163]. Each convolutional layer consists of the convolution operation and pooling feature recognition layers. Each fully connected layer connects every neuron of a layer to another layer and provides classification output using softmax function. The output of the last fully-connected layer is fed to a softmax producing a distribution of the class labels [80]. AlexNet can yield a 4096-dimensional feature vector per image, containing the activations of the hidden layer right before the output layer [64].

c. VGGNet

The Visual Geometry Group Network (VGGNet) was developed in 2014 by Simonyan and Zisserman [164]. The VGGNet architecture was developed to demonstrate that architecture with very small filters (3x3) can be trained to increasingly higher depths (up to 19 layers) and achieve state-of-the-art classification on the ImageNet Large Scale Visual Recognition Challenge (ILSVRC). It is similar to AlexNet, with only 3x3 convolutions and many filters. It is arguably the most preferred choice in the ANN community for extracting features from images. The VGGNet method is widely used for image classification and has proven to be very effective [63].

VGGNet is made up of 13 convolutional layers and three fully connected layers and has a very uniform architecture [165]. The algorithm has 6 network models with depths ranging from 11 to 19 layers [155]. Among these layers, the deepest two sets of 16-19 layers offer the best classification and location performance. The entire structure is made up of 5 convolutions, with a 3x3 convolution kernel. A max-pooling layer follows each convolution operation. After the last pooling layer, three fully-connected layers are connected to integrate the features in the feature map. The last layer in the fully-connected network is the softmax layer used for classification. Dropout is introduced to solve for overfitting and regulation is added to the loss function to constrain the weight parameter of the fully connected layer [165].

The VGGNet method reduces the size of the convolution kernel and pool kernel [155]. It increases the number of convolution layers and uses pre-trained data to initialize parameters, speeding up convergence and improving accuracy. This results in a big performance gain

compared to VGGNet's predecessors. The use of small kernels brings more detailed capturing of information, with the max-pooling method bringing greater local information difference [155]. The depth of the algorithm plays a huge role in the performance of the algorithm.

However, although the VGGNet algorithm is widely used in numerous applications, it is by far the most expensive architecture due to its high computational requirements and a large number of parameters used [166]. VGGNet uses 138 million parameters when extracting features, which is very computationally expensive. When the number of layers is increased, the accuracy gets saturated and rapidly reduces, a phenomenon known as gradient fading [167]. This degradation is not caused by overfitting, it has been proven that adding more layers to a deep learning model leads to higher training error [168].

d. GoogLeNet

The GoogLeNet architecture, also known as Inception, is based on the LeNet architecture with a novel element called the inception module. It is one of the most commonly implemented structures due to its higher accuracy compared to other architectures [157]. It won the 2014 ImageNet Large Scale Visual Recognition Challenge (ILSVRC) using almost 12 times fewer parameters than AlexNet and achieving a considerably higher accuracy rate [80]. It achieved an accuracy rate of 93.33% classifying common daily objects [169]. GoogLeNet uses only 5 million parameters and performs slightly better than VGGNet [166].

GoogLeNet uses 1x1 convolutions to reduce dimensionality and increase the depth and width of the network without affecting performance [157]. The GoogLeNet architecture is made up of 22 layers, which include 9 inception modules and 3 softmax layers [169]. The resulting architecture uses 4 million parameters, reduced from 60 million. GoogLeNet uses batch normalization, image distortions and RMSprop concepts.

The inception module is based on numerous tiny convolutions implemented to vastly reduce the number of parameters. It is implemented to increase the network depth and width through processing data through several convolutions simultaneously [169]. The Inception module is intended to find the optimal local construction and repeat it spatially [170].

A single module in the inception module has four 1x1 convolutions, one 3x3 convolution, one 5x5 convolution and one 3x3 max-pooling [169]. Data from the previous layer is divided into four batches and concatenated into one batch upon exiting the inception module. Implementing the inception module shortens the network's training time while increasing computer resources utilization. It enables the extraction of local feature representation using flexible convolutional kernel filter sizes with layer-by-layer structure, which has shown to be robust and effective when dealing with high-resolution images [170].The result is an architecture that allows an increment of units at each stage without suffering from high computational complexity. The inception module allows the entire GoogLeNet architecture to be very deep and efficient when training.

e. ResNet

The Residual Neural Network (ResNet) architecture was developed in 2015 by the Microsoft organization, containing 152 hidden layers [152]. ResNet introduces a novel architecture which features skip connections and heavy batch normalization. The main reason to develop ResNet was to ease network training [171].

While other CNN architectures rely on unreferenced functions for learning, ResNet explicitly defines the layer to learn the residual function as the reference input layer [171]. The residual function allows the network to increase depth while making the network easier to optimize. The ResNet architecture improves on other CNN architectures because other CNN architectures cannot be trained optimally when they are deep. ResNet can increase accuracy and reduce training error by significantly increasing the depth of the network [168].

The skip connections are also known as gated units or gated recurrent units. They are similar elements to the ones applied in Recurrent Neural Networks (RNNs). These skip connections allow the algorithm to train a neural network with 152 layers exhibiting a lower complexity than the VGGNet architecture [168]. ResNet consists of small convolution filters of 1x1 and 3x3, making the architecture very simple [172]. These convolution layers reduce complexity and help extract high-level feature maps.

6.3 Conclusion

A model provides the blueprint to answering a research question. A model is a representation or abstraction of an artefact showing interrelationships of an action and reaction [91]. In our case, it provides the methods and algorithms that can be used to answer our research questions.

In this chapter we provided methods and algorithms that we will pick from to solve our research problem based on our literature review. We provided potential solutions for both the traditional biometrics pipeline and the artificial neural networks pipeline. Some of these methods will be used for our research prototype to find a model that can recognise inebriation in humans using computer vision.

By completing our model chapter, we have met objective 2 of our research. In the next chapter, we will discuss the benchmark for our research. We will aim to discuss the functional and non-functional requirements of our research and how we will measure the success or failure of our research in answering our research question.

Chapter 7 Benchmark

7.1 Introduction

In the previous chapters we discussed the research problem statement and environment. We also discussed similar systems and provided a model for our solution to detect inebriation in humans using computer vision. A research prototype will be developed based on our research model. In this chapter we will discuss our research benchmark which will be used to analyse our prototype against.

To determine the performance of our research, it is important to have a benchmark to measure against. A benchmark consists of the functional and non-functional requirements used to measure the success of the research. Functional requirements are a description of the service the system must offer. Non-functional requirements define how the system must perform the functional requirements of the system. These requirements are used to evaluate the performance of our research prototype.

In this chapter, we will discuss the functional and non-functional requirements that are relevant to our research. In Section 2, we discuss functional requirements for our research and how they will be measured. In Section 3, we discuss the non-functional requirements of our research and how they are measured.

7.2 Functional Requirements

Functional requirements describe the type of service the system must offer [173]. Functional requirements answer the question “what does the system do?” [174]. They represent the structure and constraints of the system. A function is the input, the system’s behaviour, and the output. Functional requirements are features that allow the system to perform as intended. If these functional requirements are not met, the system will not work. In this regard, functional requirements are the product features developed based on user requirements. In this section we will discuss the functional requirements of our research and how they can be measured.

7.2.1. Preprocessing

Preprocessing is used to correct problems that might have risen while capturing our input. These problems might vary, from sensor or lighting issues to dead pixels, shadows, and uneven lighting [72]. Preprocessing is also used for noise correction to reduce the effects of noisy input. Image rotation can also be done at this stage for images that are rotated or out of perspective. Preprocessing can also be used to redistribute color saturation or correct image illumination. In our research, we also use preprocessing for image enhancement to reduce the lowest gray values to black and the highest to white [94].

Preprocessing is implemented to readily captured input for feature extraction. For our research, preprocessing is considered successful when the input image has been denoised and is in its optimal state for localisation to take place.

7.2.2. Localisation

Object recognition is used to draw a bounding box around the object of interest. For our research, object recognition is used to draw bounding boxes around face images. If there is more than one face in an image, the bounding boxes are drawn on all faces. In our research, localisation is successful when all faces in an image are detected correctly and bounding boxes are drawn around them. Localisation is unsuccessful if no faces are detected in an image.

7.2.3. Feature Extraction

After face localisation, it is important to extract the face features we need for classification purposes. Feature extraction is used to extract the most distinct features that can be used for classification. It is used to extract only the features of interest while disregarding the rest of the features. This reduces both computation and space complexity. In our research, feature extraction is successful if distinct features are extracted from the localised face image and can be used to develop a feature space for classification.

7.2.4. Classification

Classification is a technique used to predict class labels from unknown data [74]. Classifying an object is the end goal of our research. The feature space developed during feature extraction is used for inebriation classification. Classification is considered successful if inebriated faces are classified as inebriated and sober faces are classified as sober.

In this section we described the functional requirements of our research. These modules make up our inebriation recognition system. In the next section we will discuss the non-functional requirements of our research.

7.3 Non-Functional Requirements

Non-functional requirements define how the system's functional requirements are to be achieved [173]. Non-functional requirements answer the question "how does the system perform" [174]. Non-functional requirements define the system behaviour and features that affect user experience. They are the product properties that focus on user expectation. Defining and meeting non-functional requirements effectively leads to great system usability. In this regard, non-functional requirements can be used to judge a system's performance. In this section, we will discuss the non-functional requirements of our research and how they can be measured. These non-functional requirements will be used to measure the performance of our prototype.

7.3.1. Confusion Matrix

The confusion matrix is a table used to describe the performance of a classification model. In a binary classification problem, the confusion matrix consists of 4 important figures, namely: Truth Positive (TP), False Positive (FP), Truth Negative (TN) and False Negative (FN). In a multi class classification problem, there are more than four figures. A multiclass confusion matrix is beyond the scope of this research and will not be discussed here. In the context of our research, TP is the inebriated faces correctly classified as inebriated, TN is the sober faces correctly classified as sober, FP is the sober faces incorrectly classified as inebriated and FN is the inebriated faces incorrectly classified as sober. These components form the basis for the other metrics described below and will be used in our research as a basis for our statistical evaluations.

7.3.2. Accuracy

Classification accuracy is the most intuitive metric used to evaluate a classification model's performance. It is a ratio of correctly predicted observations to the total observations. Accuracy has the following definition:

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (1)$$

Accuracy works well if there are equal number of samples belonging to each class. Accuracy alone does not tell the full story about the performance of the model, especially when working with a class-imbalanced data set where there is a disparity between the number of positive and negative labels. In a class-imbalanced data set scenario, classification accuracy may give us the false sense of achieving high accuracy. The real problem arises when the cost of miscalculation of the minor class samples are very high. For example, when dealing with cancer classification, the cost of misdiagnosing a sick patient is much higher than the cost of sending a healthy person to more tests.

In our research, the accuracy metric is very important. Our dataset is relatively balanced and there is no disparity between the number of positive and negative cases. Our research can be used in various domains, and we believe the cost of miscalculation on both classes have equal weight. That is, the cost of classifying an inebriated individual as sober is equal to classifying a sober individual as inebriated. In this case, the accuracy value is very important in determining the overall performance of our research.

7.3.3. Precision

Precision attempts to answer the question: what proportion of positive predictions was correct? It is defined as:

$$Precision = \frac{TP}{TP+FP} \quad (2)$$

High precision relates to the low FP rate. Precision is a great measure to determine when the cost of FP is high. For example, in spam classification, a high FP results in non-spam emails being classified as spam, resulting in the user losing very important emails. In our research, we want the precision value to be as high as possible.

7.3.4. Recall/Sensitivity

Recall attempts to answer the following question: what proportion of positives was predicted correctly? It is defined as:

$$Recall = \frac{TP}{TP+FN} \quad (3)$$

High recall relates to low FN rate. Recall is an important metric when there is a high cost associated with FN. For example, in cancer classification, high FN can result in sick patients being misdiagnosed as healthy with fatal consequences. In our research, we want the recall value to be as high as possible.

7.3.5. F1-Score

F1 is a function of precision and recall. It is the Harmonic Mean between precision and recall used to measure a test's accuracy. It is defined as:

$$F1 = 2 * \frac{precision*recall}{precision+recall} \quad (4)$$

The F1 score aims to find the balance between precision and recall by giving equal weight to both measurements. It is a very powerful measurement when seeking a balance between precision and recall and there is uneven class distribution. It is also used to find an optimal blend of precision and recall.

F1 score's value ranges $[0; 1]$ and it tells us how precise a classifier is based on the number of instances classified correctly. High precision but lower recall results in an extremely accurate model that misses many instances that are difficult to classify. It uses the harmonic mean instead

of the arithmetic mean because the harmonic mean punishes extreme values more. The greater the F1 score, the better the performance of our model.

The F1 score is the most important metric in our model. Our research aims to develop a balanced classification model with the optimal balance of recall and precision. The F1 score punishes high recall and low precision. It also punishes low recall and high precision. A higher F1 score represents a higher precision and recall. A model with a higher F1 score means it has a low FN and FP.

7.3.6. Area Under the Curve

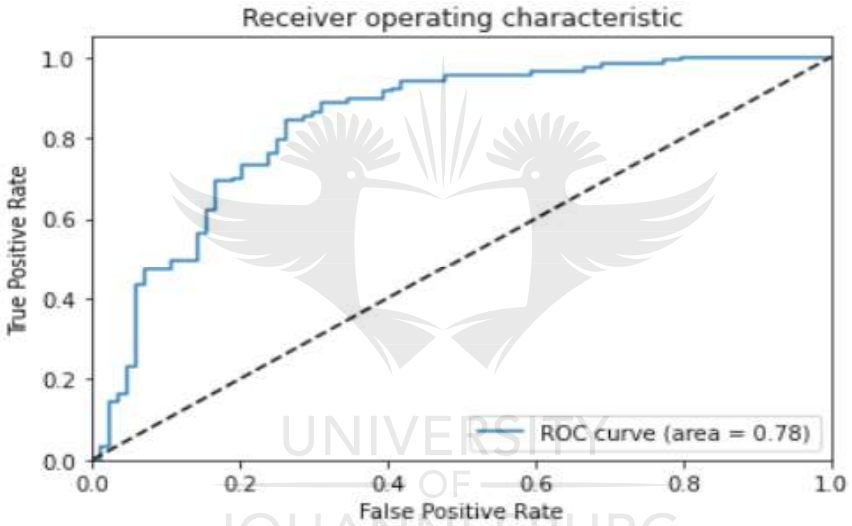

Figure 7: A ROC Curve showing the AUC.

A receiver operating characteristic (ROC) curve is a two-dimensional curve traced out by pairs of FP and TP based on various decision threshold settings [175]. It is commonly used to visualise the performance of a binary classifier. The area under the curve is the area under the ROC curve of a model. It is computed as a figure of merit to summarize a diagnostic system's performance [176].

Area under the curve (AUC) is one of the most common metrics used for model evaluation in binary classification problems. The area under the curve of a classifier is the probability that the classifier will rank randomly chosen positive example higher than a randomly chosen negative

example. The AUC is used to measure a binary classifier's overall accuracy [177]. The AUC has a range of $[0; 1]$, and the greater the value, the better the model's performance. We will use the AUC in our research to calculate the performance of our model in detecting inebriation. The metrics discussed in this chapter are also used in literature to compare methods too.

7.4 Conclusion

Benchmarking is used to provide a barometer for measuring the performance of our research. A research benchmark is made up of functional and non-functional requirements. Functional requirements are the services the system must offer, and non-functional requirements define how the system must perform these services. Providing requirements enables us to measure the performance of our research. In this chapter we discussed both functional and non-functional requirements of our research and how to measure the success/failure of each requirement.

Describing our benchmark allows us to measure the performance of our research. We can measure both the functional and non-functional requirements of our research. Describing how the system must perform beforehand simplifies the prototype and results analysis phases of our research.

By completing this chapter, we have met research objective 3. In the following chapter, we will discuss our prototype. We will describe different pipelines we used in our research and how to recreate the prototype. The benchmark will be used against the prototype to measure our research's performance in the results chapter.

Chapter 8 Prototype

8.1 Introduction

In the previous chapter we discussed the benchmark to measure our research against in the benchmark chapter. This chapter begins the implementation phase of our research, where we will develop our prototype. In this chapter we will cover the implementation details of our research. We will provide the different pipelines we used to detect inebriation in humans. The method details are provided in the Model chapter.

In section 2 we will describe the platform we used for our prototype. In section 3 and 4 we will provide the implementation details of our traditional biometrics pipeline, namely Local Binary Patterns and Histogram of Gradients, respectively. In section 5 and 6 we will discuss the deep neural networks pipelines, namely You Only Look Once (YOLO) and Faster R-CNN, respectively. In section 7 we will briefly cover the optimisation and validation methods we used on our prototype, namely hyperparameter-tuning and k-fold cross-validation.

8.2 Platform

Our prototype was developed in the Python programming language, version 3.8. Google Colab was used as the integrated development environment and platform.

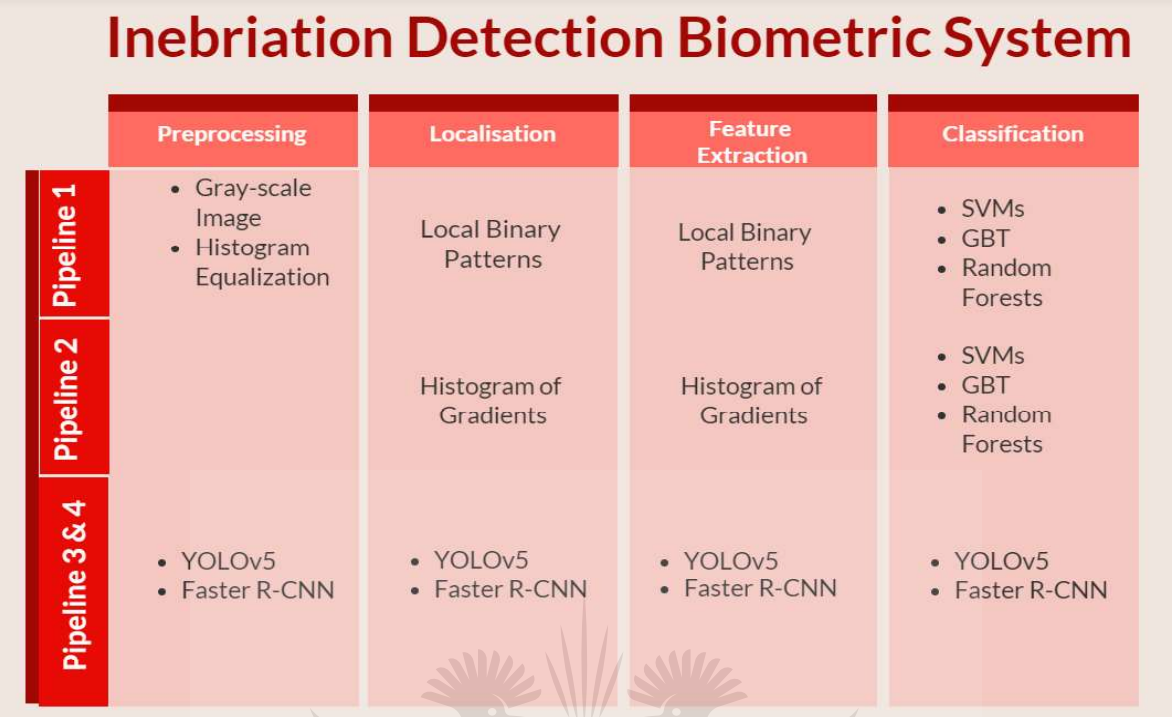

Figure 8: The Inebriation Recognition System Overview

8.3 Pipeline 1 (Local Binary Patterns)

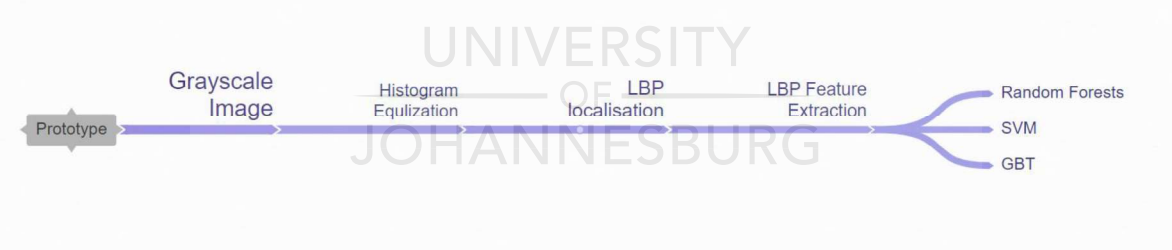

Figure 9: The Local Binary Patterns Pipeline.

In this section, we will provide the implementation details for the LBP-based method.

8.3.1. Local Binary Patterns-Support Vector Machines (LBP-SVMs)

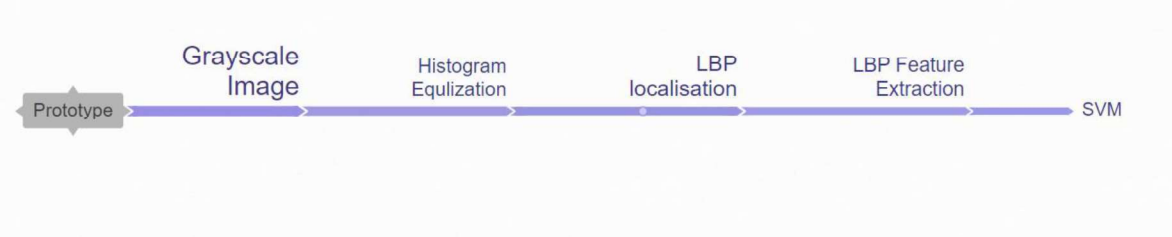

Figure 10: LBP with SVM Classifier Pipeline.

In this section, we will discuss how we implemented our pipeline. We used grayscale and histogram equalisation methods for preprocessing. We then used local binary patterns for both localisation and feature extraction. Images are resized after localisation. We used support vector machines (SVMs) to classify inebriation.

a. Grayscale

The first step in our preprocessing stage is to grayscale the image. We grayscale an image to minimise the noise. We used the *opencv* library to grayscale images. Firstly, we used the opencv's method below to read an image,

$$image = cv2.imread(path),$$

where *path* is the image path. After reading the image, we used opencv's method below to convert our image to grayscale,

$$gray = cv2.cvtColor(image, cv2.COLOR\_BGR2GRAY),$$

where *image* is the image to be converted to grayscale and a constant *cv2.COLOR_BGR2GRAY* tells the algorithm the type of image conversion to make. The grayscale method used is the weighted grayscale. The resulting grayscale image is used for histogram equalisation.

b. Contrast Limited Adaptive Histogram Equaliser

The second step in our preprocessing stage is to perform histogram equalisation on our image. Histogram equalisation is a technique for adjusting image intensities to enhance contrast, allowing areas of lower local contrast to gain a higher contrast. Histogram equalisation enhances features for extraction.

For histogram equalisation, we used opencv's method below,

$$clahe = cv2.createCLAHE(clipLimit = 3.0, tileGridSize = (8,8)),$$

where *clipLimit* is the contrast limit for localised changes in contrast and *tileGridSize* is the size of a grid for local histogram equalisation. CLAHE differs from the normal HE because it computes several histograms using image grids and uses them to redistribute the contrast of the entire image. To avoid over-amplifying the noise during the redistribution, a *clipLimit* is used to limit the amplification. *tileGridSize* is used to define the size of each grid. The image will be divided into equally sized rectangular tiles, with the *tileGridSize* value defining the number of tiles per grid in rows and columns.

We chose *clipLimit* to be of value 3 because it performs well. Values between 2 and 3 have proven to work well, while very large values fail to control the histogram equalisation, resulting in an unwanted maximal local contrast. The value 1 will result in the original image. We chose *tileGridSize* to be of value (8,8), which is the default value for *opencv*.

After initialising CLAHE, we applied it to the image as follows,

$$clahe.appy(image),$$

where *image* is the image going through HE. After performing CLAHE, we use the image for localisation.

c. Local Binary Patterns Localisation

After performing preprocessing, the next step is to localise our face image. To achieve this, we used opencv's cascading classifiers using LBP-based models for face recognition. *Opencv* offers a pretrained model for LBP-based face recognition. To load the pre-trained model, we used the opencv method below

$$cascade = cv2.CascadeClassifier(path),$$

where *path* is the path to the pretrained model file. To detect the face images, we used the LBP features instead of the Haar-like features because it has shown to perform faster under limited resources and has a high recognition accuracy [178]. The LBP *opencv* method below was used,

$$faces = cascade.detectMultiScale(image, scaleFactor = 1.1, minNeighbors = 2),$$

where *image* is our input image, *scaleFactor* specifies how much the image size is reduced at each image scale, *minNeighbors* specifies how many neighbours required for each candidate rectangle to be retained. The method also has parameters we did not use: *minSize* which is the minimum possible size for an object to be considered and *maxSize* which is the maximum possible size for an object to be considered. We used the default values of these parameters.

The scale factor specifies how the image size is modified per image scale. A small scale factor results in a slower, more precise object recognition whereas a large scale factor may result in the recognition algorithm missing some objects. We chose our *scaleFactor* to 1.1 because it is the default value on opencv and it performs well. This means we are reducing the image by 10% of its size at each iteration. *scaleFactor* is used to rescale the image, allowing previously larger faces to become smaller and detectable by the algorithm.

We chose *minNeighbors* to be 2 because we believe it is a good value to use for face recognition. This parameter determines the quality of the detected faces, with 2-6 performing well. Higher values result in less recognitions with good quality, whilst low values result in more recognitions of poor quality.

After detecting our face images, we use the detected faces' coordinates and crop them from the original color images as opposed to the histogram equalised images. We do this for feature extraction because there are more features with color images for LBP than with grayscale images. Also, we discard detected faces of sizes smaller than (128, 128). For the remaining face images, we resize them using the below opencv method,

$$resize = cv2.resize(face, (128, 128)),$$

where *face* is the detected face image and *(128, 128)* is the size to resize image to. We resize the detected face images because the feature extraction algorithm requires uniform feature space to effectively extract features that can be used for classification. We chose the size *(128, 128)* because it is a relatively good size for face images and provides enough detail for feature extraction.

d. Local Binary Patterns Feature Extraction

For feature extraction, we took the resized *(128, 128)* face image and split it into 16 *(32,32)* blocks. We then used *scikit-image* library's local binary pattern method below,

$$lbp = feature.local\_binary\_pattern(image\_block, nPoints = 24, radius = 1, method = "uniform"),$$

where *image_block* is the image block, *nPoints* is the number of sample points to use to build the circular local binary pattern, *radius* is the radius of the circle and *method* is the variant of the LBP method we are using.

We chose a big *nPoints* value of 24 because it worked well for our solution, a lot better than the usually recommended *8*radius.* We chose a small *radius* of 1 because after dividing the image into blocks, a small radius gets more features from the relatively small-sized block than a larger radius. We used the uniform *method* because it improves rotation invariance while reducing the size of the feature vector.

We created a histogram from the extracted features and normalised it. We normalised it to provide uniformity in terms of the range of values. This improves our classifier performance.

After normalising the histograms for each image block, we concatenated them. The concatenated normalised histograms of our image blocks become our feature space for classification. We used *scikit-learn*'s *train-test-split* method to randomly split our feature set into two groups: the training set and the test set. 80% of the data becomes our training set and we used the remaining 20% for testing our method. K-Fold cross-validation is performed on the training set.

e. Support Vector Machines

For classification, we used the *scikit-learn*'s support vector machines library. We used 4 parameters for hyperparameter-tuning, namely:

C: range [0.1; 1; 10; 100]

gamma: range [1; 0.1; 0.01; 0.001]

kernel: range ['rbf'; 'poly'; 'sigmoid'; 'linear']

probability: range [True]

C is the regulation parameter, which is a tradeoff between correct classification against maximisation of the decision's margin. For larger values of C, a smaller margin is accepted if the function is classification accuracy is high. A lower value of C encourages a larger margin at the cost of training accuracy.

Gamma defines the influence of a single training example. Low values result in a very influential single training example, with a high gamma value being the opposite. We chose common gamma values that are widely used. The kernel is the function used by the algorithm to take data as input and transform it into required form. The redial basis function (RBF) is the most used because it has finite response along the entire x-axis. We chose all the functions that are offered by the algorithm as options. The probability value determines if our method will enable probability estimates.

After getting our best model, we fit it using the method below,

$$search.fit(X, y\_train),$$

where X is the dataset to be classified and y_train is the class each sample belong to. After training, we test our SVM model against unseen data. The predictions made are then evaluated.

8.3.2. Local Binary Patterns-Gradient Boosted Trees (LBP-GBT)

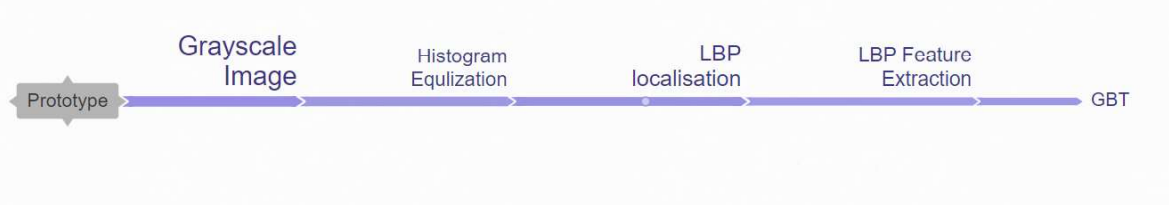

Figure 11: LBP with Gradient Boosted Trees (GBT) Classifier Pipeline.

In this pipeline, we grayscaled the image, performed the histogram equalisation and extracted the LBP features the same way as in the SVM pipeline. The only difference is the use of the gradient boosted trees classifier as opposed to support vector machines.

For classification, we used the *xgboost*'s eXtreme Gradient Boosting library as below,

$$xgb\_model = xgb.XGBClassifier(objective = "binary: logistic", random\_state = 42),$$

where *objective* is set to a binary classifier as opposed to linear regression or a multi class classifier and *random_state* is a pseudo-random number generator state used for random number sampling.

We used 11 parameters for hyperparameter-tuning, namely:

eta: uniform(0.01; 0.4)

min_child_weight: randint (0; 0.3)

max_depth: randint(3; 10)

colsample_bytree: uniform(0.3;10)

gamma: uniform(0;1.5)

subsample: uniform(0.5;1.5)

lambda: uniform(0;2.0)

alpha: uniform(0;2.0)

learning_rate: uniform(0.03;0.3)

n_estimators: randint(100, 150),

where *uniform* returns a random floating number in the provided range and *randint* returns a random integer from the provided range. *Eta* is the learning rate used to prevent overfitting, set to 0.3 by default. *min_child_weight* is the sum of instance weight needed in a child. *max_depth* is the maximum depth, which will likely result in overfitting if set too high and is set to 6 by default. *colsample_bytree* is the subsample ratio of columns when constructing a tree and is set to 1 by default. gamma is the minimum loss reduction required to make a further partition on a leaf node of the tree and is set to 0 by default. *subsample* is the subsample ratio of the training instances which occurs once in every boosting iteration. It is set to 1 by default. *lambda* is the L2 regularisation term on weights and is set to 1 by default. *alpha* is the L1 regulation on weights and is set to 0 by default.

After getting our best model, we fit it using the method below,

$$search.fit(X, y\_train),$$

where *X* is the dataset to be classified and *y_train* is the class each sample belong to. After training, we test our SVM model against unseen data. The predictions made are then evaluated.

8.3.3. Local Binary Patterns-Random Forests

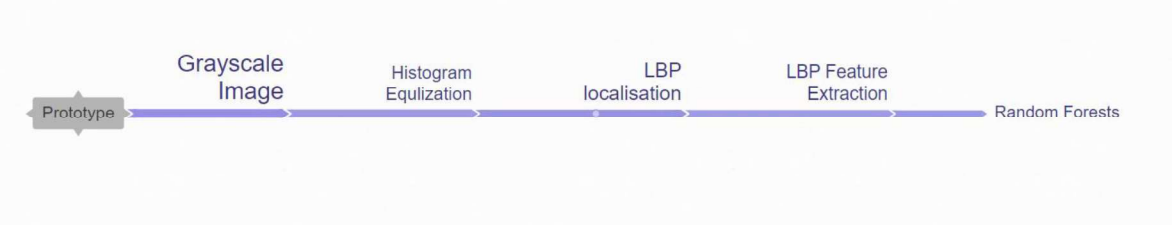

Figure 12: LBP with Random Forests Classifier Pipeline.

In this pipeline, we grayscaled the image, performed the histogram equalisation and extracted the LBP features the same way as in the SVM pipeline. The only difference is the use of the random forests classifier as opposed to support vector machines.

For classification, we used the *scikit-learn*'s RandomForestClassifier library. We used the below method to create our classifier,

$$random\_forest = RandomForestClassifier(random\_state = 42),$$

where *random_state* is a pseudo-random number generator state used for random number sampling.

We used 6 parameters for hyperparameter-tuning, namely:

bootstrap: random(True; False)

max_depth: random(10; 20; 30; 40; 50; 60; 70; 80; 90; 100)

max_features: random('auto';'sqrt')

min_samples_leaf: random(1;2;4)

min_samples_split: random(2;5;10)

n_estimators: random(200; 400; 600; 800; 1000; 1200; 1400; 1600; 1800; 2000)

where *random* returns a random value from the provided options. *bootstrap* determines whether bootstrap (subset) samples are used when building trees. If set to *False,* the whole dataset is used to build the tree. *max_depth* is the maximum depth of the tree, set to *None* by default, which expands nodes until all leaves are pure. *max_features* is the number of features to consider when looking for the best split, set to '*auto'* by default. *min_samples_leaf* is the minimum number of samples required to be at a leaf node, set to 1 by default. *min_samples_split* is the minimum number of samples required to split an internal node, set to 2 by default. *n_estimators* is the number of trees in a forest.

After getting our best model, we fit it using the method below,

$$search.fit(X, y\_train),$$

where X is the dataset to be classified and y_train is the class each sample belong to. After training, we test our SVM model against unseen data. The predictions made are then evaluated.

8.4 Pipeline 2 (Histogram of Gradients)

Figure 13: The Histogram of Gradients Pipeline.

In this section, we provide implementation details for the histogram of gradients method.

8.4.1. Histogram of Gradients-Supporter Vector Machines (HOG-SVMs)

Figure 14: Histogram of Gradients with Support Vector Machines pipeline.

In this section, we will discuss how we implemented our pipeline. We used histogram of gradients for both localisation and feature extraction. We used support vector machines to classify inebriation.

a. Histogram of Gradients Localisation

We did not perform preprocessing because we believed HOG performs better with RGB images as opposed to grayscaled images. For localisation, we used *dlib*'s HOG face recognition method. We used the method below to load our face detector,

$$hog\_face\_detector = dlib.get\_frontal\_face\_detector()$$

This parameterless method returns an object detector configured to detect human faces that are looking towards the camera. To perform face recognition, we used the method below,

$$face\_images = hog\_face\_detector(image, 1),$$

where *image* is our input image to detect faces from and *1* is the number of times the algorithm should upsample the image. Upsampling the image helps the algorithm detect smaller faces. *1* is the default value that has proven to work for most cases, and it works very well for our problem.

After detecting our face images, we use the detected faces' coordinates and crop them from the original image. We discard detected faces of sizes smaller than (128, 128) because they are not

optimal for feature extraction and classification purposes. For the remaining face images, we resize them using the below opencv method,

$$resize = cv2.resize(face, (128, 128)),$$

where *face* is the detected face image and *(128, 128)* is the size to resize image to. We resize to this size because HOG feature extractor works best with images of this size.

b. Histogram of Gradients Feature Extraction

For feature extraction, we used the resized *(128, 128)* face image. We used *scikit-image* library's hog method for feature extraction below,

$$feature\_vector, hog\_image = hog(image, orientations = 9, pixels\_per\_cell = (8,8), cells\_per\_block = (2,2), visualize = True, multichannel = True),$$

where *image* is the input image, *pixels_per_cell* is the size of a cell in pixels, *cells_per_block* is the number of cells in a block, *visualise* is the option to return an image of the HOG with line segments centered at the cell center of each cell and orientation bin and *multichannel* is the option to consider the last dimension as a color channel if *True* or as spatial if *False*. *feature_vector* is a 1-dimensional array containing the HOG descriptor for the image and *hog_image* is a visualisation of the HOG image provided if *visualize* is set to *True.* We used the parameter values based on a mixture of our trial-and-error and popular values used by the *dlib* libary.

We used *scikit-learn*'s *train-test-split* method to randomly split our feature set into two groups: the training set and the test set. 80% of the data becomes our training set and we used the remaining 20% for testing our method. K-Fold validation is implemented on the training set.

These features are then used for classification purposes. For classification, we implemented the support vector machines classifier. The implementation details for the SVM algorithm are discussed in Section 8.3.1.

8.4.2. Histogram of Gradients-Gradients Boosted Trees (HOG-GBT)

Figure 15: HOG with Gradient Boosted Trees Classifier Pipeline.

In this section, we will discuss how we implemented our pipeline. We used histogram of gradients for both localisation and feature extraction. We used gradient boosted trees' XGBoost method to classify inebriation.

We implemented HOG localisation and HOG feature extraction the same way as in HOG-SVM pipeline and the implementation details of these algorithms is discussed in section 8.4.1. For classification, we used XGBoost. The implementation details for the XGBoost algorithm are discussed in Section 8.3.2.

8.4.3. Histogram of Gradients-Random Forests

Figure 16: HOG with Random Forests Classifier Pipeline.

In this section, we will discuss how we implemented our pipeline. We used histogram of gradients for both localisation and feature extraction. We used the random forests classifier to classify inebriation.

We implemented HOG localisation and HOG feature extraction the same way as in HOG-SVM pipeline and the implementation details of these algorithms is discussed in section 8.4.1 and. For classification, we used random forests. The implementation details for the random forests algorithm are discussed in Section 8.3.3.

8.5 Pipeline 3 (YOLO)

In this section, we will discuss the YOLOv5 implementation. We used Roboflow's YOLOv5 implementation based on Ultralytics Pytorch Framework's implementation, publicly accessible on Ultralytics' github repository since May 2020. We used YOLOv5 because it is the latest state-of-the-art object recognition algorithm that has proven to be very intuitive with faster inferences. We chose YOLOv5s because it is the smallest, fastest base model of YOLOv5. The Google Colab environment was used for training because it accelerates training time and has more memory.

8.5.1. Model Configuration

YOLOv5 is a single-stage object detector consisting of 3 important parts: model backbone, model neck and model head. The model backbone is used to extract important features from an input image. Our YOLOv5 implementation uses Cross Stage Partial Networks (CSP) as a model backbone, which has improved processing time. This allows for faster extraction of features from our input image. CSP scales both up and down and is applicable to both small and large networks while maintaining optimal speed and accuracy [179].

Model neck is used to generate feature pyramids, which help models generalise well on image scaling. This helps models identify the same object with different sizes and scales. Models perform well on unseen data as a result. Our YOLOv5 implementation uses path aggregation network (PANet) as a model neck. PANet aims at boosting information flow on proposal-based instance segmentation framework by enhancing the entire feature hierarchy with accurate localisation signals in lower layers by bottom-up path augmentation, shortening the information path between lower layers and topmost feature [180].

Model head is used for final object recognition by applying anchor boxes on features and generating final output vectors with class probabilities, objectness scores and bounding boxes. Our YOLOv5s model is made up of 191 layers, 7.5 million parameters and 7.4 million gradients.

8.5.2. Activation Function

The choice of activation function is very important in a deep neural network. In YOLOv5, Leaky ReLU (Rectifier Linear Unit) and sigmoid activation functions are used. ReLU as an activation function returns 0 for any negative input and returns the actual value of any positive value. Leaky ReLU differs from ReLU because instead of converting a negative value to zero, it will convert it to a very small value such as 0.01. Sigmoid activation function maps the whole real range of z into [0,1] in the $g(z)$. Leaky ReLU activation function is used in middle/hidden layers and the sigmoid function is used in the final recognition layer.

8.5.3. Learning Optimiser

For optimisation, the YOLOv5 architecture has two options: Stochastic Gradient Descent (SGD) and Adam. SGD is an optimisation algorithm that computes the gradient of the network loss function with respect to each individual weight in the network. Adam is an adaptive learning rate optimiser that has shown to optimise less efficiently than SGD [181]. Adam's parameter updates are invariant to the gradient rescaling and the effective learning rates of weight vectors tend to decrease during training, leading to sharp local minima that fail to generalise well [182]. For this reason, we chose SGD as our optimisation function.

8.5.4. Loss Function

In the YOLOv5, a compound loss is calculated based on the objectness score in an image, class probability score and bounding box regression score. Our implementation used Pytorch's Binary Cross-Entropy with Logits Loss function for loss calculation of class probability and object score.

8.5.5. Model Training

For model training, we ran the command below,

$$!python\ train.py\ --img\ 416\ --batch\ 16\ --epoch\ 100\ --data\ 'path'\ --cfg\ yolov5s.yaml\ --weights\ ''\ --name\ yolov5s\_results\ --nosave\ --cache,$$

where *train.py* is our python class handling training, *img* is the input image size, *batch* is the batch size, *epoch* is the number of training epochs, *data* is our yaml file with our model configuration and architecture, *cfg* specifies our model configuration, *weights* is a custom path to weights for transfer learning, *name* is the result names, *nosave* allows us to only save the final checkpoint and *cache* caches images for faster training.

We chose *img* value 416 because that is a good representative of our image sizes in our dataset. Although we used 128 as an image size in the traditional pipelines, the difference is traditional pipelines were only using detected face images and here we are using an entire image. *batch* refers to the number of training examples processed before the model is updated. There are no universal rules for choosing the *batch* size, we chose 16 we wanted a small batch to be used by the algorithm. We used 100 *epochs* because we did not want the training process to be computationally expensive. We believe 100 *epochs* are enough to train our model effectively because of the use of small *batch* value. For our YOLOv5 algorithm, we did not use transfer learning. We chose the parameter values above based on the Ultralytics implementation because their implementation produced optimal results. After training, we used the model extracted from training for classification.

8.6 Pipeline 4 (Faster R-CNN)

In this section, we will discuss the Faster R-CNN implementation. We used Roboflow's Faster-RCNN implementation based on the tensorflow object recognition API, publicly accessible on roboflow-ai's github repository. We used Faster-RCNN because it performs faster and better than its predecessors. We used Google Colab environment for training because it accelerates training time.

Faster-RCNN is a two-stage object detector: it first identifies regions of interest then it passes the regions to a CNN. The resulting feature maps are passed to a SVMs algorithm for classification. Regression between predicted bounding boxes and ground truth bounding boxes are computed. Faster R-CNN uses two networks: region proposal network (RPN) for generating region

proposals and a network using these proposals to detect objects. RPN is used to rank region boxes, called anchors, and proposes the ones flagged as containing objects.

8.6.1. Model Training

For model training, we used Inception v2 CNN algorithm as our architecture because of its high accuracy. We also used a batch size of 12 for training because it has been proven to work well. We used L2 regularisation because of its ability to force the weights to be small but not zero.

To train our model, we used the below command,

$$!python\ model\_main.py\ --pipeline\_config\_path = pipeline\_name\ --model\_dir = model\_dir\ --alsologtoderr\ --num\_train\_steps = 20000\ --num\_eval\_steps = 50,$$

where *model_main.py* is the python class with our model, *pipeline_config_path* is the pipeline for our algorithm, *model_dir* contains our faster R-CNN model architecture, *alsologtoderr* logs any errors encountered while training our model, *num_train_steps* is the number of the algorithm's training steps and *num_eval_steps* is the number of the algorithm's evaluation steps after training. For transfer learning, the COCO dataset was used. COCO is a large-scale object recognition, segmentation, and captioning dataset. The hyperparameter values we used are the default values. We used these values because they have proven to be optimal based on both our implementation and the roboflow's implementation. After training, we saved the best performing model and used it for inferences.

8.7 Optimisation Techniques

In our research, we implemented the hyperparameter tuning and k-fold validation techniques to search for the best performing model and make our model more robust. We only implemented these optimisation techniques on traditional pipelines because deep learning pipelines already have optimisation techniques imbedded in them. These optimisation techniques are discussed briefly in the sections below.

8.7.1. K-Fold Cross-Validation Assessment Protocol

k-fold cross-validation is a re-sampling technique used for evaluating machine learning models on a small sample dataset [183]. It is a validation technique used to make a model robust. In k-fold validation, data is split into k equal folds. The model is trained on k-1 folds and one fold is left out for testing [184]. This process is repeated k times while changing the test part one-by-one until testing has been done on all k parts. Data is reshuffled and re-stratified before each round. The accuracy obtained in each iteration is averaged to get the overall model accuracy. Repeating the process k times makes the algorithm robust and likely to perform reliable estimations or comparisons [184].

The higher the value of k, the higher the accuracy in cross-validation [184]. However, increasing the value of k might lead to overfitting. There are other cross-validation techniques, but we chose k-fold because it is easy to understand and results in a more accurate and robust model that is less biased than other approaches, such as a train-test divide [183]. In the next section, we will discuss the algorithms we used in the prototype.

For k-fold cross-validation, we used *scikit-learn*'s *KFold* method below,

$$kfold = KFold(n\_splits = 5, shuffle = True),$$

where *n_splits* is the number of folds and *shuffle* is the option to shuffle the samples. We chose 5 splits because they are a great balance between effective cross-validation and computationally not too expensive. We used k-fold cross-validation during hyperparameter-tuning to make sure each randomly chosen set of parameters is validated and evaluated for accuracy and robustness effectively. Hyperparameter-tuning is discussed below.

8.7.2. Hyperparameter-Tuning Optimisation

A machine learning algorithm has sensitive parameters that affects its performance. Bad parameter values lead to an inefficient model, and optimal parameter values lead to optimal model performance. Hyperparameter tuning is an outer optimization technique on top of machine learning model training used to pick the algorithm's optimal parameters [185]. It is meant to be a

once-off activity to get the best performing model's hyperparameters. These parameters are then used to test the model's performance. Evaluating a hyperparameter configuration can be costly, ranging from hours to days depending on the dataset and the parameter space [185].

Machine Learning (ML) algorithms have complex and large hyperparameter spaces and are very sensitive to the choice of hyperparameters. Tuning a classifier's hyperparameters is important when selecting the best model but it increases the computation overhead [186]. Traditional brute force techniques such as random, grid and sequential searches are used to find the optimal hyperparameters. These techniques come with their advantages and disadvantages.

In our research we used the random search method. The random search algorithm picks hyperparameter value combinations at random and tests their performance. This process is repeated for a specified number of iterations. At the end of the iterations, the hyperparameters with the best performance are chosen as our model. We chose this method because it has proven to be very effective in literature and it has better time complexity than grid search.

We implemented hyperparameter-tuning using *scikit-learn*'s *randomSearchCV* method below,

$$search = RandomizedSearchCV(ml\_model, param\_distributions = params, random\_state = 42, n\_iter = 200, cv = kfold\_5, verbose = 1, n\_jobs = 1, refit = True, return\_train\_score = True),$$

where *ml_model* is our machine learning model, *param_distributions* is a list of our four SVM parameters listed above with their value ranges, *random_state* is a pseudo-random number generator state used for random number sampling, *n_iter* is the number of parameter settings that are sampled, *cv* is the cross-validation generator, *verbose* is the messages from the algorithm in its varying states, *n_jobs* is the number of jobs run in parallel, *refit* refits an estimator with the best found parameters on the whole dataset and *return_train_score* provides insights on how different parameter settings impact the overfitting/underfitting tradeoff.

We chose *n_iter* value of 200 because we believe 200 parameter settings with k-fold validation are enough to find good accuracy. We chose *refit* value true because we want our best performing model to be trained on the whole dataset before testing. We also chose

return_train_score value true to keep track of the overfitting/underfitting tradeoff impact on the randomly chosen parameter settings.

8.8 Conclusion

In this chapter we discussed the implementation details of our research prototype. The methods discussed are based on our research model, and our prototype aims to provide a solution to our research problem. We provided both the traditional biometrics pipelines and deep learning pipelines with the intention of widening our solution space to find the best performing model for inebriation recognition in humans using computer vision. For the traditional biometrics algorithms, we used optimisation techniques such as K-fold cross-validation and hyperparameter-tuning.

In this chapter we provided implementation details to state-of-the-art algorithms that have been used in various environments with varying success in literature. With these implementation details, we believe we provided enough information for our readers to recreate our experiments. We provided the environment details, the algorithms and their parameters and the optimisation techniques we used to find the best model.

By completing this chapter, we have met the first part of our research objective 5. The next section we will provide results for our research. Our results are gathered by using our benchmark to statistically analyse the performance of our prototype. Our research benchmark is discussed in the benchmark chapter.

Chapter 9 Results

9.1 Introduction

In this chapter, we are going to be presenting our results from the research prototype. We provided the implementation details of our various pipelines in the Prototype chapter.

In this chapter we will be showing results of the parameters discussed in the prototype chapter. We will use the functional and non-functional requirements discussed in the Benchmark chapter to statistically evaluate and analyse the performance of our various pipelines making up our model. We will provide the results of each pipeline and choose the best performing pipeline.

In Section 2 we will provide the results for the Local Binary Patterns (LBP) prototype, with support vector machines (SVMs), gradient boosted trees and random forests as various classifiers. In Section 3 we will show results for the Histogram of Gradients pipeline, with SVMs, gradient boosted trees and random forests as classifiers. In Section 4 and 5, we will provide the results for YOLOv5 and Faster R-CNN pipelines, respectively. We will close our chapter with a brief comparison of our pipelines' performance and choose the best performing pipeline.

9.2 Pipeline 1 (Local Binary Patterns)

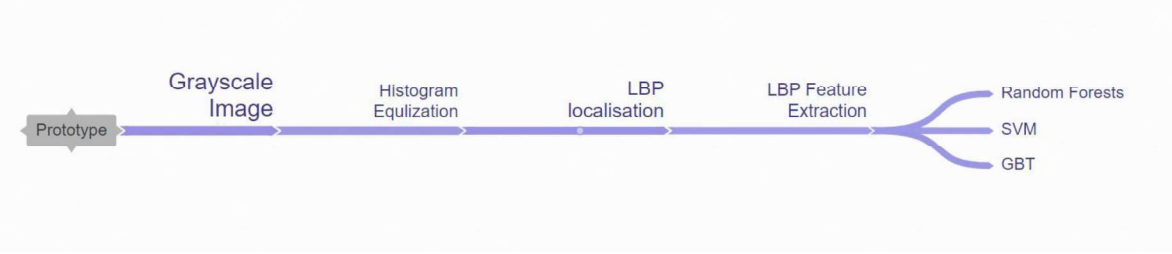

Figure 17: The Local Binary Patterns Pipeline.

In this section, we will discuss the results of the Local Binary Patterns pipeline. We will discuss results for both the functional and non-functional requirements.

9.2.1. Local Binary Patterns-Support Vector Machines

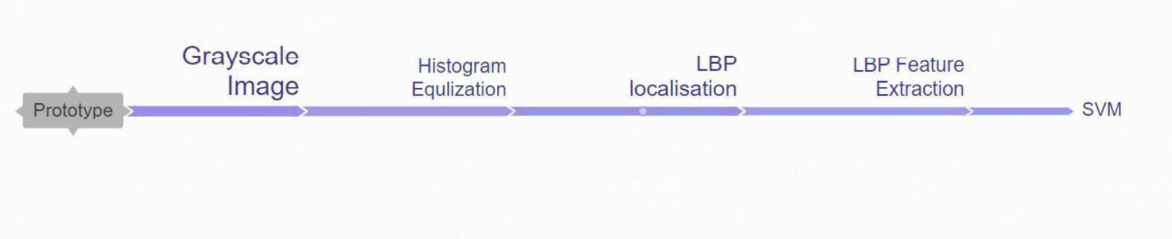

Figure 18: LBP with SVM Classifier Pipeline.

a. Functional Requirements

For functional requirements, we look at the research's ability to perform preprocessing, localisation, feature extraction and classification.

For preprocessing, we performed two operations: grayscaling an image and performing histogram equalisation on the resulting grayscaled image. The preprocessing step was successfully executed as shown by the image below.

Figure 19: Original image (left), grayscaled image (center) and histogram equalised image (right).

For localisation, we used local binary patterns to detect face images and crop them from the original image. Our algorithm performed well on the localisation step as shown in the image below.

Figure 20: Original image (left) and detected face image (right).

We used local binary pattern as a feature extractor. We stored a vector containing a histogram of our LBP output for classification. Our feature extractor was able to extract features that can be used to train a classifier to separate between sober and inebriated individuals. Below is a visualisation of our LBP feature.

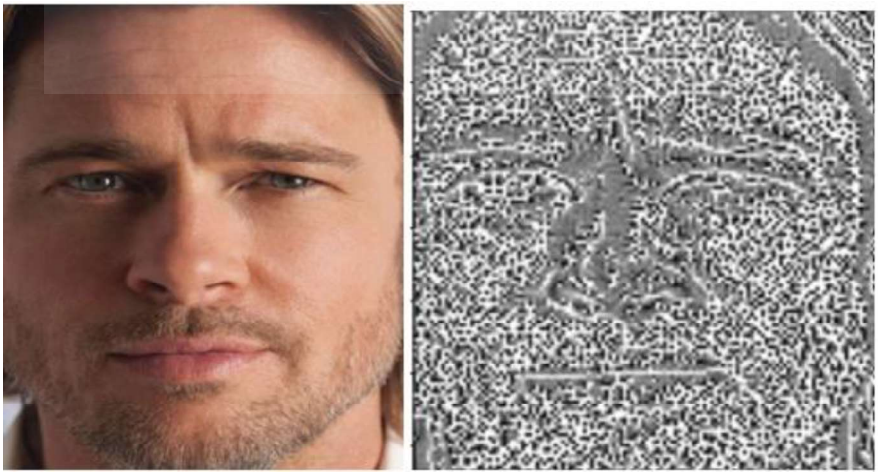

Figure 21: Detected face image (left) and local binary pattern of the face image (right).

b. Non-Functional Requirements

For non-functional requirements, we used the confusion matrix to calculate the accuracy, precision, recall, f1-score and area under the curve (AUC). These metrics are covered in the sections below.

After testing our model using the Support Vector Machines classifier with LBP features, we obtained the confusion matrix below:

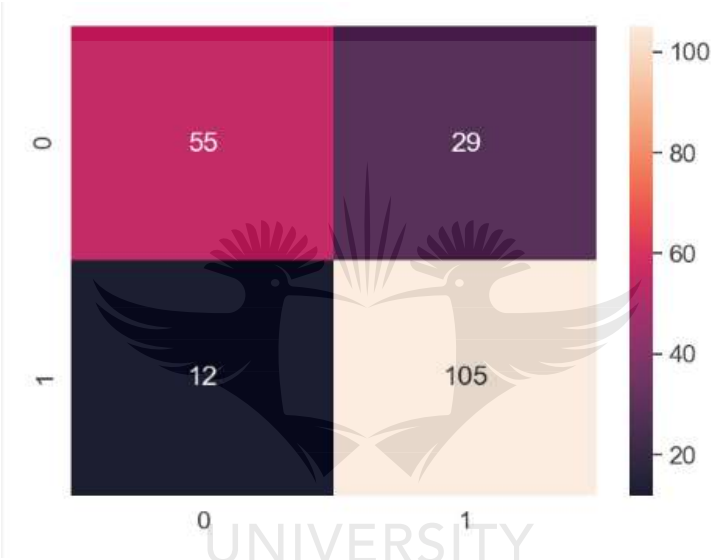

Figure 22: Confusion Matrix of the LBP-SVM pipeline.

We used 201 images for testing our model's performance. 55 individuals were correctly predicted as sober and 105 individuals were correctly predicted as inebriated. However, 29 individuals were wrongly predicted as inebriated and 12 individuals were wrongly predicted as sober. We believe this is a high misclassification rate. Some of the misclassified images are listed below.

Figure 23: Misclassification examples. Inebriated individuals classified as sober (top) and sober individuals classified as inebriated (bottom).

Of the misclassified cases, 66 percent (29 cases) involved augmented images. We believe our model does not perform well on rotated images. However, our model performed significantly well on horizontally flipped images. The misclassification cases on both inebriated and sober classes are equal.

The metrics used based on this confusion matrix are discussed below.

We calculated our accuracy, precision, recall and f1-score from the confusion matrix. These metrics are listed in the table below:

Table 1: LBP-SVM Metrics Results.

Metrics	Performance
Accuracy	79.6%
Precision	78.4%
Recall	89.7%
AUC	77.6%
F1-Score	83.7%

Our prototype yielded an accuracy of 79.6%, precision of 78.4%, recall of 89.7% and f1-score of 77.6%. Our precision was low because of a high number of sober cases misclassified as inebriated. We believe this is an acceptable performance on predicting inebriation in humans. Below is the receiver operating characteristic (ROC) curve:

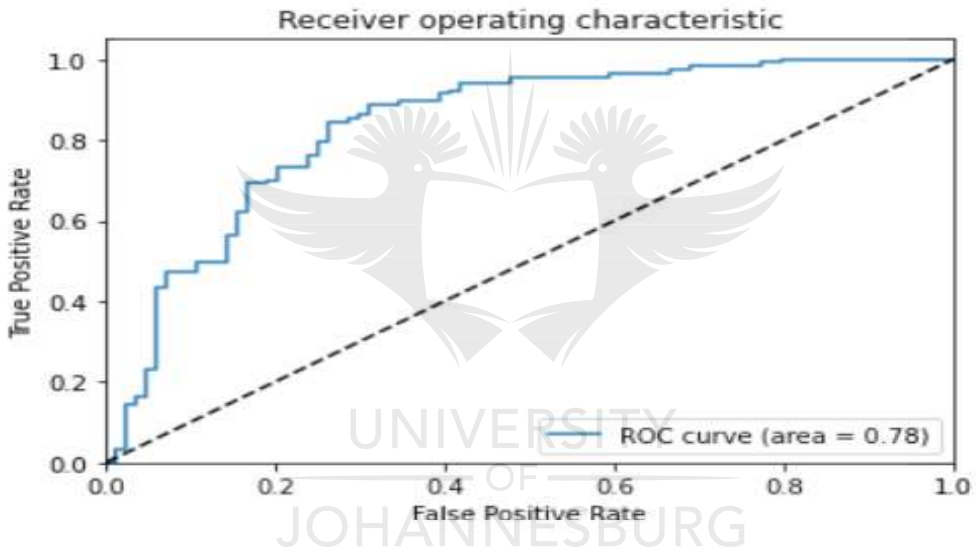

Figure 24: ROC Curve for LBP-SVM showing the AUC.

The ROC curve above shows the performance of our model in predicting inebriation in humans. The area under the curve (AUC) is 77.6%. Our ROC curve shows that there was imbalanced data, which is evident in our data sample.

9.2.2. Local Binary Patterns-Gradient Boosted Trees

Figure 25: LBP-GBT pipeline.

In this section, we will discuss the results of the Local Binary Patterns with Gradient Boosted Trees classifier pipeline.

a. Functional Requirements

For functional requirements, we looked at our pipeline's performance in preprocessing, localisation and feature extraction modules. The input and output of these modules is identical to the LBP-SVM pipeline discussed in Section 9.3.1. The results are identical as well, and we will not mention them here.

b. Non-Functional Requirements

For non-functional requirements, we used the confusion matrix to calculate the accuracy, precision, recall, f1-score and area under the curve (AUC). These metrics are covered in the sections below.

After obtaining results from our Gradient Boosted Trees classifier using features extracted from LBP, we obtained the confusion matrix below:

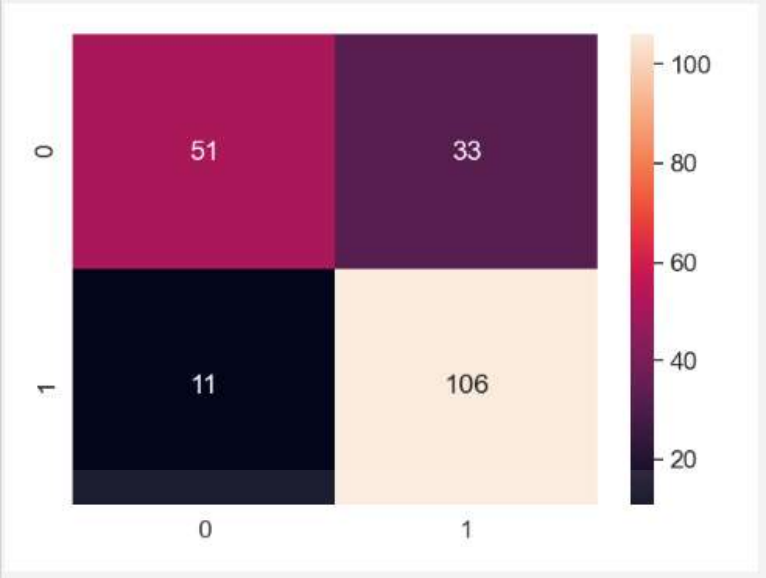

Figure 26: Confusion Matrix of the LBP-GBT pipeline.

We used 201 images to test our model. 51 individuals were correctly predicted as sober and 106 individuals were correctly predicted as inebriated. However, 33 individuals were wrongly predicted as inebriated and 11 individuals were wrongly predicted as sober. We believe this is a high misclassification rate.

Although our model performance was inferior to the LBP-SVM pipeline, the misclassified cases are almost identical. The model struggled to classify augmented images. We believe our model does not perform well on rotated images.

We calculated our accuracy, precision, recall and f1-score from the confusion matrix. These metrics are listed in the table below:

Table 2: LBP-GBT Metrics Results.

Metrics	Performance
Accuracy	78.1%
Precision	76.2%
Recall	90.1%
AUC	75.7%
F1-Score	82.3%

Our prototype yielded an accuracy of 78.1%, precision of 76.2%, recall of 90.1% and f1-score of 82.3%. Our precision was low because of a high number of sober cases misclassified as

inebriated. Although the performance is inferior to the LBP pipeline, we believe this is a relatively acceptable performance on predicting inebriation in humans. Below is the receiver operating characteristic (ROC) curve:

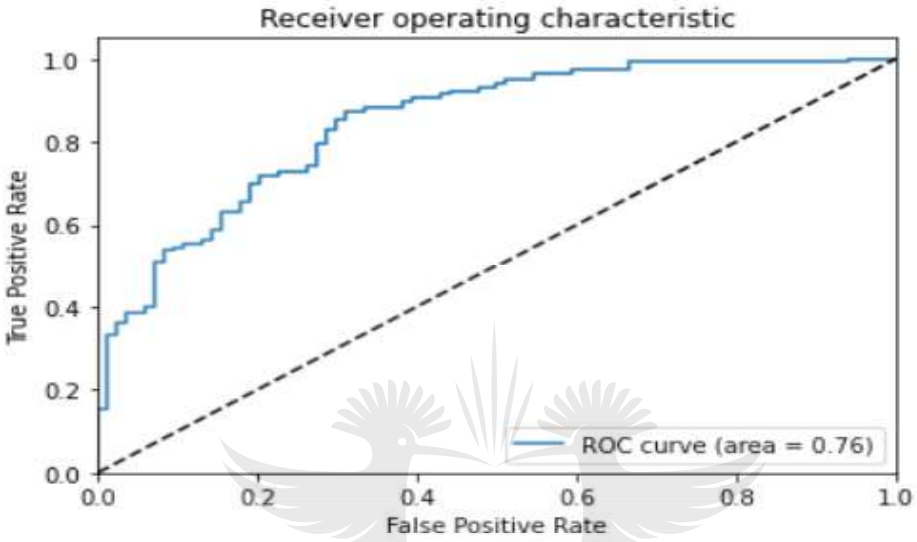

Figure 27: ROC Curve for LBP-GBT showing the AUC.

The ROC curve above shows the performance of our model in predicting inebriation in humans. The area under the curve (AUC) is 76%. The ROC curve also shows the imbalanced problem with our dataset.

9.2.3. Local Binary Patterns-Random Forests

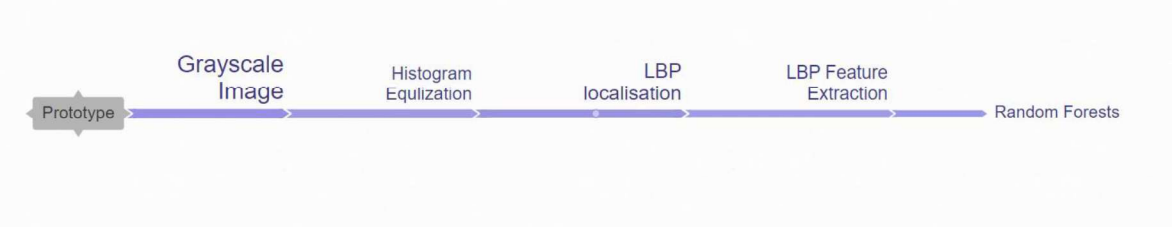

Figure 28: LBP-Random Forests pipeline.

In this section, we will discuss the results of the Local Binary Patterns with Random Forests classifier pipeline.

a. Functional Requirements

For functional requirements, we looked at our pipeline's performance in preprocessing, localisation and feature extraction modules. The input and output of these modules is identical to the LBP-SVM pipeline discussed in Section 9.3.1. The results are identical as well, and we will not mention them here.

b. Non-Functional Requirements

For non-functional requirements, we used the confusion matrix to calculate the accuracy, precision, recall, f1-score and area under the curve (AUC). These metrics are covered in the sections below.

After testing our Random Forests classifier using features extracted from LBP, we obtained the confusion matrix below:

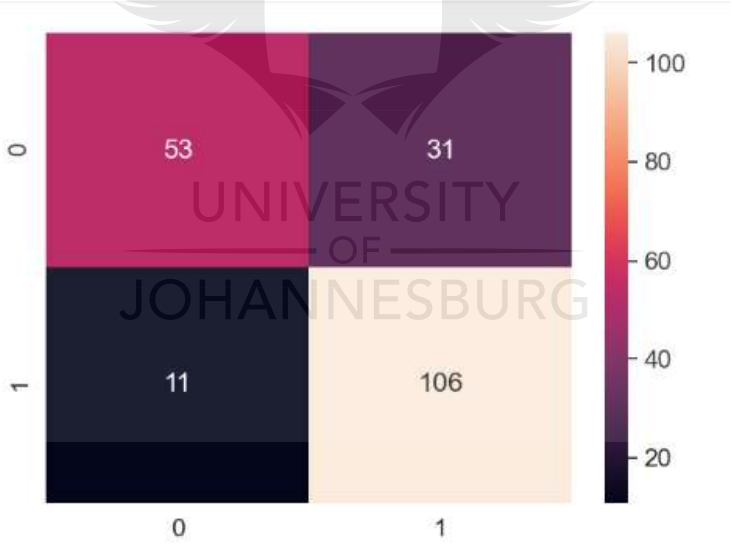

Figure 29: Confusion Matrix of the LBP-RT pipeline.

We used 201 images to test our model. 53 individuals were correctly predicted as sober and 106 individuals were correctly predicted as inebriated. However, 31 individuals were wrongly predicted as inebriated and 11 individuals were wrongly predicted as sober. We believe this is a high misclassification rate.

Our model did not perform well compared to the LBP-SVM pipeline and was slightly better than the LBP-GBT pipeline. Similar to both pipelines, our model had very similar misclassified cases. This might point to the LBP feature extractor as the reason for the misclassifications. The model struggled to classify augmented images. We believe our model does not perform well on rotated images.

The metrics used based on this confusion matrix are discussed below.

We calculated our accuracy, precision, recall and f1-score from the confusion matrix. These metrics are listed in the table below:

Table 3: LBP-RF Metrics Results.

Metrics	Performance
Accuracy	79.1%
Precision	77.4%
Recall	90.1%
AUC	76.8%
F1-Score	83.4%

Our prototype achieved an acceptable accuracy of 79.1%, precision of 77.4%, recall of 90.1% and f1-score of 83.4%. Our precision was low because of a high number of sober cases misclassified as inebriated. We believe this is a good performance on predicting inebriation in humans. Below is the receiver operating characteristic (ROC) curve:

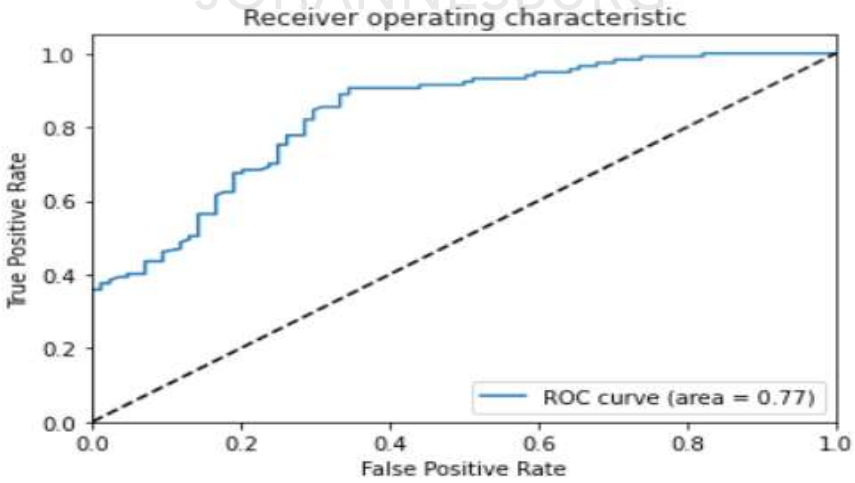

Figure 30: ROC Curve for LBP-RT showing the AUC.

The ROC curve above shows the performance of our model in predicting inebriation in humans. The area under the curve (AUC) is 76.8.

9.3 Pipeline 2 (Histogram of Gradients)

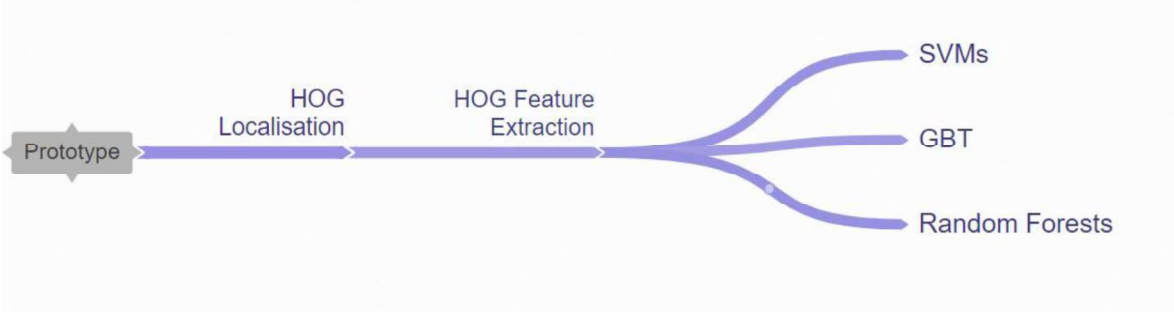

Figure 31: HOG pipeline.

In this section, we will discuss the results of the Histogram of Gradients pipeline. We will discuss results for both the functional and non-functional requirements.

9.3.1. Histogram of Gradients-Support Vector Machines

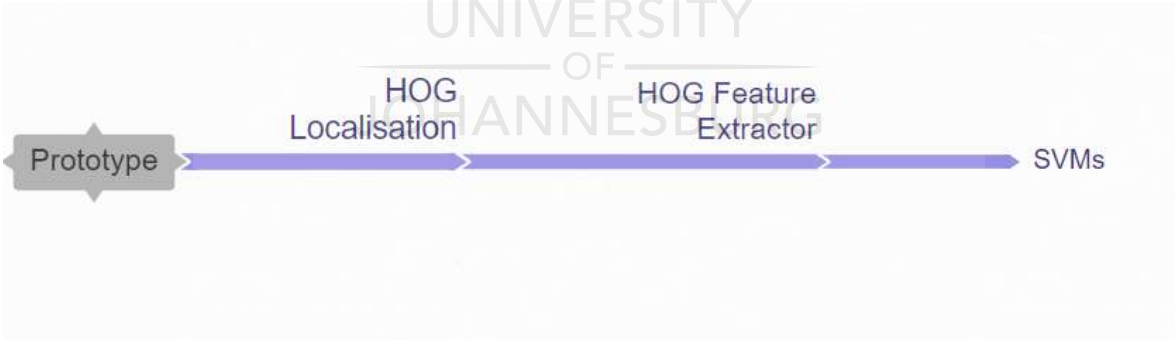

Figure 32: HOG-SVM pipeline

a. Functional Requirements

For functional requirements, we look at the research's ability to perform localisation, feature extraction and classification.

For localisation, we used histogram of gradients to detect face images and crop them from the original image. Our histogram of gradients algorithm that uses HoG to segment the face performed well on localising face images as shown in the image below.

Figure 33: Original image (left) and the extracted face image (right).

We used histogram of gradients for feature extraction as well. We stored a vector containing a histogram of gradients for each image as output for classification. Our feature extractor was able to extract features that can be used to train a classifier to separate between sober and inebriated individuals.

b. Non-Functional Requirements

For non-functional requirements, we used the confusion matrix to calculate the accuracy, precision, recall, f1-score and area under the curve (AUC). These metrics are covered in the sections below.

After obtaining results from our Support Vector Machines classifier using features extracted from HOG, we obtained the confusion matrix below:

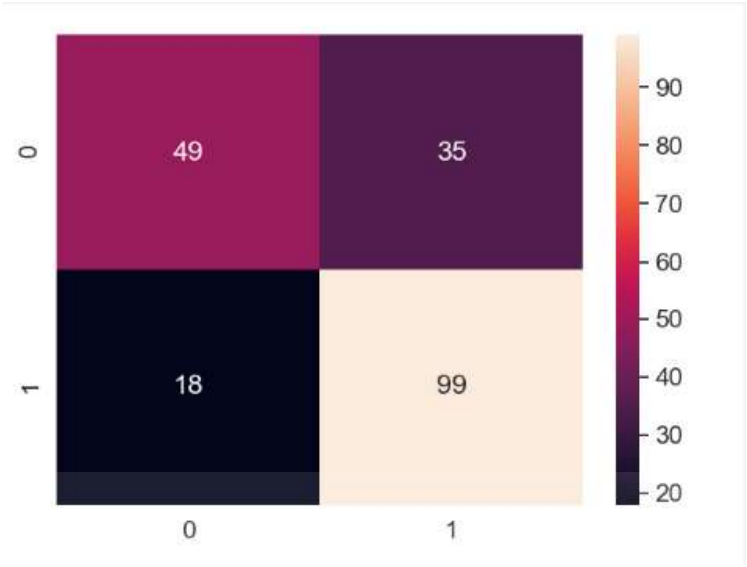

Figure 34: Confusion Matrix of the HOG-SVM pipeline.

We used 201 images for testing our model. 49 individuals were correctly predicted as sober and 99 individuals were correctly predicted as inebriated. However, 35 individuals were wrongly predicted as inebriated and 18 individuals were wrongly predicted as sober. We believe this is a quite high misclassification rate. Some of the misclassified images are listed below.

Figure 35: Misclassification examples. Inebriated individuals classified as sober (top) and sober individuals classified as inebriated (bottom).

Of the misclassified cases, 68 percent (34 cases) involved augmented images. Our model struggles to detect inebriation in humans. 56 percent (28 cases) of misclassification cases involve inebriated individuals. Our model struggled to classify inebriated individuals as inebriated and sober individuals as sober. Also, 68% of the misclassified cases by our model were also misclassified by the LBP-SVM pipeline. However, our model had more misclassifications than the LBP-SVM pipeline. The metrics used based on this confusion matrix are discussed below.

We calculated our accuracy, precision, recall and f1-score from the confusion matrix. These metrics are listed in the table below:

Table 4: HOG-SVM Metrics Results.

Metrics	Performance
Accuracy	76.3%
Precision	73.8%
Recall	84.6%
AUC	71.5%
F1-Score	78.9%

Our prototype yielded an accuracy of 76.3%, precision of 73.8%, recall of 84.6% and f1-score of 78.9%. Our precision was low because of a high number of sober cases misclassified as inebriated. We believe this is not a good performance on predicting inebriation in humans. Below is the receiver operating characteristic (ROC) curve:

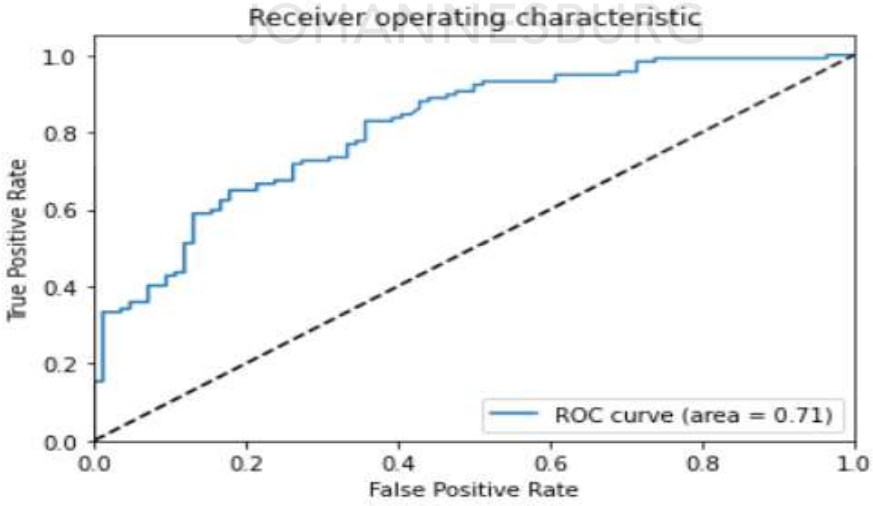

Figure 36: ROC Curve for LBP-GBT showing the AUC.

The ROC curve above shows the performance of our model in predicting inebriation in humans. The area under the curve (AUC) is 71.5%. The ROC curve also shows the imbalance problem with our dataset.

9.3.2. Histogram of Gradients-Gradient Boosted Trees

Figure 37: HOG with Gradient Boosted Trees Classifier.

In this section, we will discuss the results of the Histogram of Gradients with Gradient Boosted Trees classifier pipeline.

a. Functional Requirements

For functional requirements, we looked at our pipeline's performance in preprocessing, localisation and feature extraction modules. The input and output of these modules is identical to the HOG-SVM pipeline discussed in Section 4.1. The results are identical as well, and we will not mention them here.

b. Non-Functional Requirements

For non-functional requirements, we used the confusion matrix to calculate the accuracy, precision, recall, f1-score and area under the curve (AUC). These metrics are covered in the sections below.

After obtaining results from our Gradient Boosted Trees classifier using features extracted from HOG, we obtained the confusion matrix below:

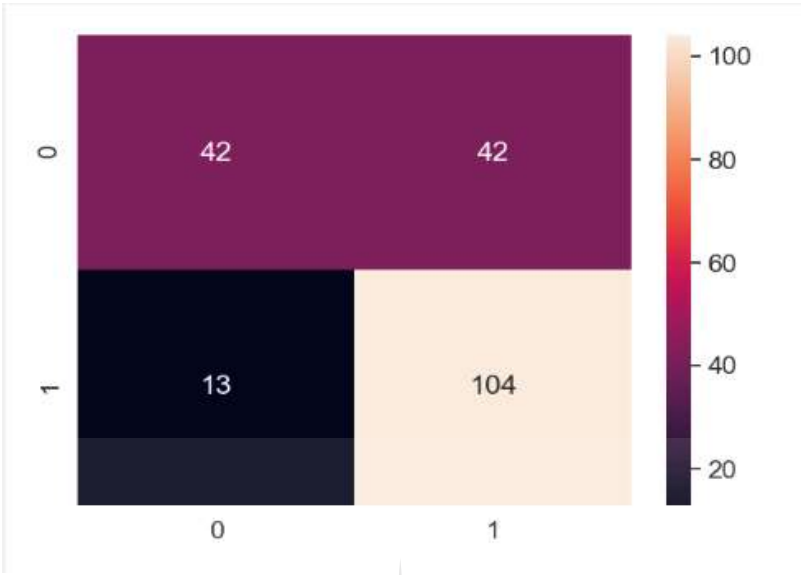

Figure 38: Confusion Matrix of the HOG-GBT pipeline.

We used 201 images for testing our model. 42 individuals were correctly predicted as sober and 104 individuals were correctly predicted as inebriated. However, 42 individuals were wrongly predicted as inebriated and 13 individuals were wrongly predicted as sober. We believe this is a quite high misclassification rate.

Of the misclassified cases, 60 percent (33 cases) involved augmented images. 53 percent (29 cases) of misclassification cases involve inebriated individuals. 98% of the misclassified cases by our model were also misclassified by the HOG-SVM pipeline. This might indicate that the feature extraction algorithm struggled to extract features with high class separability. The high misclassification rate suggests that our model struggled to detect inebriation.

We calculated our accuracy, precision, recall and f1-score from the confusion matrix. These metrics are listed in the table below:

Table 5: HOG-GBT Metrics Results.

Metrics	Performance
Accuracy	72.6%
Precision	71.2%
Recall	88.9%
AUC	69.4%
F1-Score	79.1%

Our prototype had an accuracy of 72.6%, precision of 71.2%, recall of 88.9% and f1-score of 79.1%. Our precision was low because of a high number of sober cases misclassified as inebriated. We believe this is not a performance on predicting inebriation in humans. Below is the receiver operating characteristic (ROC) curve:

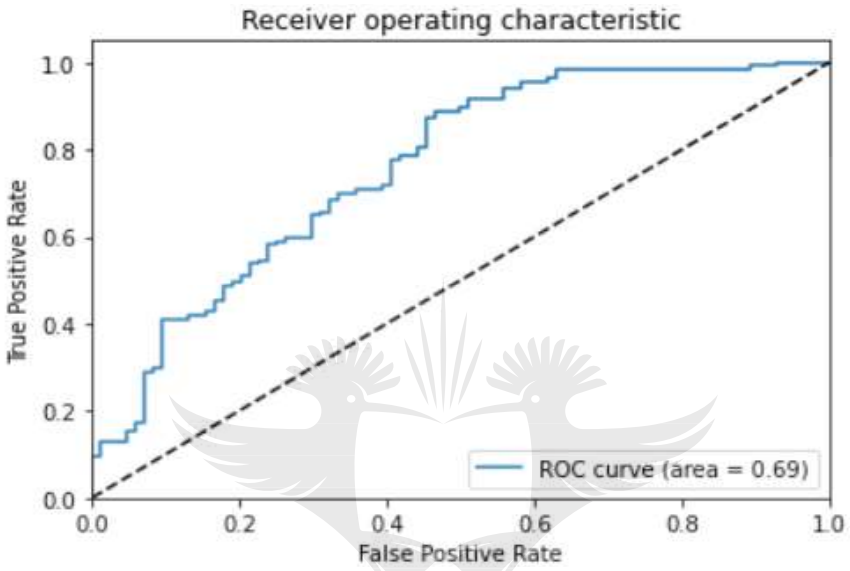

Figure 39: ROC Curve for LBP-GBT showing the AUC.

The ROC curve above shows the performance of our model in predicting inebriation in humans. The area under the curve (AUC) is 69%, which we believe is not a good score. The ROC curve also shows the imbalance problem with our dataset.

9.3.3. Histogram of Gradients-Random Forests

Figure 40: HOG with Random Forests Classifier Pipeline.

Chapter 9: Results

In this section, we will discuss the results of the Histogram of Gradients with Random Forests classifier pipeline.

a. Functional Requirements

For functional requirements, we looked at our pipeline's performance in preprocessing, localisation and feature extraction modules. The input and output of these modules is identical to the HOG-SVM pipeline discussed in Section 4.1. The results are identical as well, and we will not mention them here.

b. Non-Functional Requirements

For non-functional requirements, we used the confusion matrix to calculate the accuracy, precision, recall, f1-score and area under the curve (AUC). These metrics are covered in the sections below.

After obtaining results from our Random Forests classifier using features extracted from HOG, we obtained the confusion matrix below:

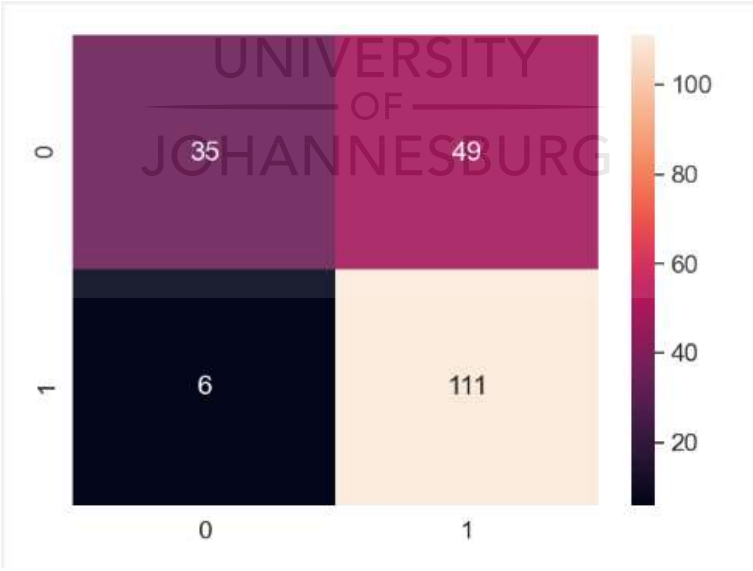

Figure 41: Confusion Matrix of the HOG-Random Trees pipeline.

Chapter 9: Results

We used 201 images for testing our model. 35 individuals were correctly predicted as sober and 111 individuals were correctly predicted as inebriated. However, 49 individuals were wrongly predicted as inebriated and 6 individuals were wrongly predicted as sober. We believe this is a quite high misclassification rate.

Of the misclassified cases, 65 percent (36 cases) involved augmented images. 53 percent (29 cases) of misclassification cases involve inebriated individuals. All the misclassified cases by our model were also misclassified by the HOG-GBT pipeline. This level of similarity on all our HOG pipelines indicate that the feature extraction algorithm struggled to extract features with high class separability. Our model predicted 80% of our dataset as inebriated, which is incorrect by 22%. This affects our model's precision. The high misclassification rate suggests that our model struggled to detect inebriation.

We calculated our accuracy, precision, recall and f1-score from the confusion matrix. These metrics are listed in the table below:

Table 6: HOG-Random Forests Metrics Results.

Metrics	Performance
Accuracy	72.6%
Precision	69.4%
Recall	94.9%
AUC	68.3%
F1-Score	80.1%

Our prototype had an accuracy of 72.6%, precision of 69.4%, recall of 94.9% and f1-score of 80.1%. Our precision was low because of a high number of sober cases misclassified as inebriated. We believe this is not a good performance on predicting inebriation in humans. Below is the receiver operating characteristic (ROC) curve:

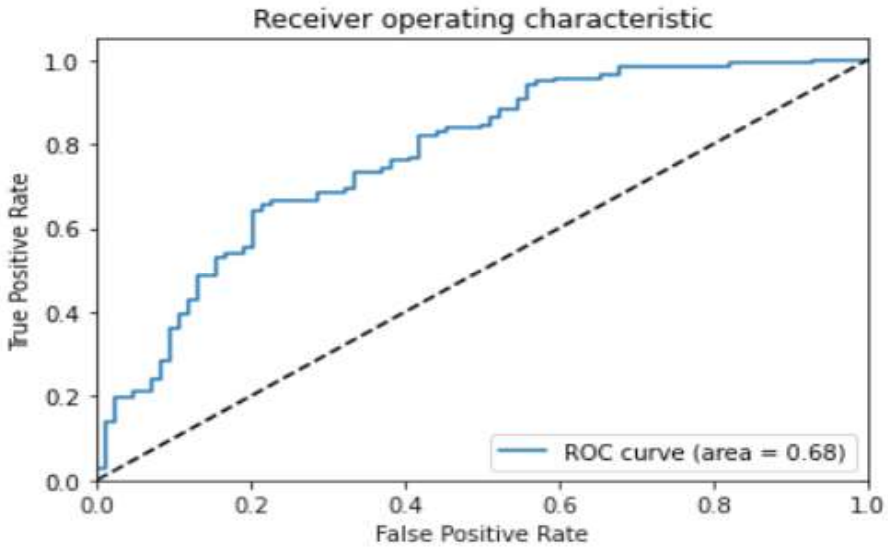

Figure 42: ROC Curve for LBP-Random Trees showing the AUC.

The ROC curve above shows the performance of our model in predicting inebriation in humans. The area under the curve (AUC) is 68.3%, which we believe is not a good score.

9.4 Pipeline 3 (YOLO)

In this section, we will discuss the results of the YOLOv5 pipeline. We will discuss results for both the functional and non-functional requirements.

9.4.1. Functional Requirements

For functional requirements, we looked at our pipeline's performance in both face detection and inebriation classification. Our YOLO pipeline performed well, showing state-of-the-art results. All face images were successfully detected and classified. Below are a few face images that were both detected and classified by our YOLO pipeline, including augmented images.

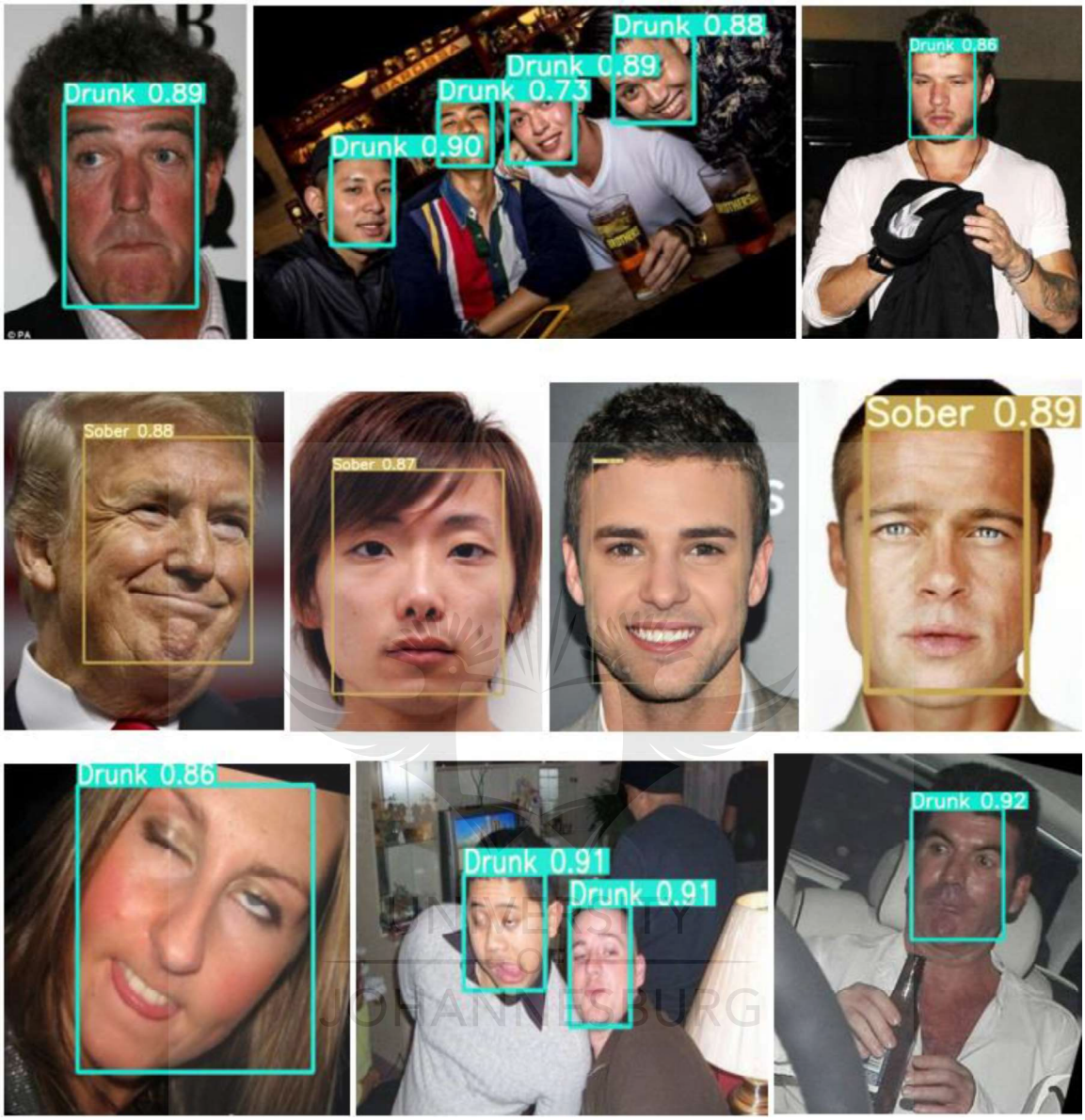

Figure 43: Face images detected and classified by YOLO.

9.4.2. Non-Functional Requirements

For non-functional requirements, we used the confusion matrix to calculate the accuracy, precision, recall, f1-score and area under the curve (AUC). These metrics are covered in the sections below.

After obtaining results from our pipeline, we obtained the confusion matrix below:

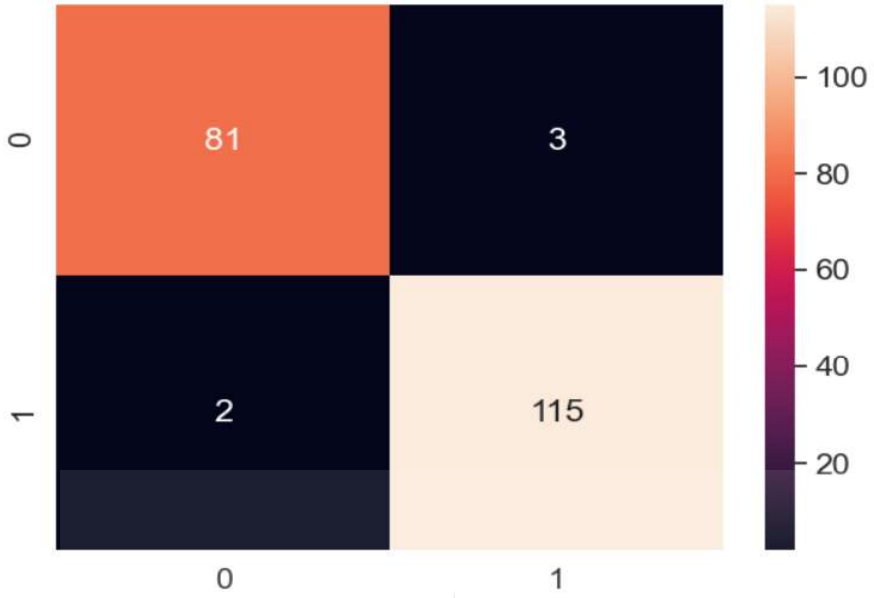

Figure 44: The Confusion Matrix for the YOLOv5 pipeline.

We used 201 face images for inferences. 81 individuals were correctly predicted as sober and 115 individuals were correctly predicted as inebriated. However, 3 individuals were wrongly predicted as inebriated and 2 individuals were wrongly predicted as sober. The 5 misclassifications are shown below.

Figure 45: The misclassification cases for YOLO.

All the 5 misclassification cases are augmented face images. Although our model performed well on most augmented images, we believe it struggles with them a bit. Also, 60 percent of the misclassified cases also have the correct classification detected as well, as shown on the above image.

Overall, we believe these are very good results and our pipeline performed very well in recognising inebriation. The metrics used based on this confusion matrix are discussed below. We calculated our accuracy, precision, recall and f1-score from the confusion matrix. These metrics are listed in the table below:

Table 7: YOLOv5 Metrics Results.

Metrics	Performance
Accuracy	97.5%
Precision	97.9%
Recall	97.5%
AUC	98.3%
F1-Score	97.4%

Our prototype had an accuracy of 97.5%, precision of 97.9%, recall of 97.5%, f1-score of 97.4% and area under the curve (AUC) of 98.3%. Our precision and recall are very high because of low misclassification rate on both classes. The low misclassification rates on sober and inebriated cases means our F1 score is very high. These are great results because in our research, we are looking for a model that has a high F1-score because it can be used in various environments and use cases that might require high precision, high recall or both. We believe this is a state-of-the-art performance on recognising inebriation in humans.

9.5 Pipeline 4 (Faster R-CNN)

In this section, we will discuss the results of the Faster R-CNN pipeline. We will discuss results for both the functional and non-functional requirements.

9.5.1. Functional Requirements

For functional requirements, we looked at our pipeline's performance in both face detection and inebriation classification. Our Faster R-CNN pipeline performed well, showing very good results. All face images used for testing our model were successfully detected. Below are a few

face images that were both detected and classified by our Faster R-CNN pipeline, including rotated images.

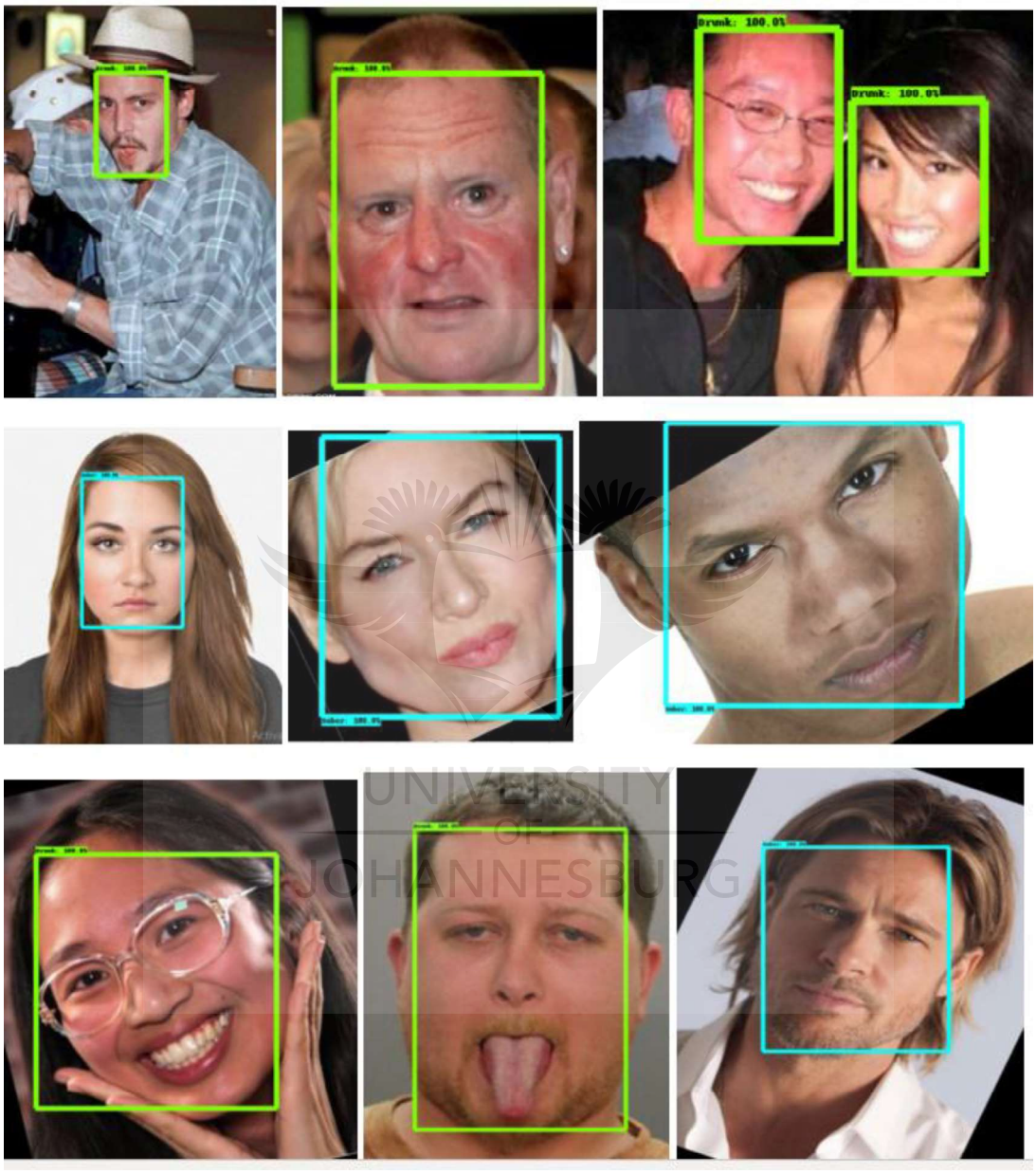

Figure 46: Rotated face images detected and classified by our algorithm.

9.5.2. Non-Functional Requirements

After obtaining results from our pipeline, we obtained the confusion matrix below:

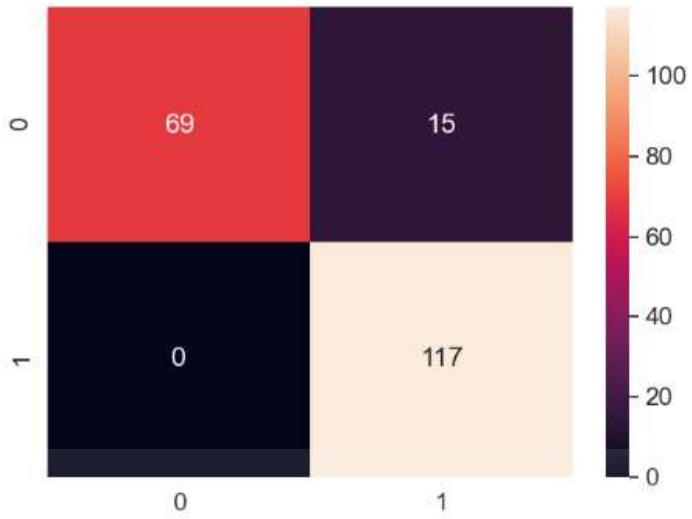

Figure 47: The Confusion Matrix for the Faster R-CNN pipeline.

We used 201 face images for inferences. 69 individuals were correctly predicted as sober and 117 individuals were correctly predicted as inebriated. However, 15 individuals were wrongly predicted as inebriated. No individuals were wrongly predicted as sober. Some of the misclassifications are shown below.

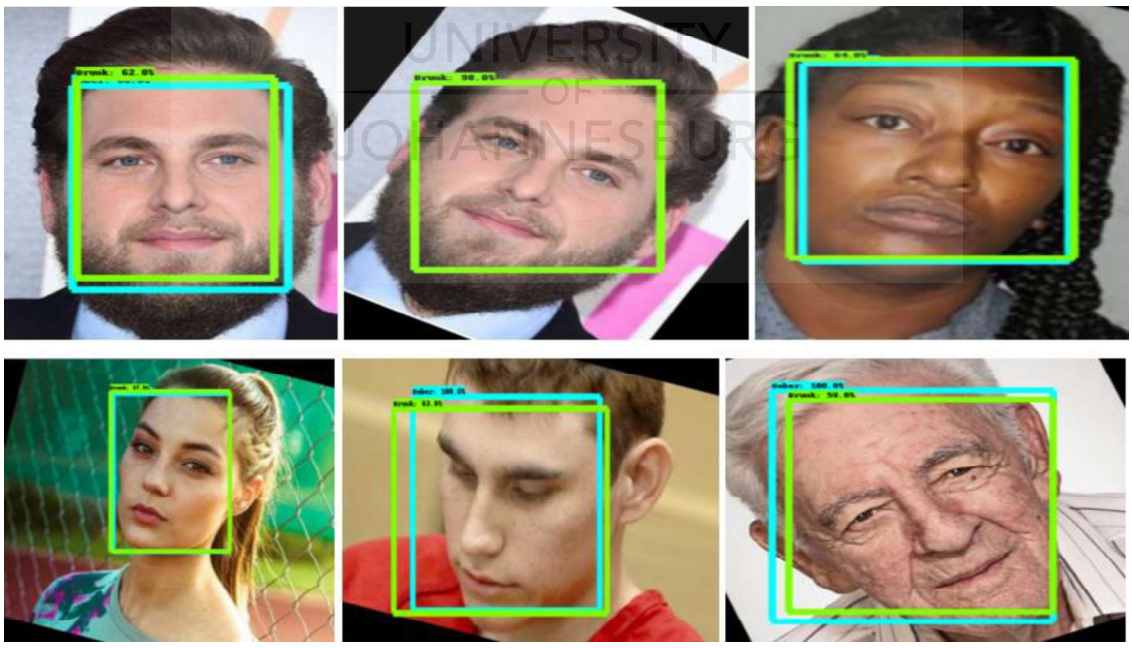

Figure 48: Misclassified images.

Of the 15 misclassified cases, 12 cases are augmented face images. Although our model performed well on most augmented images, we believe it significantly struggles with them. The original face images performed significantly better compared to their augmented counterparts as shown in the image below. Moreover, 14 of the 15 misclassified cases also have the correct classification detected as well, as shown on the above image. These misclassified images are the same ones that could not be classified correctly by our traditional methods.

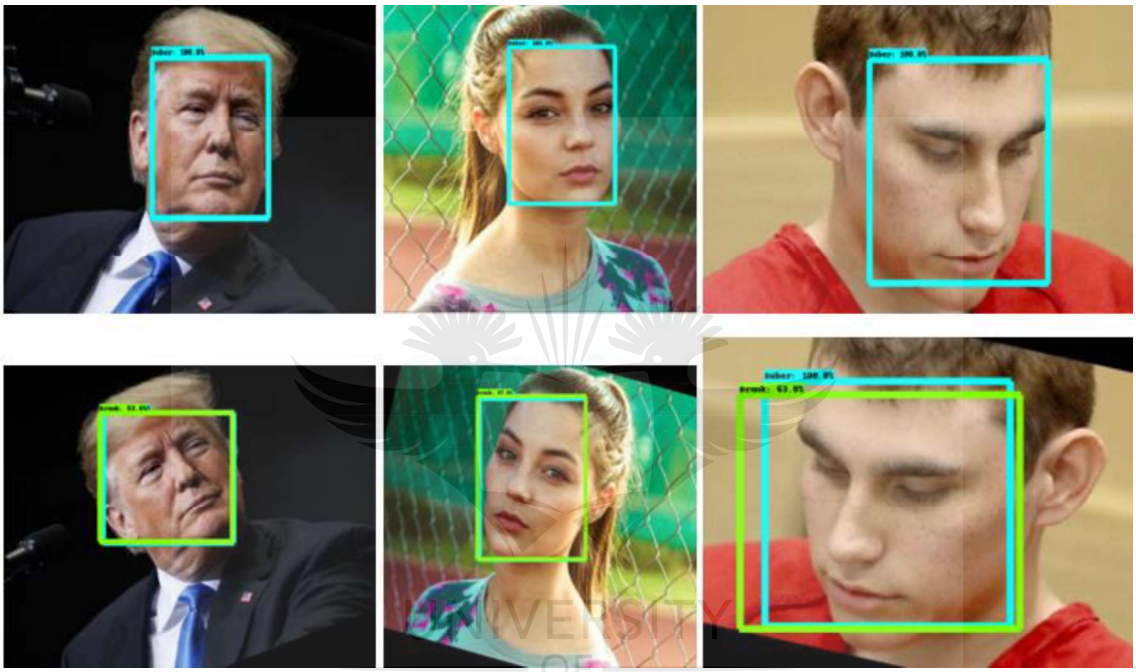

Figure 49: The original image (top) with correct classification vs the augmented images (bottom) misclassified.

Overall, we believe these are good results and our pipeline performed well in detecting inebriation. The metrics used based on the confusion matrix are discussed below.

We calculated our accuracy, precision, recall and f1-score from the confusion matrix. These metrics are listed in the table below:

Table 8: Faster R-CNN Metric Results.

Metrics	Performance
Accuracy	92.5%
Precision	88.6%
Recall	100%
AUC	91.1%
F1-Score	94%

Our prototype had an accuracy of 92.5%, precision of 88.6%, recall of 100%, f1-score of 94% and area under the curve (AUC) of 91.1%. The precision is low because of the sober cases misclassified as inebriated. Our recall is very high because there were no misclassifications on inebriated cases. We believe this is a good performance in predicting inebriation in humans.

9.6 Results Summary

Below is a metric summary of all our pipelines.

Table 9: Pipeline comparisons.

Pipeline	Accuracy	Precision	Recall	F1-Score	AUC
LBP-SVM	79.6%	78.4%	89.7%	83.7%	77.6%
LBP-GBT	78.1%	76.2%	90.1%	82.3%	75.7%
LBP-RF	79.1%	77.45	90.1%	83.4%	76.8%
HOG-SVM	76.3%	73.8%	84.6%	78.9%	71.5%
HOG-GBT	72.6%	71.2%	88.9%	69.4%	79.1%
HOG-RF	72.6%	69.4%	94.9%	80.1%	68.3%
YOLOv5	**97.5%**	**97.9%**	97.5%	**97.4%**	**98.3%**
Faster R-CNN	92.5%	88.6%	**100.0%**	94.0%	91.1%

As we mentioned in our benchmark chapter, the f1-score is our most important metric for our model. The f1-score is the balance between precision and recall, giving equal weight to both measurements. In our research, a high precision relates to low sober cases misclassified as

inebriated. A high recall relates to low inebriated cases misclassified as sober. A high f1-score relates to high precision and high recall. Our research aims to develop a balanced classification model with the optimal balance of recall and precision. A model with a higher f1-score means it exhibits a low FN and FP, which is the aim of this research. The greater the f1-score, the better the performance of our model.

Table 9 shows Faster R-CNN with the highest recall but a low recall in comparison, resulting in a low f1-score. YOLOv5 has the highest precision, a very good recall, and the highest f1-score because of the balance between its precision and recall. YOLOv5 had the least misclassification cases, all related to data augmentation. The traditional pipelines (LBP and HOG) had the highest misclassification cases. We believe traditional pipelines struggled because LBP and HOG do not extract enough features for inebriation classification that can be separated easily. We believe this is the reason why our classifiers struggled to recognise inebriation.

Table 10: Comparison of our YOLOv5 method with similar systems in literature. "-" denotes metrics that were not provided by the researchers.

Pipeline	Data Sampling	Accuracy	Precision	Recall	F1-Score
Wu et al. [38]	50	88%	-	88%	-
Azhar et al. [86]	120	90%	86%	94%	-
Mehta et al. [187]	166	88.39%	85%	99%	-
Lee et al. [188]	890	-	**98%**	93%	-
Neagoe and Diaconescu [189]	400	95.75%	-	-	-
Hnoohom and Yuenyong [87]	**3960**	88%	-	-	-
Bhango and van der Haar [190]	153	84.3%	84.4%	77.1%	77.1%
Faster R-CNN	920	92.5%	88.6%	**100%**	94%
YOLOv5	920	**97.5%**	97.9%	97.5%	**97.4%**

Deep learning pipelines performed well in comparison. We believe this is because of transfer learning, a deep network and the ability to integrate feature extraction within the training process. YOLOv5 was the best performing pipeline in our model. Below is a comparison of our methods against similar systems in literature. We chose methods that had at least 50 subjects for data sampling.

Table 10 shows that our method for inebriation recognition performed better than similar systems in literature. YOLOv5 had the best accuracy and F1-score. Although Lee et al [188] had a higher precision, their recall is lower than YOLOv5's, and to the best of our knowledge, they did not provide their accuracy and f1-score. Faster R-CNN had the best recall, but it was inferior to YOLOv5 in accuracy, precision, and the f1-score. YOLOv5 had the best overall results, with each metric having a score above 97%. YOLOv5 produced state-of-the-art results.

By completing our results chapter, we have met our research objective 5. We are now able to answer our research problem. Computer vision can be used to recognise inebriation in humans using computer vision. It provides for a faster, real-time, non-invasive, and convenient inebriation recognition solution.

9.7 Conclusion

In this chapter, we statistically evaluated and critically analysed the pipelines discussed in the Prototype chapter using the functional and non-functional requirements discussed in the Benchmark chapter. We analysed both the traditional and deep learning pipelines. We provided the classified images to show what worked and what did not.

Our pipelines performed relatively well, with our deep learning methods (YOLOv5 and Faster R-CNN) showing the best results. YOLov5 was our best performing pipeline, showing high precision and recall, resulting in high f1-score and AUC. YOLOv5 had the lowest misclassification cases. We noticed that augmented images played a huge role in misclassification cases, but the original images had a lower misclassification rate. Traditional pipelines struggled with detecting inebriation in face images.

In the next chapter, we will provide a fitting conclusion to our research.

Chapter 10 Conclusion

10.1 Introduction

In the previous chapter, we critically analysed and presented the results of our research prototype using the benchmark methods discussed in our benchmark chapter. We were also able to answer our research problem. A model for inebriation recognition using computer vision can be formulated with high performing results.

In this chapter, we will provide a conclusion to our research study. We will revisit our objectives, provide a summary of our research, discuss our findings, research impact, and provide avenues beyond our research we believe are worth venturing into.

In section 2 we will revisit our objectives. In section 3 and 4 we will discuss the findings and impact of our research, respectively. We will close off our research in section 4 with future work.

10.2 Objectives

Our research aim was to create a model for inebriation recognition using computer vision and statistically analyse and compare the performance of different pipelines to find the best performing pipeline. We used the design-oriented research science methodology to provide a robust platform and process for our research. We provided the research objectives that we needed to meet to achieve our research aim.

Objective 1 was to perform a literature review to understand inebriation, computer vision and investigate how similar work in computer vision and inebriation recognition was implemented in literature. We achieved this objective by performing literature review on our environment (Inebriation Recognition chapter), potential solution (Computer Vision chapter) and similar systems (Inebriation Recognition Using Computer Vision chapter|). By achieving this objective, we were able to learn our research environment and find gaps in literature for our research.

Objective 2 was to create our research model. We used the knowledge from our literature review to choose methods for our model. We achieved this objective by creating our research model in the Model chapter.

Objective 3 was to create a benchmark to quantify our research against. We developed both functional and non-functional requirements for our study that meets our research aim. These requirements are also used in literature. We achieved this objective by creating our benchmark in the Benchmark chapter.

Objective 4 was to create a dataset using publicly available face images of sober and inebriated individuals on the Internet. We achieved this objective by developing our dataset consisting of images that are free to use and have enough features to objectively sample our model and quantify our results.

Objective 5 was to develop our research prototype, based on our model, made up of various pipelines and critically analysed them statistically using our benchmark. We then compared the performance of our pipelines and picked the best performing pipeline. We achieved this objective by developing a prototype, critically analysing our results and comparing various pipelines to choose our best performing pipeline.

By completing our research objectives, we were able to answer our research questions. It is possible to recognise inebriation in humans using computer vision. We believe using computer vision to recognise inebriation is a great approach because it is fast, robust, non-invasive, and convenient.

10.3 Summary

Excessive alcohol consumption leads to inebriation. Alcohol abuse has become a social issue in need of solutions. A person in the world dies daily due to drunk driving [2] According to a study by the WHO, vehicle accidents will become the 5th highest cause of death if inebriated driving is not mitigated [30]. Inversely, 70% of the world population can be protected by handling drunk driving effectively. Excessive alcohol consumption leads to approximately 5.9% of all global death (3.3 million) each year [34]. Excessive alcohol use is the third leading lifestyle-related

cause of death in the United States and about 1.24 million people die on the road annually [35]. If this trend does not change, the death toll on the road is expected to reach 2.4 million by 2030, mostly caused by driving while inebriated [36].

Drinking too much alcohol has a heavy impact on our health. Heavy alcohol consumption meddles with the delicate balance of neurotransmitters. Neurotransmitters are chemicals in the brain that communicate with the nerve cells and are responsible for brain function [1]. Research suggests after long-term alcohol exposure, the brain attempts to restore equilibrium by compensating for the depressant effects of alcohol, thus, the brain decreases inhibitory and enhances excitatory neurotransmission [191].

Short-term effects of alcohol include lower inhibitions, lower caution, loss of fine motor coordination and inability to operate a motor vehicle [3]. Excessive alcohol consumption results in slurred speech, weakened balance, slow reaction times, staggering walk, or inability to walk. Glossy appearance to the eyes, blurry and double vision, loss of memory, heavy sweating, slower pupil response, slowed heart rate and breathing and reduced blood pressure can also result from drinking alcohol.

These short-term effects can be used to recognise inebriation in humans. Many systems have been used for inebriation recognition. These include using biomarkers such as blood tests, urine tests or hair tests. Behavioural traits emanating from inebriation can also be used to recognise inebriation. Heart-based signals can be used because the heart rate slows down when inebriated [50]. Thermal imaging can be used to recognise inebriation because inebriated people have more active blood vessels [47]. Gait can be used to recognise inebriation because of the loss of balance that results from inebriation [35]. Although these methods have been used to recognise inebriation with varying performance, they have their shortcomings, such as being too invasive (heart-based), requiring expensive equipment (thermal imaging), being too inconvenient (biomarkers) and not yielding enough inter-class and intra-class variability (gait).

The advent of faster, powerful, and easily accessible machines has facilitated the rapid growth of computer vision as a discipline. Convolutional Neural Networks (CNNs) have made

breakthroughs in character recognition, face recognition, pedestrian recognition, robot navigation, image restoration, object recognition and semantic segmentation.

Since its inception, computer vision systems have been deployed in retail, automotive, healthcare, agriculture, banking, and industry. Today, computer vision can be used for self-driving cars, diagnosing patients, monitoring crops, detecting anomalies such as terrorist threats and automating repetitive tasks with outstanding results.

We propose a model for inebriation detection using computer vision. Such a model will offer a real-time, non-intrusive, convenient, and efficient approach to inebriation recognition that can be used in various environments that require sobriety.

Our study produced a classifier that recognised inebriation in humans using computer vision with a 97.5% accuracy rate. We believe these are state-of-the-art results and the overall outcome of the research was a success.

10.4 Findings

From our research, we learnt that there is enough feature separability between sober and inebriated individuals to separate them, which support our hypothesis. Inebriation recognition in humans can be done using computer vision with state-of-the-art results. Traditional biometrics algorithms struggle to separate inebriation and sober individuals. They especially struggle with precision – recognising sober individuals as sober. In contrast, deep learning algorithms are high performing at recognising inebriation in humans. Although Faster R-CNN struggled slightly with precision, the recognition performance of both Faster R-CNN and YOLOv5 methods was very high. To the best of our knowledge, no publicly available datasets on inebriated and sober face images that can be used to classify inebriation. Both deep learning and traditional methods struggle with recognising augmented face images as opposed to normal images. Most of our misclassification cases were related to augmented images. However, adding augmented images for training vastly improved the deep learning pipelines' overall classifier performance.

10.5 Impact

Socially, alcohol abuse unattended leads to physical and mental harm [37]. Financially, accidents on the road cost USD 500 billion a year, which is between 1% to 3% of the world's GDP [38]. There are other issues related to excessive alcohol consumption such as lifestyle diseases, alcohol poisoning and inebriated driving with devastating and fatal consequences.

We believe our research has a role to play in society. We expect the model to be useful in societies by being able to recognise inebriation in individuals who are struggling with alcohol addiction and provide help. The model can also be used to reduce inebriation driving by capturing images and videos of drivers, thereby reducing fatalities and injuries on the road caused by inebriated driving.

Potential users for our research are public workers who work in environments where inebriation can potentially harm people, such as truck drivers and taxi drivers. Our research can also be used to monitor alcohol consumption at bars, with the bartender monitoring inebriation levels before selling more alcoholic beverages to individuals. Our research can also be used by traffic officers instead of breathalyzers to detect inebriation in drivers.

This enables the inebriation recognition system to be deployed in environments that require fast inebriation recognition such as roadblocks implemented by traffic officers and access control points where only sober individuals can enter a building or a vehicle. We believe there is a high potential for our research, and it can be used in literature to improve our method results. We will also be publishing our research dataset for public use.

10.6 Future Work

One of the issues we struggled with in our research was getting a scientifically proven secondary dataset available for public use. There are legal, ethical, and copyright issues related to that, but we believe having a scientifically proven dataset for benchmarking can facilitate research in inebriation recognition using computer vision. Our inebriation recognition model only recognises if an individual is inebriated or not. Different types of alcohol have varying effects on different

people. In-depth research on different alcohol types and their effects might help recognise the actual levels of inebriation in humans.

10.7 Conclusion

In this chapter, we revisited our research aim and objectives. We provided a summary of our research and discussed our research findings and the impact of our study. We also provided future work based on our research.

In this research, we successfully developed a model for inebriation recognition in humans using computer vision. We had a 97.5% inebriation recognition accuracy, thereby making it state-of-the-art results.

"First you take a drink. Then the drink takes a drink. Then the drink takes you" – F. Scott Fitzgerald.

References

[1] N. I. o. A. A. a. A. (U.S.), Beyond Hangovers: Understanding Alcohol's Impact Your Health, U.S. Department of Health and Human Services, National Institutes of Health, National Institute on Alcohol Abuse and Alcoholism, 2010.

[2] Y. Wu, Y. Xia, P. Xie and X. Ji, "The Design of an Automotive Anti-Drunk Driving System to Guarantee the Uniqueness of Driver," *2009 International Conference on Information Engineering and Computer Science,* pp. 1-4, 2009.

[3] A. Mariakakis, S. Parsi, S. N. Patel and J. O. Wobbrock, "Drunk User Interfaces: Determining Blood Alcohol Level Through Everyday Smartphone Tasks," *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems,* pp. 1-13, 2018.

[4] Y. Fan, Y. Zhang, Y. Ye, X. Li and W. Zheng, "Social Media for Opioid Addiction Epidemiology: Automatic Detection of Opioid Addicts from Twitter and Case Studies," in *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, Singapore, 2017.

[5] S. Gregor and A. Hevner, "Positioning and Presenting Design Science Research for Maximum Impact," *MIS Quarterly,* vol. 37, pp. 337-356, 2013.

[6] I. Akhtar, "Research Design," *Research in Social Science: Interdisciplinary Perspectives,* vol. I, pp. 68-84, 2016.

[7] K. J. Sileyew, "Research Design and Methodology," 7 August 2019. [Online]. Available: https://www.intechopen.com/online-first/research-design-and-methodology. [Accessed 29 January 2020].

[8] K. O. Darko-Ampem, "Scholarly Publishing In Africa: A Case Study Of The Policies And Practises Of African University Presses," 2003.

[9] P. K. Astalin, "Qualitative Research Designs: A Conceptual Framework," *International Journal of Social Science & Interdisciplinary Research,* vol. 2, pp. 118-124, 2013.

[10] M. Shuttleworth and L. T. Wilson, "Qualitative Research Design," Explorable, 14 September 2008. [Online]. Available: https://explorable.com/qualitative-research-design. [Accessed 4 March 2020].

[11] A. Hashizume and M. Kurosu, “Understanding User Experience and Artifact Development through Qualitative Investigation: Ethnographic Approach for Human-Centered Design,” *Human-Computer Interaction. Human-Centred Design Approaches, Methods, Tools, and Environments,* pp. 68-76, 2013.

[12] A. Teherani, T. Martimianakis, T. Stenfors-Hayes, A. Wadhwa and L. Varpio, “Choosing a Qualitative Research Approach,” *Journal of Graduate Medical Education,* vol. 7, no. 4, pp. 669-670, 2015.

[13] H. Elkatawneh, “The Five Qualitative Approaches: Problem, Purpose, and Questions/The Role of Theory in the Five Qualitative Approaches/Comparative Case Study,” *SSRN Electronic Journal,* pp. 1-17, 2016.

[14] M. Shuttleworth, “Quantitative Research Design,” Explorable, 7 March 2008. [Online]. Available: https://explorable.com/quantitative-research-design. [Accessed 5 March 2020].

[15] J. Schoonenboom and R. B. Johnson, “How to Construct a Mixed Methods Research Design,” *Kolner Zeitschrift fur Soziologie und Sozialpsychologie,* vol. 69, pp. 107-131, 2017.

[16] P. Žukauskas, J. Vveinhardt and R. Andriukaitienė, “Philosophy and Paradigm of Scientific Research, Management Culture and Corporate Social Responsibility,” 18 April 2018. [Online]. Available: https://www.intechopen.com/books/management-culture-and-corporate-social-responsibility/philosophy-and-paradigm-of-scientific-research. [Accessed 15 March 2020].

[17] P. Johannesson and E. Perjons, “Research Paradigms,” in *An Introduction to Design Science*, Springer International Publishing, 2014, pp. 167-179.

[18] A. A. Rehman and K. Alharthi, “An introduction to research paradigms,” *International Journal of Educational Investigations,* vol. 3, pp. 51-59, 2016.

[19] C. Kivunja and A. B. Kuyini, “Understanding and Applying Research Paradigms in Educational Contexts,” *International Journal of Higher Education,* vol. 6, pp. 26-41, 2017.

[20] L. Pham, “A Review of key paradigms: positivism, interpretivism and critical inquiry,” pp. 1-7, 2018.

[21] S. Wensveen and B. Matthews, “Prototypes and prototyping in design research,” *Routledge Companion to Design Research,* pp. 262-276, 2014.

[22] A. R. Hevner, S. T. March, J. Park and S. Ram, “Design Science in Information Systems

Research," *Management Information Systems Quarterly,* vol. 28, pp. 75-105, 2004.

[23] C. L. Owen, "Understanding Design Research: Towards an Achievement of Balance," *Japanese Society for the Science of Design,* pp. 36-45, 1997.

[24] H. Österle, J. Becker, U. Frank, T. Hess, D. Karagiannis , H. Krcmar, P. Loos, P. Mertens, A. Oberweis and E. J. Sinz , "Memorandum on design-oriented information systems research," *European Journal of Information Systems volume,* vol. 20, pp. 7-10, 2010.

[25] K. Peffers, T. Tuunanen, M. Rothenberger and S. Chatterjee, "A Design Science Research Methodology for Information Systems Research," *J. Manage. Inf. Syst.,* vol. 24, p. 45–77, 2007.

[26] J. Herrington, S. McKenney, T. C. Reeves and R. Oliver, "Design-based research and doctoral students: Guidelines for preparing a dissertation proposal," 2007.

[27] S. A. Barab and K. Squire, "Design-Based Research: Putting a Stake in the Ground," *The Journal of the Learning Sciences,* pp. 1-14, 2004.

[28] I. Cooper and J. Yon, "Ethical Issues in Biometrics," *Science Insights,* vol. 30, pp. 63-69, 2019.

[29] "4 Alcohol-Impaired Driving Interventions.," in *Getting to Zero Alcohol-Impaired Driving Fatalities: A Comprehensive Approach to a Persistent Problem*, Washington, DC, The National Academies Press, 2018, pp. 173-250.

[30] C. K. Wu, K. F. Tsang and H. R. Chi, "A wearable drunk detection scheme for healthcare applications," *2016 IEEE 14th International Conference on Industrial Informatics (INDIN),* pp. 878-881, 2016.

[31] E. Mirielli and L. Webster, "Modeling Alcohol Absorption and Elimination from the Human Body: A Case Study in Software Development: Nifty Assignment," *J. Comput. Sci. Coll.,* vol. 30, pp. 110-112, 2015.

[32] W.-F. Wang, C.-Y. Yang and Y.-F. Wu, "SVM-based Classification Method to Identify Alcohol Consumption Using ECG and PPG Monitoring," *Personal Ubiquitous Comput.,* vol. 22, pp. 275-287, 2018.

[33] J. Fell and R. Voas, "The effectiveness of reducing illegal blood alcohol concentration (BAC) limits for driving: Evidence for lowering the limit to .05 BAC," *Journal of safety research,* vol. 37, pp. 233-243, 2006.

[34] M. K. Toroghi, W. R. Cluett and R. Mahadevan, "Multiscale Metabolic Modeling Approach for Predicting Blood Alcohol Concentration," *IEEE Life Sciences Letters,* vol. 2, pp. 59-62, 2016.

[35] Z. Arnold, D. Larose and E. Agu, "Smartphone Inference of Alcohol Consumption Levels from Gait," *2015 International Conference on Healthcare Informatics,* pp. 417-426, 2015.

[36] V. Neagoe and S. Carata, "Drunkenness diagnosis using a Neural Network-based approach for analysis of facial images in the thermal infrared spectrum," *2017 E-Health and Bioengineering Conference (EHB),* pp. 165-168, 2017.

[37] E. Agu and C. Aiello, "Investigating postural sway features, normalization and personalization in detecting blood alcohol levels of smartphone users," *2016 IEEE Wireless Health (WH),* pp. 1-8, 2016.

[38] C. K. Wu, K. F. Tsang, H. R. Chi and F. H. Hung, "A Precise Drunk Driving Detection Using Weighted Kernel Based on Electrocardiogram," *Sensors,* vol. 16, 2016.

[39] G. Koukiou and V. Anasassopoulos, "Face locations suitable drunk persons identification," *2013 International Workshop on Biometrics and Forensics (IWBF),* pp. 1-4, 2013.

[40] M. Keall, W. J. Frith and T. L. Patterson, "The influence of alcohol, age and number of passengers on the night-time risk of driver fatal injury in New Zealand," *Accident; analysis and prevention,* pp. 49-61, 2004.

[41] M. Burns, "An overview of field sobriety test research," *Perceptual and motor skills,* vol. 97, pp. 1187-1199, 2004.

[42] K. Smith and W. Ulwelling, "The PEth Blood Test in the Security Environment: What it is; Why it is Important; and Interpretative Guidelines," *Journal of Forensic Sciences,* vol. 63, pp. 1634-1640, 2018.

[43] A. Varga, P. Hansson, C. Lundqvist and C. Alling, "Phosphatidylethanol in Blood as a Marker of Ethanol Consumption in Healthy Volunteers: Comparison with Other Markers," *Alcoholism: Clinical and Experimental Research,* vol. 22, pp. 1832-1837, 1998.

[44] H. Gnann, W. Weinmann and A. Thierauf, "Formation of Phosphatidylethanol and Its Subsequent Elimination During an Extensive Drinking Experiment Over 5 Days," *Alcoholism: Clinical and Experimental Research,* vol. 36, pp. 1507-1511, 2012.

[45] E. Weaver, D. Horyniak, R. Jenkinson, P. Dietze and M. Lim, ""Let's get Wasted!" and Other Apps: Characteristics, Acceptability, and Use of Alcohol-Related Smartphone

Applications," *JMIR mHealth and uHealth,* vol. 1, p. e9, 2013.

[46] H.-L. C. Kao, B.-J. Ho, A. C. Lin and H.-H. Chu, "Phone-based Gait Analysis to Detect Alcohol Usage," *Proceedings of the 2012 ACM Conference on Ubiquitous Computing,* pp. 661-662, 2012.

[47] M. K. Bhuyan, S. Bhuyan, P. Sasmal and G. Koukiou, "Intoxicated Person Identification Using Thermal Infrared Images and Gait," *2018 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET),* pp. 1-3, 2018.

[48] G. Koukiou, G. Panagopoulos and V. Anastassopoulos, "Drunk person identification using thermal infrared images," *2009 16th International Conference on Digital Signal Processing,* pp. 1-4, 2009.

[49] I. Chatterjee, Isha and A. Sharma, "Driving Fitness Detection : A Holistic Approach For Prevention of Drowsy and Drunk Driving using Computer Vision Techniques," *2018 South-Eastern European Design Automation, Computer Engineering, Computer Networks and Society Media Conference (SEEDA_CECNSM),* pp. 1-6, 2018.

[50] P. Koskinen, J. Virolainen and M. Kupari, "Acute Alcohol Intake Decreases Short-Term Heart Rate Variability in Healthy Subjects," *Clinical science (London, England : 1979),* vol. 87, pp. 225-230, 1994.

[51] E. Etemad and Q. Gao, "Object localization by optimizing convolutional neural network detection score using generic edge features," *2017 IEEE International Conference on Image Processing (ICIP),* pp. 675-679, 2017.

[52] T. H. Le, K. Luu, K. Seshadri and M. Savvides, "Beard and mustache segmentation using sparse classifiers on self-quotient images," *2012 19th IEEE International Conference on Image Processing,* pp. 165-168, 2012.

[53] X. Bai, B. Yin, Q. Shi and Y. Sun, "Face recognition using extended Fisherface with 3D morphable model," *2005 International Conference on Machine Learning and Cybernetics,* vol. 7, pp. 4481-4486, 2005.

[54] B. Liu, Z. Hao and X. Yang, "Nesting support vector machinte for muti-classification [machinte read machine]," *2005 International Conference on Machine Learning and Cybernetics,* vol. 7, pp. 4220-4225, 2005.

[55] I. I. Lychkov, A. N. Alfimtsev and S. A. Sakulin, "Tracking of Moving Objects With Regeneration of Object Feature Points," *2018 Global Smart Industry Conference (GloSIC),*

pp. 1-6, 2018.

[56] C. Wojek, S. Walk, S. Roth, K. Schindler and B. Schiele, "Monocular Visual Scene Understanding: Understanding Multi-Object Traffic Scenes," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 35, no. 4, pp. 882-897, 2013.

[57] J. E. Solem, Programming Computer Vision with Python, O'Reilly Media, Inc., 2012.

[58] S. J. D. Prince, Computer Vision: Models, Learning, and Inference, Cambridge: Cambridge University Press, 2012.

[59] A. Kaiser, "What is Computer Vision?," Hayo, 12 January 2017. [Online]. Available: https://hayo.io/computer-vision/. [Accessed 19 November 2020].

[60] V. Mishra, S. Kumar and N. Shukla, "Image Acquisition and Techniques to Perform Image Acquisition," *SAMRIDDHI: A Journal of Physical Sciences, Engineering and Technology,* 2017.

[61] S. Porob, G. Naik, H. Velingkar, D. Amonkar, R. Patil and P. Bhat, "Plant Health Monitoring using Digital Image Processing," *International Journal of Emerging Trends in Engineering and Development,* vol. 3, no. 7, pp. 147-151, 2017.

[62] P. Matula, M. Maska, O. Danek, P. Matula and M. Kozubek, "Acquiarium: Free software for the acquisition and analysis of 3D images of cells in fluorescence microscopy," *2009 IEEE International Symposium on Biomedical Imaging: From Nano to Macro,* pp. 1138-1141, 2009.

[63] U. Muhammad, W. Wang, S. P. Chattha and S. Ali, "Pre-trained VGGNet Architecture for Remote-Sensing Image Scene Classification," *2018 24th International Conference on Pattern Recognition (ICPR),* pp. 1622-1627, 2018.

[64] A. A. Almisreb, N. Jamil and M. N. Din, "Utilizing AlexNet Deep Transfer Learning for Ear Recognition," *2018 Fourth International Conference on Information Retrieval and Knowledge Management (CAMP),* pp. 1-5, 2018.

[65] S. Papert, "The Summer Vision Project," 1966.

[66] Y. LeCun, L. Bottou, Y. Bengio and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE,* vol. 86, no. 11, pp. 2278-2324, 1998.

[67] D. G. Lowe, "Object recognition from local scale-invariant features," *Proceedings of the Seventh IEEE International Conference on Computer Vision,* vol. 2, pp. 1150-1157, 1999.

[68] P. Viola and M. Jones, “Robust Real-time Object Detection,” *International Journal of Computer Vision - IJCV,* vol. 57, 2001.

[69] O. Russakovsky , J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla , M. Bernstein, A. C. Berg and L. Fei-Fei, “ImageNet Large Scale Visual Recognition Challenge,” *International Journal of Computer Vision (IJCV),* vol. 115, no. 3, pp. 211-252, 2015.

[70] A. Krizhevsky, I. Sutskever and G. E. Hinton, “ImageNet Classification with Deep Convolutional Neural Networks,” *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1,* pp. 1097-1105, 2012.

[71] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. E. Reed, D. Anguelov, D. Erhan, V. Vanhoucke and A. Rabinovich, “Going Deeper with Convolutions,” *Computing Research Repository,* 2014.

[72] S. Krig, “Image Pre-processing,” in *Computer Vision Metrics: Survey, Taxonomy, and Analysis*, Berkerly, CA, Apress, 2014, pp. 39-83.

[73] A. I. Taloba, A. A. Sewisy and Y. A. Dawood, “Accuracy Enhancement Scaling Factor of Viola- Jones Using Genetic Algorithms,” *2018 14th International Computer Engineering Conference (ICENCO),* pp. 209-212, 2018.

[74] S. Taneja, C. Gupta, S. Aggarwal and V. Jindal, “MFZ-KNN — A modified fuzzy based K nearest neighbor algorithm,” *2015 International Conference on Cognitive Computing and Information Processing(CCIP),* pp. 1-5, 2015.

[75] Y. Guo, L. Bai, S. Lao, S. Wu, M. S. Lew, W. T. Ooi, C. G. M. Snoek, H. K. Tan, C. Ho, B. Huet and C. Ngo, “A Comparison between Artificial Neural Network and Cascade-Correlation Neural Network in Concept Classification,” *Advances in Multimedia Information Processing -- PCM 2014,* pp. 248-253, 2014.

[76] StopLift, “StopLift,” StopLift, [Online]. Available: https://www.stoplift.com/. [Accessed 20 November 2019].

[77] Waymo, “Waymo,” Waymo, [Online]. Available: https://waymo.com/. [Accessed 20 November 2019].

[78] Tesla, “Tesla,” Tesla, [Online]. Available: https://www.tesla.com/. [Accessed 20 November 2019].

[79] Gauss Surgical, “Gauss Surgical,” Gauss Surgical, [Online]. Available:

http://www.gausssurgical.com/. [Accessed 20 November 2019].

[80] X. Zhang, W. Pan and P. Xiao, "In-Vivo Skin Capacitive Image Classification Using AlexNet Convolution Neural Network," *2018 IEEE 3rd International Conference on Image, Vision and Computing (ICIVC),* pp. 439-443, 2018.

[81] X. Han, "Nonnegative Principal Component Analysis for Cancer Molecular Pattern Discovery," *IEEE/ACM Transactions on Computational Biology and Bioinformatics,* vol. 7, pp. 537-549, 2010.

[82] R. Kurose, M. Hayashi, T. Ishii and Y. Aoki, "Player pose analysis in tennis video based on pose estimation," *2018 International Workshop on Advanced Image Technology (IWAIT),* pp. 1-4, 2018.

[83] Y. Qu, L. Jiang and X. Guo, "Moving vehicle detection with convolutional networks in UAV videos," *2016 2nd International Conference on Control, Automation and Robotics (ICCAR),* pp. 225-229, 2016.

[84] S. Sutor, F. Matusek and R. Reda, "WSSU: High Performance Wireless Self-Contained, Surveillance Unit; an Ad Hoc Video Surveillance System," *2008 Fourth Advanced International Conference on Telecommunications,* pp. 157-161, 2008.

[85] D. Barik and M. Mondal, "Object identification for computer vision using image segmentation," *2010 2nd International Conference on Education Technology and Computer,* vol. II, pp. 170-172, 2010.

[86] K. Azhar, F. Murtaza, M. H. Yousaf and H. A. Habib, "Computer vision based detection and localization of potholes in asphalt pavement images," *2016 IEEE Canadian Conference on Electrical and Computer Engineering (CCECE),* pp. 1-5, 2016.

[87] N. Hnoohom and S. Yuenyong, "Thai fast food image classification using deep learning," *2018 International ECTI Northern Section Conference on Electrical, Electronics, Computer and Telecommunications Engineering (ECTI-NCON),* pp. 116-119, 2018.

[88] A. Amodio, M. Ermidoro, D. Maggi, S. Formentin and S. M. Savaresi, "Automatic Detection of Driver Impairment Based on Pupillary Light Reflex," *IEEE Transactions on Intelligent Transportation Systems,* vol. 20, pp. 3038-3048, 2019.

[89] C. Willoughby, I. Banatoski, P. Roberts and E. Agu, "DrunkSelfie: Intoxication Detection from Smartphone Facial Images," *2019 IEEE 43rd Annual Computer Software and Applications Conference (COMPSAC),* pp. 496-501, 2019.

[90] M. Sooraj, J. Swathi, S. K. Anit, P. N. Anu and S. Sarath, "Driver Face Recognition and Sober Drunk Classification using Thermal Images," *2019 International Conference on Communication and Signal Processing (ICCSP),* pp. 400-404, 2019.

[91] F. Shafique and K. Mahmood, "Model Development as a research tool: An example of PAK-NISEA," *Library Philosophy and Practice,* pp. 1-12, 2010.

[92] R. H. Wiggins , C. H. Davidson, R. H. Harnsberger, J. R. Lauman and P. A. Goede, "Image File Formats: Past, Present, and Future," *RadioGraphics,* vol. 21, pp. 789-798, 2001.

[93] G. Roelofs, "History of the Portable Network Graphics (PNG) Format," Linux Journal, 1 April 1997. [Online]. Available: https://www.linuxjournal.com/article/2125. [Accessed 17 June 2019].

[94] M. Sahnoun, F. Kallel, M. Dammak, C. Mhiri, K. Ben Mahfoudh and A. Ben Hamida, "A comparative study of MRI contrast enhancement techniques based on Traditional Gamma Correction and Adaptive Gamma Correction: Case of multiple sclerosis pathology," *2018 4th International Conference on Advanced Technologies for Signal and Image Processing (ATSIP),* pp. 1-7, 2018.

[95] J. Han, S. Yang and B. Lee, "A Novel 3-D Color Histogram Equalization Method With Uniform 1-D Gray Scale Histogram," *IEEE Transactions on Image Processing,* vol. 20, pp. 506-512, 2011.

[96] S. Patel and M. Goswami, "Comparative analysis of Histogram Equalization techniques," *2014 International Conference on Contemporary Computing and Informatics (IC3I),* pp. 167-168, 2014.

[97] A. Anand, S. S. Tripathy and R. S. Kumar, "An improved edge detection using morphological Laplacian of Gaussian operator," *2015 2nd International Conference on Signal Processing and Integrated Networks (SPIN),* pp. 532-536, 2015.

[98] A. S. Mohamad, N. S. Abdul Halim, M. N. Nordin, R. Hamzah and J. Sathar, "Automated Detection of Human RBC in Diagnosing Sickle Cell Anemia with Laplacian of Gaussian Filter," *2018 IEEE Conference on Systems, Process and Control (ICSPC),* pp. 214-217, 2018.

[99] C. Bao and C. Sheng, "A parameterized logarithmic image processing method based on Laplacian of Gaussian filtering for lung nodules enhancement in chest radiographs," *2013 2nd International Symposium on Instrumentation and Measurement, Sensor Network and*

Automation (IMSNA), pp. 649-652, 2013.

[100] M. Nehru and S. Padmavathi, "Illumination invariant face detection using viola jones algorithm," *2017 4th International Conference on Advanced Computing and Communication Systems (ICACCS),* pp. 1-4, 2017.

[101] M. V. Alyushin and A. A. Lyubshov, "The Viola-Jones algorithm performance enhancement for a person's face recognition task in the long-wave infrared radiation range," *2018 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (EIConRus),* pp. 1813-1816, 2018.

[102] K. Vikram and S. Padmavathi, "Facial parts detection using Viola Jones algorithm," *2017 4th International Conference on Advanced Computing and Communication Systems (ICACCS),* pp. 1-4, 2017.

[103] K. Meena and A. Suruliandi, "Local binary patterns and its variants for face recognition," *2011 International Conference on Recent Trends in Information Technology (ICRTIT),* pp. 782-786, 2011.

[104] X. Wang, T. X. Han and S. Yan, "An HOG-LBP human detector with partial occlusion handling," *2009 IEEE 12th International Conference on Computer Vision,* pp. 32-39, 2009.

[105] Y. Ma, "Number Local binary pattern: An Extended Local Binary Pattern," *2011 International Conference on Wavelet Analysis and Pattern Recognition,* pp. 272-275, 2011.

[106] K. S. do Prado, "Face Recognition: Understanding LBPH Algorithm," Towards Data Science, 10 November 2017. [Online]. Available: https://towardsdatascience.com/face-recognition-how-lbph-works-90ec258c3d6b. [Accessed 22 November 2020].

[107] A. Y. Pratiwi, W. T. Budi and K. N. Ramadhani, "Identity recognition with palm vein feature using local binary pattern rotation Invariant," *2016 4th International Conference on Information and Communication Technology (ICoICT),* pp. 1-6, 2016.

[108] L. Cerkezi and C. Topal, "Gender recognition with uniform local binary patterns," *2018 26th Signal Processing and Communications Applications Conference (SIU),* pp. 1-4, 2018.

[109] M. Pietikäinen, "Local Binary Patterns," *Scholarpedia,* vol. 5, p. 9775, 2010.

[110] P. Torrione, K. D. Morton, R. Sakaguchi and L. M. Collins, "Histogram of gradient

features for buried threat detection in ground penetrating radar data," *2012 IEEE International Geoscience and Remote Sensing Symposium,* pp. 3182-3185, 2012.

[111] M. R. Guedira, A. E. Qadi, M. R. Lrit and M. E. Hassouni`, "A novel method for image categorization based on histogram oriented gradient and support vector machine," *2017 International Conference on Electrical and Information Technologies (ICEIT),* pp. 1-5, 2017.

[112] Q. Wu, H. Li, J. Niu and Y. Wang, "Gradient histogram Markov stationary features for image retrieval," *2012 5th International Congress on Image and Signal Processing,* pp. 790-794, 2012.

[113] S. Hsu, Y. Wang and C. Huang, "Human Object Identification for Human-Robot Interaction by Using Fast R-CNN," *2018 Second IEEE International Conference on Robotic Computing (IRC),* pp. 201-204, 2018.

[114] R. Girshick, J. Donahue, T. Darrell and J. Malik, "Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation," *2014 IEEE Conference on Computer Vision and Pattern Recognition,* pp. 580-587, 2014.

[115] R. Girshick, "Fast R-CNN," *2015 IEEE International Conference on Computer Vision (ICCV),* pp. 1440-1448, 2015.

[116] V. Kafedziski, S. Pecov and D. Tanevski, "Detection and Classification of Land Mines from Ground Penetrating Radar Data Using Faster R-CNN," *2018 26th Telecommunications Forum (TELFOR),* pp. 1-4, 2018.

[117] S. Ren, K. He, R. Girshick and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 39, 2015.

[118] B. Zhu, X. Wu, L. Yang, Y. Shen and L. Wu, "Automatic detection of books based on Faster R-CNN," *2016 Third International Conference on Digital Information Processing, Data Mining, and Wireless Communications (DIPDMWC),* pp. 8-12, 2012.

[119] W. Lan, J. Dang, Y. Wang and S. Wang, "Pedestrian Detection Based on YOLO Network Model," *2018 IEEE International Conference on Mechatronics and Automation (ICMA),* pp. 1547-1551, 2018.

[120] Z. Krawczyk and J. Starzyński, "Bones detection in the pelvic area on the basis of YOLO neural network," *19th International Conference Computational Problems of Electrical*

Engineering, pp. 1-4, 2018.

[121] P. Ren, W. Fang and S. Djahel, "A novel YOLO-Based real-time people counting approach," *2017 International Smart Cities Conference (ISC2),* pp. 1-2, 2017.

[122] X. Zhang and X. Ren, "Two Dimensional Principal Component Analysis based Independent Component Analysis for face recognition," *2011 International Conference on Multimedia Technology,* pp. 934-936, 2011.

[123] R. He, B. Hu, W. Zheng and X. Kong, "Robust Principal Component Analysis Based on Maximum Correntropy Criterion," *IEEE Transactions on Image Processing,* vol. 20, pp. 1485-1494, 2011.

[124] M. Johnson and A. Savakis, "Fast L1-eigenfaces for robust face recogntion," *2014 IEEE Western New York Image and Signal Processing Workshop (WNYISPW),* pp. 1-5, 2014.

[125] H. Zhang, W. Liu, L. Dong and Y. Wang, "Sparse eigenfaces analysis for recognition," *2014 12th International Conference on Signal Processing (ICSP),* pp. 887-890, 2014.

[126] P. Marasamy and S. Sumathi, "Automatic recognition and analysis of human faces and facial expression by LDA using wavelet transform," *2012 International Conference on Computer Communication and Informatics,* pp. 1-4, 2012.

[127] B. Huang, J. Li and S. Hu, "Texture feature extraction using ICA filters," *2008 7th World Congress on Intelligent Control and Automation,* pp. 7631-7634, 2008.

[128] A. S. Barhatte, R. Ghongade and S. V. Tekale, "Noise analysis of ECG signal using fast ICA," *2016 Conference on Advances in Signal Processing (CASP),* pp. 118-122, 2016.

[129] M. Phegade, P. Mukherji and U. S. Sutar, "Hybrid ICA algorithm for ECG analysis," *2012 12th International Conference on Hybrid Intelligent Systems (HIS),* pp. 478-483, 2012.

[130] M. M. Prasad, "1D-LDA verses 2D-LDA in online handwriting recognition," *International Conference on Circuits, Communication, Control and Computing,* pp. 431-433, 2014.

[131] S. Pang, T. Ban, Y. Kadobayashi and N. K. Kasabov, "LDA Merging and Splitting With Applications to Multiagent Cooperative Learning and System Alteration," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics),* vol. 42, pp. 552-564`, 2012.

[132] A. B. Rathod, S. M. Gulhane and S. R. Padalwar, "A comparative study on distance measuring approches for permutation representations," *2016 IEEE International*

Conference on Advances in Electronics, Communication and Computer Technology (ICAECCT), pp. 251-255, 2016.

[133] L. Greche, M. Jazouli, N. Es-Sbai, A. Majda and A. Zarghili, "Comparison between Euclidean and Manhattan distance measure for facial expressions classification," *2017 International Conference on Wireless Technologies, Embedded and Intelligent Systems (WITS),* pp. 1-4, 2017.

[134] R. Kumar, "Analysis of shape alignment using Euclidean and Manhattan distance metrics," *2017 International Conference on Recent Innovations in Signal processing and Embedded Systems (RISE),* pp. 326-331, 2017.

[135] J. Li and L. Peng, "Human expression recognition based on feature block 2DPCA and Manhattan distance classifier," *2008 7th World Congress on Intelligent Control and Automation,* pp. 5941-5945, 2008.

[136] J. Huang, Y. Wei, J. Yi and M. Liu, "An Improved kNN Based on Class Contribution and Feature Weighting," *2018 10th International Conference on Measuring Technology and Mechatronics Automation (ICMTMA),* pp. 313-316, 2018.

[137] A. Gong and Y. Liu, "Improved KNN Classification Algorithm by Dynamic Obtaining K," *Advanced Research on Electronic Commerce, Web Application, and Communication,* pp. 320-324, 2011.

[138] H. Yigit, "A weighting approach for KNN classifier," *2013 International Conference on Electronics, Computer and Computation (ICECCO),* pp. 228-231, 2013.

[139] Y. Ji, S. Yu and Y. Zhang, "A novel Naive Bayes model: Packaged Hidden Naive Bayes," *2011 6th IEEE Joint International Information Technology and Artificial Intelligence Conference,* vol. 2, pp. 484-487, 2011.

[140] A. H. Jahromi and M. Taheri, "A non-parametric mixture of Gaussian naive Bayes classifiers based on local independent features," *2017 Artificial Intelligence and Signal Processing Conference (AISP),* pp. 209-212, 2017.

[141] G. Singh, B. Kumar, L. Gaur and A. Tyagi, "Comparison between Multinomial and Bernoulli Naïve Bayes for Text Classification," *2019 International Conference on Automation, Computational and Technology Management (ICACTM),* pp. 593-596, 2019.

[142] B. Liu, Z. Hao and X. Yang, "Nesting support vector machinte for muti-classification [machinte read machine]," *2005 International Conference on Machine Learning and*

Cybernetics, vol. 7, pp. 4220-4225, 2005.

[143] X. Wang and S. Lu, "Improved Fuzzy Multicategory Support Vector Machines Classifier," *2006 International Conference on Machine Learning and Cybernetics,* pp. 3585-3589, 2006.

[144] Z. Liu and L. Bai, "Evaluating the supplier cooperative design ability using a novel support vector machine algorithm," *2008 12th International Conference on Computer Supported Cooperative Work in Design,* pp. 986-989, 2008.

[145] G. Ji, P. Han and Y. Zhai, "Wind Speed Forecasting Based on Support Vector Machine with Forecasting Error Estimation," *2007 International Conference on Machine Learning and Cybernetics,* vol. 5, pp. 2735-2739, 2007.

[146] J. Huang, Y. Wei, J. Yi and M. Liu, "An Improved kNN Based on Class Contribution and Feature Weighting," *2018 10th International Conference on Measuring Technology and Mechatronics Automation (ICMTMA),* pp. 313-316, 2018.

[147] L. Jinshu, W. Yijiang, W. Ganjun, P. Xiaoasheng, L. Taiwei and J. Yuhang, "Gradient Boosting Decision Tree and Random Forest Based Partial Discharge Pattern Recognition of HV Cable," *2018 China International Conference on Electricity Distribution (CICED),* pp. 327-331, 2018.

[148] P. Sheng, L. Chen and J. Tian, "Learning-based road crack detection using gradient boost decision tree," *2018 13th IEEE Conference on Industrial Electronics and Applications (ICIEA),* pp. 1228-1232, 2018.

[149] A. Gupta, K. Gusain and B. Popli, "Verifying the value and veracity of extreme gradient boosted decision trees on a variety of datasets," *2016 11th International Conference on Industrial and Information Systems (ICIIS),* pp. 457-462, 2016.

[150] K. Nugroho, E. Noersasongko, Purwanto, Muljono, A. Z. Fanani, Affandy and R. S. Basuki, "Improving Random Forest Method to Detect Hatespeech and Offensive Word," *2019 International Conference on Information and Communications Technology (ICOIACT),* pp. 514-518, 2019.

[151] Y. Guo, Y. Zhou, X. Hu and W. Cheng, "Research on Recommendation of Insurance Products Based on Random Forest," *2019 International Conference on Machine Learning, Big Data and Business Intelligence (MLBDBI),* pp. 308-311, 2019.

[152] R. U. Khan, X. Zhang, R. Kumar and H. A. Tariq, "Analysis of resnet model for malicious code detection," *2017 14th International Computer Conference on Wavelet Active Media*

Technology and Information Processing (ICCWAMTIP), pp. 239-242, 2017.

[153] Y. Xie, H. Jin and E. C. Tsang, "Improving the lenet with batch normalization and online hard example mining for digits recognition," *2017 International Conference on Wavelet Analysis and Pattern Recognition (ICWAPR),* pp. 149-153, 2017.

[154] M. Ma, Z. Gao, J. Wu, Y. Chen and X. Zheng, "A Smile Detection Method Based on Improved LeNet-5 and Support Vector Machine," *2018 IEEE SmartWorld, Ubiquitous Intelligence Computing, Advanced Trusted Computing, Scalable Computing Communications, Cloud Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI),* pp. 446-451, 2018.

[155] H. Jun, L. Shuai, S. Jinming, L. Yue, W. Jingwei and J. Peng, "Facial Expression Recognition Based on VGGNet Convolutional Neural Network," *2018,* 4146-4151.

[156] W. Lin, H. Lin, P. Wang, B. Wu and J. Tsai, "Using convolutional neural networks to network intrusion detection for cyber threats," *2018 IEEE International Conference on Applied System Invention (ICASI),* pp. 1107-1110, 2018.

[157] Z. Zhu, J. Li, L. Zhuo and J. Zhang, "Extreme Weather Recognition Using a Novel Fine-Tuning Strategy and Optimized GoogLeNet," *2017 International Conference on Digital Image Computing: Techniques and Applications (DICTA),* pp. 1-7, 2017.

[158] G. Feng, Z. Hu, S. Chen and F. Wu, "Energy-efficient and high-throughput FPGA-based accelerator for Convolutional Neural Networks," *2016 13th IEEE International Conference on Solid-State and Integrated Circuit Technology (ICSICT),* pp. 624-626, 2016.

[159] G. Wei, G. Li, S. Guan, J. Zhao and X. Sun, "Study on an Improved LeNet-5 Gas Identification Structure for Electronic Noses," *2018 IEEE SENSORS,* pp. 1-4, 2018.

[160] S. Sharma, A. Negi, A. Negi, G. Jasmine, V. V, S. Singh, D. S. Sathia Raj and S. Ganesan, "Eye state detection for use in advanced driver assistance systems," *2018 International Conference on Recent Trends in Advance Computing (ICRTAC),* pp. 155-161, 2018.

[161] Z. Ma, J. Ding, H. Wang and F. Wang, "Multi-type Digital Recognition Based on TensorFlow," *2018 Chinese Automation Congress (CAC),* pp. 1983-1985, 2018.

[162] L. Lv and Y. Tan, "Detection of cabinet in equipment floor based on AlexNet and SSD model," *The Journal of Engineering,* pp. 605-608, 2019.

[163] S. Fairuz, M. H. Habaebi and E. M. Elsheikh, "Finger Vein Identification Based On

Transfer Learning of AlexNet," *2018 7th International Conference on Computer and Communication Engineering (ICCCE),* pp. 465-469, 2018.

[164] D. C. Khrisne and I. M. Suyadnya, "Indonesian Herbs and Spices Recognition using Smaller VGGNet-like Network," *2018 International Conference on Smart Green Technology in Electrical and Information Systems (ICSGTEIS),* pp. 221-224, 2018.

[165] Y. Chen and X. Jiang, "No-reference Image Quality Assessment Based on Convolutional Neural Network," *2018 IEEE 18th International Conference on Communication Technology (ICCT),* pp. 1251-1255, 2018.

[166] P. Aswathy, Siddhartha and D. Mishra, "Deep GoogLeNet Features for Visual Object Tracking," *2018 IEEE 13th International Conference on Industrial and Information Systems (ICIIS),* pp. 60-66, 2018.

[167] B. Li and Y. He, "An Improved ResNet Based on the Adjustable Shortcut Connections," *IEEE Access,* pp. 18967-18974, 2018.

[168] K. He, X. Zhang, S. Ren and J. Sun, "Deep Residual Learning for Image Recognition," *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR),* pp. 770-778, 2016.

[169] T. Fang, "A Novel Computer-Aided Lung Cancer Detection Method Based on Transfer Learning from GoogLeNet and Median Intensity Projections," *2018 IEEE International Conference on Computer and Communication Engineering Technology (CCET),* pp. 286-290, 2018.

[170] Z. Zhong, L. Jin and Z. Xie, "High performance offline handwritten Chinese character recognition using GoogLeNet and directional feature maps," *2015 13th International Conference on Document Analysis and Recognition (ICDAR),* pp. 846-850, 2015.

[171] C. Chen and F. Qi, "Single Image Super-Resolution Using Deep CNN with Dense Skip Connections and Inception-ResNet," *2018 9th International Conference on Information Technology in Medicine and Education (ITME),* pp. 999-1003, 2018.

[172] M. F. Haque, H. Lim and D. Kang, "Object Detection Based on VGG with ResNet Network," *2019 International Conference on Electronics, Information, and Communication (ICEIC),* pp. 1-3, 2019.

[173] P. R. Anish, B. Balasubramaniam, J. Cleland-Huang, R. Wieringa, M. Daneva and S. Ghaisas, "Identifying Architecturally Significant Functional Requirements," *2015 IEEE/ACM 5th International Workshop on the Twin Peaks of Requirements and*

Architecture, pp. 3-8, 2015.

[174] V. Bajpai and R. P. Gorthi, "On non-functional requirements: A survey," *2012 IEEE Students' Conference on Electrical, Electronics and Computer Science,* pp. 1-4, 2012.

[175] X. Sun and W. Xu, "Fast Implementation of DeLong's Algorithm for Comparing the Areas Under Correlated Receiver Operating Characteristic Curves," *IEEE Signal Processing Letters,* vol. 21, pp. 1389-1393, 2014.

[176] D. Mossman , "Three-way ROCs," *Medical decision making : an international journal of the Society for Medical Decision Making,* vol. 19, no. 1, p. 78–89, 1999.

[177] W. A. Yousef, R. F. Wagner and M. H. Loew, "Assessing Classifiers from Two Independent Data Sets Using ROC Analysis: A Nonparametric Approach," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 28, pp. 1809-1817, 2006.

[178] S. Guennouni , A. Ahaitouf and A. Mansouri, "A Comparative Study of Multiple Object Detection Using Haar-Like Feature Selection and Local Binary Patterns in Several Platforms," *Modelling and Simulation in Engineering,* pp. 1687-5591, 2015.

[179] C. Wang, A. Bochkovskiy and H. M. Liao, "Scaled-YOLOv4: Scaling Cross Stage Partial Network," 2020.

[180] S. Liu, L. Qi, H. Qin, J. Shi and J. Jia, "Path Aggregation Network for Instance Segmentation," *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition,* pp. 8759-8768, 2018.

[181] Z. Zhang, "Improved Adam Optimizer for Deep Neural Networks," *2018 IEEE/ACM 26th International Symposium on Quality of Service (IWQoS),* pp. 1-2, 2018.

[182] S. Hochreiter and J. Schmidhuber, "Flat Minima," *Neural computation,* vol. 9, pp. 1-42, 1997.

[183] P. Tamilarasi and R. U. Rani, "Diagnosis of Crime Rate against Women using k-fold Cross Validation through Machine Learning," *2020 Fourth International Conference on Computing Methodologies and Communication (ICCMC),* pp. 1034-1038, 2020.

[184] S. Yadav and S. Shukla, "Analysis of k-Fold Cross-Validation over Hold-Out Validation on Colossal Datasets for Quality Classification," *2016 IEEE 6th International Conference on Advanced Computing (IACC),* pp. 78-83, 2016.

[185] S. S. Sandha, M. Aggarwal, I. Fedorov and M. Srivastava, "Mango: A Python Library for

Parallel Hyperparameter Tuning," *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP),* pp. 3987-3991, 2020.

[186] W. Alawad, M. Zohdy and D. Debnath, "Tuning Hyperparameters of Decision Tree Classifiers Using Computationally Efficient Schemes," *2018 IEEE First International Conference on Artificial Intelligence and Knowledge Engineering (AIKE),* pp. 168-169, 2018.

[187] V. Mehta, S. S. Katta, D. P. Yadav and A. Dhall, "Dif dataset of perceived intoxicated faces for drunk person identification," *2019 International Conference on Multimodal Interaction,* pp. 367-374, 2019.

[188] J. Lee, S. Choi and J. Lim, "Detection of high-risk intoxicated passengers in video surveillance," *2018 15th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS),* pp. 1-6, 2018.

[189] V. E. Neagoe and P. Diaconescu, "An ensemble of deep convolutional neural networks for drunkenness detection using thermal infrared facial imagery," *2020 13th International Conference on Communications (COMM),* pp. 147-150, 2020.

[190] Z. Bhango and D. T. van der Haar, "A model for inebriation recognition in humans using computer vision," *Business Information Systems,* pp. 259-270, 2019.

[191] C. F. Valenzuela, "Alcohol and neurotransmitter interactions," *Alcohol health and research world,* vol. 21, no. 2, pp. 144-148, 2020.

Appendix A – A Model for Inebriation Recognition in Humans Research Paper

A Model for Inebriation Recognition in Humans Using Computer Vision

Zibusiso Bhango and Dustin van der Haar

Cnr University Road and Kingsway Avenue, Academy of Computer Science and Software Engineering, University of Johannesburg, APK Campus, Johannesburg, 2006 zbhango@gmail.com, dvanderhaar@uj.ac.za

Abstract. The cost of substance use regarding lives lost, medical and psychiatric morbidity and social disruptions by far surpasses the economic costs. Alcohol abuse and dependence has been a social issue in need of addressing for centuries now. Methods exist that attempt to solve this problem by recognizing inebriation in humans. These methods include the use of blood tests, breathalyzers, urine tests, ECGs and wearables devices. Although effective, these methods are very inconvenient for the user, and the required equipment is expensive. We propose a method that provides a faster and convenient way to recognize inebriation. Our method uses Viola-Jones-based face-detection for the region of interest. The face images become input to a Convolutional Neural Network (CNN) which attempts to classify inebriation. In order to test our model's performance against other methods, we implemented Local Binary Patterns (LBP) for feature extraction, and Support Vector Machines (SVM), Gaussian Naive Bayes (GNB) and k-Nearest Neighbor (kNN) classifiers. Our model had an accuracy rate of 84.31% and easily outperformed the other methods.

Keywords: Computer Vision, Convolutional Neural Networks, Machine Learning, Inebriation Recognition, Support Vector Machines, k-Nearest Neighbor, Naive Bayes.

1 Introduction

Substance abuse has been a social issue in need of addressing for centuries now [8]. The human costs of substance use problems regarding lives lost, medical and psychiatric morbidity and social disruptions by far surpasses the economic costs [1]. Substances intercept and alter the messages going to the nervous system, resulting in altered perception. Usually, alcohol induces euphoria, relaxation or hyperventilation, thereby changing the mood of the drinker considerably. The euphoria is the feeling the user is after, and the user will continue consuming alcohol to keep getting the same effect. However, tolerance builds up swiftly; increased doses are required to satisfy the same level of effects, leading to dependence. When unattended, this can lead to an accidental fatal alcohol poisoning.

Depending on the type of alcohol and its effectiveness, it is often difficult to identify alcohol drinkers, especially in their early stages. Physical ways to detect

alcohol abuse include rapid heart rate, high blood pressure, poor muscle coordination, total mental confusion and dilated pupils, excessive sweating, among others [11]. However, as alcoholics continue drinking alcohol, many behavioural traits become more apparent. These include constant depression, introversion, lack of cleanliness and personal hygiene, valuable possessions going missing, intellectual ineptitude and lack of problem-solving skills [2].

More Americans die from substance overdose than they do in car accidents [3]. In order to tackle this social issue, there is a need for novel methods to gain more insight and combat abuse and addiction. The most common way of detecting alcohol abuse is by using a breathalyser. This method, although useful, is quite invasive and requires participation from the user. There are also legal implications that come with this approach. Although law enforcement officers have the right to breathalyse people, private citizens do not necessarily share that right. Due to the sensitive information captured by breathalysers, there are also ethical issues connected with it.

Due to these issues, there is a need for an alternative way to recognise inebriation. There's a need for a system that is faster, more convenient and generally accepted by people. The side effects of alcohol consumption make it possible to recognise drunkenness in an image or video. A biometrics system that recognises inebriation using computer vision will save time and effort, and is generally accepted by people. It will bring about real-time inebriation recognition faster, without effort from the user.

The rest of the paper aims to outline the problem at hand and compare methods used to recognize inebriation and provide the best-performing ones. The next sections are divided as follows. In problem background, we outline the problem at hand, which is how to detect inebriation in humans using computer vision. In related Work, we describe tried methods in the literature that tackle similar problems to ours. In the experimental setup, we describe our proposed approach in detail and explain the methods that we will use to preprocess, extract and classify features as either inebriated or sober. In results, we provide results on our methods and compare them against each other and others found in the literature. In conclusion, we provide our findings from the research and future work.

2 Problem Background

The human body handles adversity well, such as dealing with the injection of toxins and poisons. The human consumption of alcohol affects physical and cognitive functions and has legal consequences such as drunk driving and underage drinking. The average human body eliminates 12 grams of alcohol per hour [4]. Blood-Alcohol Concentration (BAC) is the most common metric used to measure the amount of alcohol in the human body at a given time, expressed in grams of alcohol per litre of body fluid. In countries such as Romania, the Czech Republic and Hungary, it is illegal to drive with any alcohol content in your system. For China, Estonia, Poland and Sweden, among others, it is illegal to

Fig. 1. *Some samples from the database used for data sampling to test our model. The top row consists of drunk individuals, while the bottom row consists of sober individuals.*

drive with a 0.02% BAC. In most Western European countries such as France, Germany and Greece, you're illegally driving under the influence if you have 0.05% BAC. The USA, New Zealand and the UK have a more lenient BAC of 0.08%. A BAC of 4 grams per litre is likely to result in a coma while a BAC of 4.5 – 5.0 grams is likely to result in death [4].

Globally, alcohol consumption leads to approximately 3.3 million deaths each year [5]. Excessive alcohol use is the third leading lifestyle-related cause of death in the United States [6]. It results in physical harm, mental malfunction and is responsible for 1 in 10 deaths among adults aged 20-64 years in the United States annually [7]. Drunk driving endangers the intoxicated driver as well as other sober drivers on the road. Despite these facts, binge drinking (which is defined as 4 or more drinks for women on a single occasion and 5 or more drinks for men on a single occasion) is still on the rise [7].

Some of the effects of alcohol include lower inhibitions, lower caution, loss of fine motor coordination and inability to do complex tasks or general problem-solving. Alcohol consumption also results in slurred speech, weakened balance, slow reaction times and staggering walk or inability to walk. Glossy appearance to eyes, blurry and double vision, loss of memory, heavy sweating, slower pupil response, slowed heart rate and breathing and reduced blood pressure can also result from drinking alcohol [11]. In some instances, nausea, vomiting or loss of consciousness can also occur.

Existing methods to detect alcohol consumption in people involve urine and saliva testing and using a breathalyzer. Expensive equipment is needed such as ones used to capture heart biosignals, infrared cameras or breathalyzers [8]. These methods are useful but very invasive. We believe there is a better way of recognizing inebriation, a way that is just as efficient but non-invasive and less inconvenient for both the subject and the one doing the inebriation testing.

When an individual is drunk, their appearance, the way they talk, walk or behave changes drastically, and it's possible to differentiate that using computer vision. Eye gaze, face pose and facial expression changes and these features can be used to differentiate between an image or video of a drunk person and that of a sober person.

3 Related Work

Since our method focuses on recognizing inebriation using computer vision, we looked at existing methods in the literature that tackled inebriation recognition and computer vision.

Aiello and Agu [7] developed a machine learning method to detect a drinker's Blood Alcohol Content (BAC) from their gait by classifying accelerometer and gyroscope sensor data collected from the drinker's smartphone. Alcohol-sensitive physical attributes such as weight, height and gender were taken into account when classifying. They used 34 intoxicated individuals (14 males and 20 females) for data sampling, and generated time and frequency domain features such as sway (gyroscope) and cadence (accelerometer). They used sensor-impairment goggles on the subjects to simulate the effects of alcohol on the body. Using this kind of special equipment is expensive and simulating intoxication effects limits the model's generalizability. They managed to implement feature normalization to account for differences in walking styles and automatic outlier elimination to reduce the effects of accidental falls. Their inebriation classifier had an accuracy of 72.66%.

Yadav and Dhall [8] proposed a new dataset called DIF (Dataset of Intoxicated Faces) containing RGB face videos of drunk and sober people obtained from online sources. 80 video samples were used, 30 being sober individuals and 50 being inebriated individuals. They analyzed the face videos to extract features related to eye gaze, face pose, and facial expressions. They implemented a convolutional neural network for feature extraction and a recurrent neural network to model the evolution of these multimodal facial features. The experiment showed that the eye gaze and facial features are discriminative for their dataset. They achieved 75.54% classification accuracy on the DIF dataset, thereby showing that face videos can be effectively used to detect drunkenness in humans.

Tseng and Jan [9] developed a unified deep learning network architecture that uses both semantic segmentation and object detection to detect people, cars, and roads simultaneously. They did this by creating a simulated environment in the Unity engine, which they used as a dataset. The simulated environment contained people, cars, roads, grass and the sky. They used the Single Short Multibox Detector (SSD), which enabled the network to detect objects of different sizes, making the predictions size-invariant and more accurate. Their proposed network performed end-to-end prediction well on the tested dataset, achieving 99.46% accuracy, although there are no details on the dataset used.

Al-Theiabat and Aljarrah [10] developed a motion analysis system which analyses tackle scenes in soccer games. They developed a computer vision system

to detect if a soccer player was intentionally falling to earn their team a free kick or penalty kick. The tackle scenes go through five stages of processing: identification of the falling player, extraction of tracking points, motion tracking, features extraction and scene classification. They tracked using Kanade-Lucas-Tomasi optical flow with the aid of pyramid levels and forward-backward error algorithm. They used 25 samples; 12 being actual fouls and 13 being dives. They executed classification using Weka software with Naive Bayes tree (NB tree) classifier. Their system had an 84% classification accuracy.

Computer vision research has been pursued for many years, but little work has been done on recognizing inebriation. Most research in the literature on substance abuse uses private datasets and expensive equipment such as sensors, which makes it difficult to measure the performance of algorithms used. We propose a system that will use a dataset made up of publicly available aggregated face images of inebriated and sober individuals gathered on the internet, and inexpensive equipment and focus on algorithms to improve on classifying inebriated and sober individuals.

4 Experimental Setup

4.1 Methodology

In our approach, we use the existing literature on computer vision such as [16] and inebriation recognition such as [8] to derive a model that uses computer vision to detect inebriation among individuals. We will use a secondary dataset to train our model and test its viability objectively. A prototype will be designed to test our model, and performance metrics used to test the performance of the prototype and its ability to detect inebriation in people are Accuracy, Precision, recall, f1 score, True Positive Rate (TPR), False Positive Rate (FPR) and Equal Error Rate (EER).

4.2 Data Sampling

In order to test the performance of our method, we used RGB face images consisting of inebriated and sober individuals collated by the authors. The dataset is made up of publicly available aggregated face images of inebriated and sober individuals gathered on the Internet.

Our dataset consists of 153 inebriated individuals and 101 sober individuals, both males and females. Images consist of people of various age groups. Since no datasets of inebriated and sober individuals exist, we are testing our classifier in the wild. We took images of reported inebriated individuals on the Internet, such as celebrities, and used their sober images and others to create our benchmark.

Fig. 2. *Input image before preprocessing (left). Resulting ROI image after implementing the Viola-Jones face detection algorithm [15] (right).*

5 Model

5.1 Preprocessing

Our model uses RGB images containing people as input. The first step in our classification process is extracting the Region of Interest (ROI), which is the face. The input image is converted to a grayscale format to remove noise. We use the Viola-Jones object detection algorithm [15] on the grayscaled image to get our ROI. The algorithm has four features: Haar feature selection, creating an integral image, AdaBoost training and cascading classifiers. Haar features contain what's common among people's faces, such as the eye region is darker than the cheeks and the nasal region being lighter than the eye region. Integral images, which allow integrals for the Haar extractors to be calculated by only adding four numbers, are used to improve efficiency. Face detection takes place inside a detection window. Adaboost training is used to train the algorithm. We test every window for face images using Haar-like features. We then use the windows containing the minimum error rates as the windows containing our face images. After Adaboost training, we use cascading classifiers to find the windows containing face images for classification. Cascading classifiers are split into classes, with each class containing fewer features to check than the next one. Each window is tested for face images using these classes. Only those images passing the test in each class are sent to other classes for further testing. If an image passes the final class, we have positively identified the face image.

5.2 Convolutional Neural Network

Convolutional Neural Networks (CNN's) have been widely used in computer vision for image or video recognition. CNN's, like any other neural network, are made up of neurons with learnable weight and bias. Each neuron receives several inputs, takes a weighted sum over them, passes it through an activation function

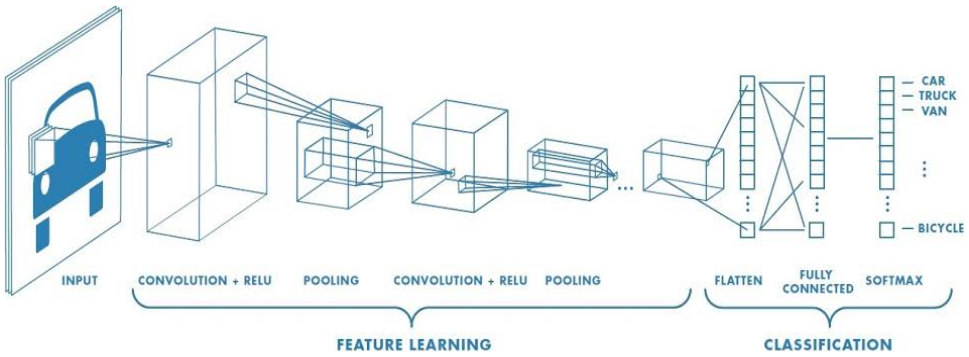

Fig. 3. *Convolutional Neural Network architecture, taken from [12].*

and outputs the result to another neuron. The entire network has a loss function, with the primary goal being to lower the loss function output and converge to a solution.

CNN's are made up of four layers: convolution, ReLU, pooling and the fully connected layer. This is what separates it from the other neural networks. In the convolution operation, we take a filter of a specific size and slide it over our image to get the dot product between the filter chunks of our image, resulting in a feature map. The feature map's pixels will be altered, and its size will also change. The convolution operation captures the local dependencies in the original image. After every convolution operation, a ReLU (Rectifier Linear Unit) is used. ReLU is an elementwise operation that replaces all negative values in the feature map with zero. Its purpose is to introduce non-linearity. The ReLU equation is as follows:

$$f(x) = x^+ = \max(0, x) \tag{1}$$

where x is the pixel in the feature map, pooling is used to reduce the dimensionality of each feature map while retaining the most essential information. This makes the feature maps dimension smaller and more manageable. We can perform convolution, ReLU and pooling operations multiple times. We then flatten our resulting feature maps and use them as input to a fully connected layer of the Neural Network.

In our implementation, we implemented a LeNet convolutional neural network because of its simplicity and efficiency. A 3x3 filter was used for convolution operation, and 32 filters in all were used. All face images went through preprocessing to detect faces and were turned back to RGB color images.The shape of the input image is (64,64,3). We chose a smaller filter size because a larger one can overlook the crucial features and miss them.

We used Max-pooling because it extracts most crucial information better than average pooling. For each region on the image represented by a filter, max-

pooling takes the most significant pixel of that region and create a new output matrix where each element is the maximum of a region in the original input.

After the convolution layer, we flattened the resulting image into a one-dimensional array and used this as input to a fully connected layer to train our network. We used cross-entropy to calculate our loss, and in the output layer, we used softmax as the activation function for binary classification. The architecture of our CNN is shown on Fig. 3.

5.3 Feature Extraction

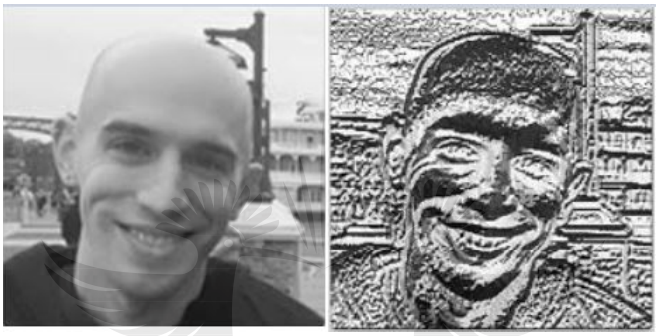

Fig. 4. *Grayscaled image (left). Resulting LBP image (right).*

In our alternative pipeline to compare against CNN, we used Local Binary Patterns (LBP's) for feature extraction. The LBP algorithm is rooted in 2D texture analysis. It works on the idea of summarising a local structure in an image by comparing each pixel to all of its neighbours. Each pixel is taken as a centre, each of the eight neighbourhoods is compared against the centre; a pixel with a higher value is converted to 1, and 0 if it's smaller. With 8 surrounding pixels, you end up with 2^8 possible combinations, commonly known as Local Binary Patterns.

After getting the LBP codes, a histogram is then generated from the resulting image. This histogram becomes our feature space, which is used for classifying images.

5.4 Classification

For classification, we used three classifiers: Support Vector Machines (SVM), Gaussian Naive Bayes (GNB) and k-Nearest Neighbor (kNN). SVM is a supervised machine learning algorithm used for classification and regression problems. It uses the kernel trick to transform your data, then based on those transformations, it finds the optimal boundary between the classes. SVMs work efficiently

in high dimensional spaces and use a subset of training points in the decision function (commonly known as support vectors), which is memory efficient. We used the Radial Basis Function (RBF) for training our model.

The Naive Bayes is a supervised learning algorithm which applies Bayes' theorem with the "naive" assumption of conditional independence between every pair of features given the class variable's value [13]. The classifier aggregates information using conditional probability and assumes independence among features. It is based on finding functions describing the probability of belonging to a specific class given features. Naive Bayes classifiers are extremely fast compared to other classification algorithms, and they do well to alleviate the curse of dimensionality. However, Naive Bayes classifiers are known to suffer from a weak assumption, making them bad estimators [13].

kNN is a learning algorithm which is most popular for classification purposes [14]. The idea behind kNN is to find a predefined number of training samples closest in distance to the new input, and predict the new input's class from these. Distance can be any metric measurement, although Euclidean distance is the most commonly.

6 Results

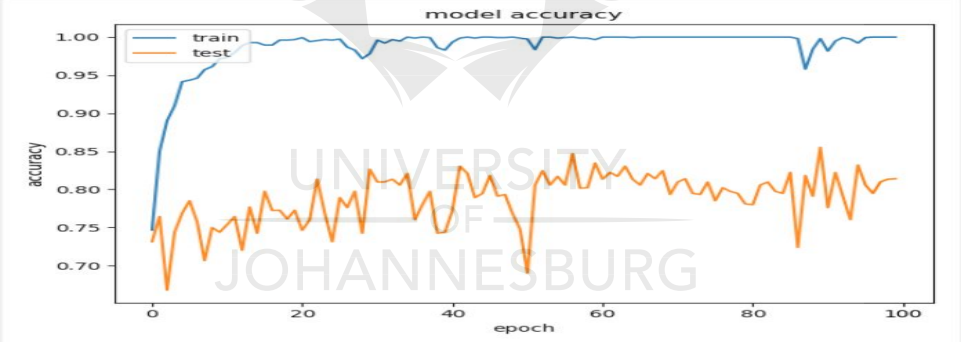

Fig. 5. *CNN model accuracy curve during training and testing.*

In order to measure the performance of our model, we calculated the accuracy, precision, recall, f1-score, equal error rate (EER) and Receiver Operating Characteristic (ROC) curve for each pipeline, including CNN. The ROC curve used to plot the True Positive Rate (TPR) versus the False Positive Rate (FPR) to measure the performance of the system when classifying inebriation. The Area Under the Curve determines whether the system performs well or not. The EER is when the FPR and the FNR are equal. Our model accuracy and model loss are shown in Fig. 5 and Fig.6, respectively.

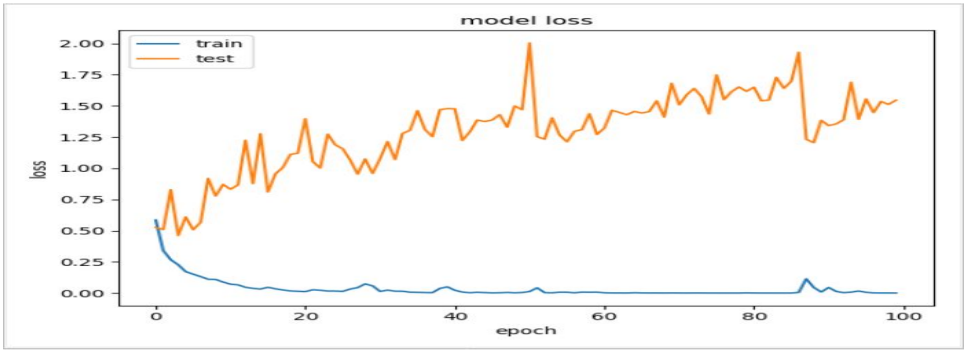

Fig. 6. *CNN model loss curve during training and testing.*

Our Convolutional Neural Network model had an accuracy of 84.31%. Our precision was 84.38%, the recall was 71.05%, f1-score of 77.14% and we achieved an EER of 22.2%. As shown in Table 1, our model outperforms the other pipelines (Gaussian Naive Bayes, k-Nearest Neighbor and Support Vector Machines), each with Local Binary Patterns used as features.

Table 2 shows the performance of our model against similar systems in the literature. We achieved higher accuracy and precision. Al-Theiabat and Aljarrah [10] achieved a higher recall and f1-score on a much smaller dataset consisting of 25 samples. Our recall and F1-score is very competitive, and we had a better EER than the rest.

Table 1. Comparing CNN with other classifiers

Method	Accuracy	Precision	Recall	F1-Score	EER	ROC Area
CNN	**84.31%**	**84.38%**	**71.05%**	**77.14%**	**22.21%**	**83%**
LBP-GNB	62.75%	50%	52.63%	51.28%	39.06%	63%
LBP-kNN	63.73%	51.35%	50%	50.67%	39.53%	64%
LBP-SVM	66.67%	60%	31.58%	41.38%	37.5%	66%

This proves that our model is feasible, and computer vision can be used to recognize inebriation using convolutional neural networks. We chose Local Binary Patterns because of how effective they are in texture analysis and their potential in differentiating facial appearances of inebriated individuals from those sober. However, Local Binary Patterns did not provide a distinct feature space for machine learning algorithms such as SVM, GNB and kNN to classify inebriation efficiently.

Table 2. Comparing our model with similar classifiers in literature.

Method	Accuracy	Precision	Recall	F1-Score	EER	ROC Area
CNN	**84.31%**	**84.38%**	71.05%	77.14%	**22.21%**	83%
Aiello and Agu [7]	72.66%	72.3%	72.7%	72.1%	-	**89.2%**
Al-Theiabat and Aljarrah [10]	84%	76.47%	**100%**	**86.7%**	-	-
Yadav and Dhall [8]	75.54%	-	76%	-	-	-

7 Conclusion

Substance abuse has taken many lives. It alters perception, and when unhandled, can lead to dependence. Its effects are rapid heart rate, high blood pressure, poor muscle coordination, total mental confusion, dilated pupils and excessive sweating, amongst others. One approach to combatting this is through inebriation detection.

Current methods of detecting inebriation, such as urine tests, blood tests, and breathing tests, using breathalyzers, using ECG to capture heart signals, fitness devices and wearable devices are effective but very inconvenient to the users. The equipment used is also costly.

In this paper, we proposed a model that uses computer vision to recognize inebriation. Only a camera is needed to achieve inebriation recognition, and there is no user participation required, such as breathing into a breathalyzer. Our model achieved excellent results, with an accuracy rate of 84.31%. We achieved results superior to the alternative implementations that use Local Binary Patterns with varying classifiers. We then compared our model to other similar models in literature and our model has higher accuracy.

Implementing a computer vision-based inebriation recognition system will make it easier and faster to detect inebriation, thereby increasing its application in other real-life problem domains. These domains include transportation, where we test drivers/pilots for alcohol use before embarking trips. In medicine, medical practitioners can be tested before diagonosing patients or performing surgeries. In social places, we can monitor customers in bars against excessive drinking. Such a system will reduce accidents and deaths, and can save people from alcohol dependence.

There is not much research done on using computer vision to detect inebriation or substance abuse and addiction. We believe our method is worthy of further research. Using deep neural networks such as Recurrent Neural Networks (RNN) to detect inebriation in videos can potentially improve on our model considerably. There is currently very little publicly available datasets of inebriated and sober individuals to test models with, and this provides uncertainty on whether a model is accurate or not. Developing a dataset that can be used to test algorithms will certainly improve inebriation recognition. Our

model's performance gives us the optimism that computer vision can indeed be used to recognize inebriation.

References

1. P. G. O'Connor and J. H. Samet, "Substance Abuse," Journal of General Internal Medicine, vol. 17, pp. 398-399, 2002.
2. NIDA, "Drugs, Brains, and Behavior: The Science of Addiction," 1 July 2014. [Online]. Available: https://www.drugabuse.gov/publications/drugs-brains-behavior-science-addiction/addiction-health. [Accessed 29 April 2018].
3. Y. Fan, Y. Zhang, Y. Ye, X. Li and W. Zheng, "Social Media for Opioid Addiction Epidemiology: Automatic Detection of Opioid Addicts from Twitter and Case Studies," in Proceedings of the 2017 ACM on Conference on Information and Knowledge Management, Singapore, 2017.
4. E. Mirielli and L. Webster, "Modeling Alcohol Absorption and Elimination from the Human Body: A Case Study in Software Development: Nifty Assignment," J. Comput. Sci. Coll., vol. 30, pp. 110-112, 2015.
5. M. K. Toroghi, W. R. Cluett and R. Mahadevan, "Multiscale Metabolic Modeling Approach for Predicting Blood Alcohol Concentration," IEEE Life Sciences Letters, vol. 2, pp. 59-62, 2016.
6. Z. Arnold, D. LaRose and E. Agu, "Smartphone Inference of Alcohol Consumption Levels from Gait," 2015 International Conference on Healthcare Informatics, pp. 417-426, 2015.
7. C. Aiello and E. Agu, "Investigating postural sway features, normalization and personalization in detecting blood alcohol levels of smartphone users," 2016 IEEE Wireless Health (WH), pp. 1-8, 2016.
8. D. P. Yadav and A. Dhall, "DIF : Dataset of Intoxicated Faces for Drunk Person Identification," ArXiv e-prints, 2018.
9. Y. H. Tseng and S. S. Jan, "Combination of computer vision detection and segmentation for autonomous driving," 2018 IEEE/ION Position, Location and Navigation Symposium (PLANS), pp. 1047-1052, 2018.
10. H. Al-Theiabat and I. Aljarrah, "A computer vision system to detect diving cases in soccer," 2018 4th International Conference on Advanced Technologies for Signal and Image Processing (ATSIP), pp. 1-6, 2018.
11. Solutions Recovery, "Physical Impact of Alcohol Abuse," [Online]. Available: https://www.solutions-recovery.com/alcohol-treatment/physical-impact/. [Accessed 5 January 2019].
12. Prabhu, Neural network with many convolutional layers. 2018.
13. A. H. Jahromi and M. Taheri, "A non-parametric mixture of Gaussian naive Bayes classifiers based on local independent features," 2017 Artificial Intelligence and Signal Processing Conference (AISP), pp. 209-212, 2017.
14. Okfalisa, I. Gazalba, Mustakim and N. G. I. Reza, "Comparative analysis of k-nearest neighbor and modified k-nearest neighbor algorithm for data classification," 2017 2nd International conferences on Information Technology, Information Systems and Electrical Engineering (ICITISEE), pp. 294-298, 2017.
15. P. Viola and M. J. Jones, "Robust Real-Time Face Detection," Int. J. Comput. Vision, vol. 57, pp. 137-154, 2004.
16. Y. LeCun, P. Haffner, L. Bottou and Y. Bengio, "Object Recognition with Gradient-Based Learning," Shape, Contour and Grouping in Computer Vision, pp. 319-345, 1999.

Appendix B – A Comparison of Deep Learning Methods for Inebriation Recognition in Humans Research Paper

A Comparison of Deep Learning Methods for Inebriation Recognition in Humans

No Author Given

No Institute Given

Abstract. Excessive alcohol consumption leads to inebriation. Driving under the influence of alcohol is a criminal offence in many countries involving operating a motor vehicle while inebriated to a level that renders safely operating a motor vehicle extremely difficult. Studies show that traffic accidents will become the 5th biggest cause of death if inebriated driving is not mitigated. Inversely, 70% of the world population can be protected by mitigating inebriated driving. Short term effects of inebriation include lack of balance, lack of inhibition, lack of fine motor coordination, dilated pupils and slow heart rate. An ideal inebriation recognition method operates in real-time, is less intrusive, more convenient, and efficient. Deep learning has been used to solve object detection, object recognition, object tracking, image segmentation along with context and scene understanding problems. In this paper we provide a comparison of deep learning inebriation recognition methods. We implemented Faster R-CNN and YOLO methods for our experiment. We created our own dataset of sober and inebriated individuals, which will be released. 920 images were used, and our best performing pipeline was YOLO with a 97.5% accuracy rate.

Keywords: Deep Learning, Computer Vision, R-CNN, YOLO, Inebriation Recognition, Inebriation detection, Drunk Driving

1 Introduction

Alcohol abuse is a social issue in need of addressing [3]. Inebriation is a temporary "sutiational impairment" affecting the subject's interaction with their environment [6]. Driving under the influence of alcohol is a criminal offense in many countries involving operating a motor vehicle after consuming alcohol beyond the legal limit [12]. A WHO study showed that unless measures to reduce inebriated driving are implemented, vehicle accidents will become the 5th leading cause of death globally [13]. Inversely, handling inebriated driving can save 70% of the population.

Heavy drinking for longer periods weakens the heart, resulting in a potential inability to pump adequate blood. Consistent excessive alcohol consumption can lead to irregular heartbeats or elevated blood pressure, leading to hypertension. After consuming alcohol, physical and physiological changes begin to take place, such as loss of inhibition, lower caution, loss of finer motor coordination and

inability to perform critical hand-coordinated tasks such as operating a motor vehicle [6]. Alcohol consumption also results in slurred speech, poor balance and lowed heart rate.

Due to these issues, it is important to recognise inebriation. The short-term effects of inebriation can be used to recognise inebriation in humans. An ideal inebriation recognition method is faster, non-invasive, convenient and effective. Deep learning methods have been used for object detection, semantic segmentation and object recognition problems. We believe they can be used effectively to recognise inebriation in humans.

The rest of the paper aims to outline the problem at hand and provide comparisons of deep learning methods in literature used to recognize inebriation. The next sections are divided as follows. In problem background, we outline the problem at hand, which is inebriation recognition in humans using deep learning. In similar work, we discuss deep learning methods in literature that tackle inebriation recognition. In experimental setup, we describe our reseach methodology, data sampling, population and benchmark for our study in detail. In implementation, we provide the implementation details of our deep learning pipelines. In results, we provide results on our methods and a comparison with methods in literature. In conclusion, we provide our findings from the research and future work.

2 Problem Background

Every 33 minutes, a person in the world is dying in a road accident instigated by an inebriated driver [15]. In 2014, about 27 people died daily due to inebriated driving in the United States of America [6]. An estimated 1.24 million people die on the road annually [1] and if this trend does not change, 2.4 million are expected to die on the road by 2030 because of inebriated driving [10]. Road accidents cost USD 500 billion a year, which is between 1% and 3% of the world's GDP [14].

10 minutes after alcohol consumption, the heart rate increases to filter the toxins from the blood stream to the kidneys. Initially, alcohol acts as a stimulant, producing intense feelings of warmth, well-being and relaxation; a feeling the alcohol consumer is after. However, inhibition, fine motor coordination and reaction time begin to suffer, with exponential effects experienced as more alcohol is consumed. The alcohol penetrates the blood-brain barrier, affecting the cognitive neuromotor functions, leading to loss of balance [1]. This renders operating a motor vehicle extremely difficult. Heavy alcohol consumption meddles with the delicate balance of neurotransmitters, which are in charge of the brain's functionality.

Due to the effects of inebriation, methods exist that can recognise inebriation in humans, such as direct and indirect biomarkers, automated biometric systems and several manual methods. Direct biomarkers include blood tests, urine tests and breath tests. These methods are very intrusive, and using them has legal and ethical conotations as blood and urine samples contain sensitive information

and take away privacy. Also, direct biomarkers can be inefficient, as use of a mouthwash with alcohol content can lead to a positive breath test. We believe there are better, faster and more efficient methods to recognise inebriation that are convenient and less invasive. These methods include automated biometrics systems that use deep learning to recognise inebriation. In this paper we will be providing a comparison of these methods in both our experiements and in literature.

3 Related Work

In this section we look at existing methods in literature that use deep learning for inebriation recognition.

Mehta et al. [7] developed a dataset called DIF (Dataset of perceived Intoxicated Faces) which contains audio-visual data of inebriated and sober individuals obtained from online sources. They argue that this is a novel approach for automatic bimodal non-invasive inebriation detection. They used CNN and Deep Neural Networks to compute the video and audio baselines, respectively. 3D CNN was used to exploit the spatio-temporal change in the video. They curated their dataset from social networks such as YouTube. The videos used include interviews, reviews and reaction videos involving inebriated individuals. Due to the nature of the collection procedure of their dataset, they tested their methods in the wild. Video title and caption from the website were used to assign class labels. They had 78 sober subjects and 88 inebriated subjects.

Their method was made up of 3 systems: the facial model, the audio model and the ensemble strategy. The facial model used the CNN-Recurrent Neural Network (RNN) and the variants of the 3D CNN methods. The audio model used a 2-layer perceptron with ReLU action function using batch normalisation and dropout. They also used the Long Short-Term Memory (LSTM) method as an alternative method for the audio model. The ensemble method uses both uniform and weighted average. Their method performed well, with the ensemble method being the best performing with an accuracy rate of 88.39%. Their research involved creating a dataset and making it publicly available, which is a very good contribution to literature. Using both the audio and visual data in a video to recognise inebriation makes the system more robust. However, the use of multimodal systems can be more computationally expensive.

Lee et al. [5] presented a method to detect abnormal behavior of inebriated individuals using surveillance videos. They looked for videos of people walking in zigzags, staggering and in a lying posture, which they argue proved potential inebriation. They argued that sober people walk straight and upright as opposed to walking in zigzags or staggering. They defined a motion efficiency as a feature to capture intoxicated motion and the aspect ratio of a bounding box of an object as a way to detect intoxicated postures. To compute these visual features, they used YOLOv3 for pedestrian detection and tracking, and motion trajectories and pose trajectories are evaluated from these detections.

25 videos were recorded and 48919 frames were used. They obtained 890 pedestrian trajectories using their tracking method. They had 846 sober trajectories and 44 inebriated trajectories. Their system achieved 93% recall and 98% precision rates. To the best of our knowledge, the authors did not provide their accuracy score. Their study had a high recall and precision, from which we can infer their accuracy and f1-score were very high as well. However, 95% of their dataset was of sober individuals, with 5% of the sample made up of inebriated individuals. We believe this level of data imbalance have an effect the experiment's results.

Neagoe and Diaconescu [9] developed a method to recognise inebriation using an ensemble of Deep Convolutional Neural Networks (DCNNs) for processing of thermal infrared facial images. Two modules were used: the first module had 12 layers and the second one had 10 layers. The two DCNNs were trained separately using different architectures and sets of parameters. The final decision is based on the confidence of the two CNN component modules. They evaluated the method using the dataset of 400 thermal infrared facial images belonging to 10 subjects. 40 images were used per subject, 20 sober and 20 taken 30 minutes after drinking 100ml of whisky. Their experiment had an accuracy rate of 95.75%, which is very high. We believe although 400 images are a relatively good sample, using 10 subjects is a very small sample to use, and might have an effect on their results. The usage of an ensemble methods is very robust, but it can also be computationally expensive as two deep neural networks are used independently.

Menon et al. [8] developed a system that captures a vehicle driver's face in thermal image spectrum. The face is recognised using a CNN then classified as inebriated or sober using Gaussian Mixture Model (GMM) with Fischer Linear Discriminant (FDA). They used capillary junction points on faces to determine differences in blood temperature used for inebriation recognition. 41 subjects were used, with samples captured while sober and also after consuming alcohol. Their face recognition algorithm had a 97% accuracy. The face image's dimensionality is reduced using FDA and classified as sober or inebriated using GMM. Their classifier had an 87% classification rate, which is a good performance. We believe 41 samples are too small for generalising a classifier, aqnd the sensor used to capture thermal imaging is expensive.

Bhango and van der Haar [2] developed a method to recognise inebriation in humans using computer vision. They used RGB face images made up of 153 inebriated subjects and 101 sober subjects. Their dataset was collated from publicly available face images of inebriated and sober individuals gathered on the internet. Their method used Viola-Jones-based face-detection for the region of interest localisation. The localised face images become input to a LeNet CNN algorithm which classified inebriation. They also implemented a traditional pipeline comprising of Local Binary Patterns (LBP) for feature extraction, and Support Vector Machines (SVM), Gaussian Naive Bayes (GNB) and k-Nearest Neighbor (kNN) classifiers for inebriation classification to compare their inebriation recognition methods. Their CNN model had an accuracy rate of 84.31%, which was superior to their traditional methods. This was a very good performance, but

we believe their dataset was small for the generalisation of their classifier. Also, their recall was low which indicates that a high number of inebriated cases was classified as sober.

4 Experimental Setup

4.1 Methodology

In our study, we used the design science research methodology. We believe the design science methodology is best suited for IT artefacts, is rooted in engineering and has been generally accepted by IT practitioners [4]. In particular, we used the design-oriented approach which aims to develop and provide an artefact as a research contribution or output [11].

The design-oriented method is made up of four phases: analysis, initial design, evaluate and validate & diffuse. During analysis, our research problem is identified and ojectives formulated. During initial design, the artefact is designed to the generally accepted methods. In the evaluate phase, the artefact is produced against the objectives in the analysis phase. During validate & diffuse, validation of the artefact takes place. In our research, we used the quantitative research design because our research is best suited for numerical data that can be statistically analysed to offer comparison of deep learning inebriation recognition methods.

4.2 Data Sampling

Fig. 1. *Some samples from the database used for data sampling to test our experiments. The top row consists of drunk individuals, while the bottom row consists of sober individuals.*

For data sampling, we could not find a dataset in literature for inebriated and sober individuals, therefore, in our research we are testing in the wild. We created a dataset of sober and inebriated people by collating our data from publicly accessible images on the internet with free usage rights. 230 images were collated and data augmentation performed on each image by rotating the image to the left and right within the 30-degree range and performing a horizontal flip. This resulted in 920 images, and 1000 face images, making up our dataset. The augmentation results are shown in Figure 2 below.

Fig. 2. *An augmented sample of a sober individual. Far left is the original image. Middle left is the horizontal flip of the original image. Middle right is the right rotation of the original image and far right is the left rotation of the original image.*

For population, our dataset is made up of randomly selected people from different walks of life. It is made up of inebriated and sober individuals and is used for inebriation recognition. 392 face images of sober individuals and 608 face images are of inebriated individuals. We used 799 face images for training, with the remaining 201 face images used for testing.

For benchmarking, we used the accuracy, precision, recall, f1 score and the area under the curve metrics to measure the performance of our experiments. We also used these metrics to compare our experiments' performance against deep learning methods on inebriation recognition in literature.

5 Implementation

For our experiment, we implemented the Faster R-CNN and YOLO methods. These methods are discussed in the sections below. We also implemented traditional biometrics methods to recognise inebriation. These methods will not be discussed here, but we will provide a comparison of their results vs the deep learning methods discussed in this section.

5.1 Faster R-CNN

Unlike it's predecessors, Faster R-CNN does not use the selective search algorithm for region proposals, but instead let's the region proposal network (RPN) learn the region proposals on its own. Faster R-CNN consists of two modules: a deep fully connected CNN for region proposals and a fast R-CNN detector using SVMs for object classification. The RPN is a fully connected layer used to predict object bounds and object probability scores at each position. The RPN is trained to generate very good proposals by minimising the classification and regression loss. RPN ranks region boxes called anchors, and proposes the ones flagged as containing objects.

We used Roboflow's Faster R-CNN implementation based on the tensorflow object recognition API. We used the COCO dataset for transfer learning. The Inception v2 CNN algorithm was used as our architecture because of its high accuracy performance. We used a batch size of 12 for training and L2 regularisation becasue of its ability to force weights to be small but not zero.

5.2 YOLO

You Only Look Once (YOLO) is an object recognition algorithm that uses a CNN to predict the confidence of the bounding boxes and the class probability for these boxes using an entire image. Object recognition is treated as a regression problem. The method excels at object recognition in real-time.

During training, an image is divided into grids of NxN cells, each cell responsible for predicting A possible bounding boxes, which are rectangles surrounding the detected object. These bounding boxes have 5 values, namely x, y, w, h and cs. x and y are coordinates of the centre of the bounding box relative to the grid. W and h are width (w) and height (h) of the bounding box relative to image dimension, and cs is the confidence value determining the confidence of the network about the object's presence inside the bounding box. The confidence value does not contain object class information.

For our experiment, we used the recently released YOLOv5 method. The YOLOv5 method consists 3 important parts: model backbone, model neck and model head. The model backbone is essential for the extraction of import features from an image. We used Cross Stage Partial Networks as our model backbone and it improved our processing time. Model neck is used to generate feature pyramids which help the model generalise well on scaled images. We used path aggregation network (PANet) as a model neck. Model head is used for final object recognition by applying anchor boxes on features and creating final output vectors containing class probabilities.

The Leaky ReLU (rectifier linear unit) and sgmoid activation functions are used. ReLU is an activation function that returns 0 for any negative input and the actual value for any positive value. The sigmoid activation function maps the range of input into [0;1]. ReLU is used in the hidden layers and the sigmoid function is used in the final recognition layer. For learning optimiser, we used

the stochastic gradient descent (SGD). SGD computes the gradient of the network's loss function with respect each individual weight in the network. We used Pytorch's Binary Cross-Entropy with Logits Loss for loss classification of class probability and object score.

6 Results

In this section, we will discuss our results from our experiments. The pipelines' results are discussed below.

6.1 Faster R-CNN

For functional requirements, we looked at pipeline's face detection and inebriation classification. Our Faster R-CNN pipeline performed well, showing very good results. All face images used for testing our model were successfully detected. In Figure 3 below are a few face images that were both detected and classified by our Faster R-CNN pipeline, including augmented images.

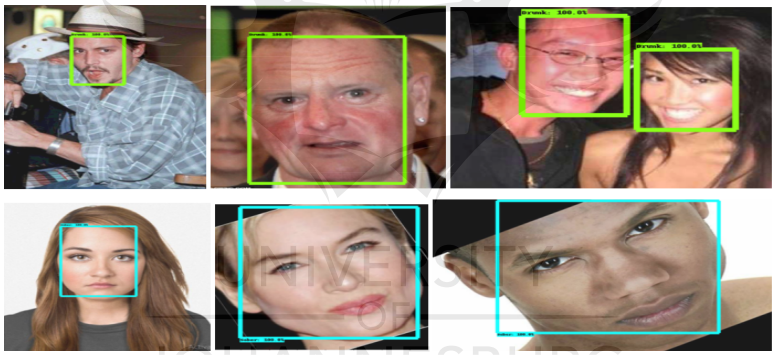

Fig. 3. *Augmented face images detected and classified by our algorithm.*

After obtaining results from our pipeline, we obtained the confusion matrix shown in Figure 4:

We used 201 face images for inferences. 69 individuals were correctly predicted as sober and 117 individuals were correctly predicted as inebriated. However, 15 individuals were wrongly predicted as inebriated. No individuals were wrongly predicted as sober. Some of the misclassifications are shown in Figure 5.

Of the 15 misclassified cases, 12 cases are augmented face images. Although our model performed well on most augmented images, we believe it significantly struggled with them. The original face images performed significantly better compared to their augmented counterparts as shown in the Figure 5. Moreover,

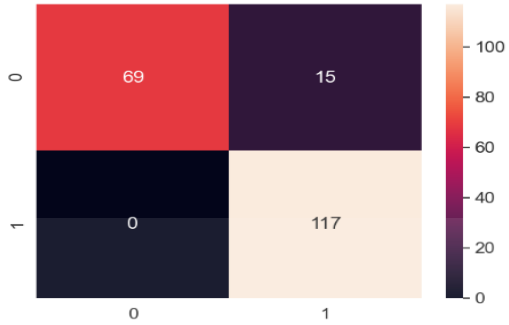

Fig. 4. *: The Confusion Matrix for the Faster R-CNN pipeline.*

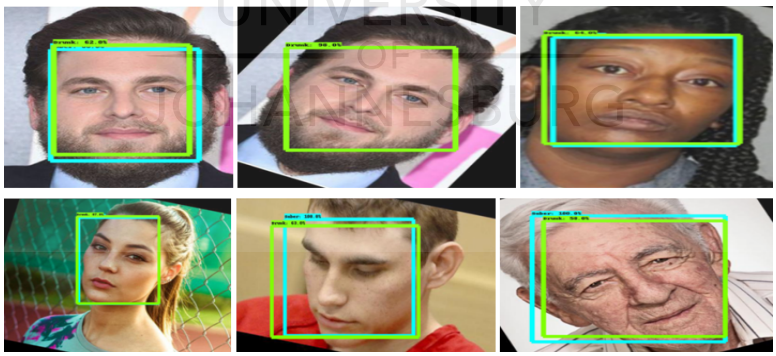

Fig. 5. *Misclassified images.*

14 of the 15 misclassified cases also have the correct classification detected as well.

We calculated our accuracy, precision, recall and f1-score from the confusion matrix. These metrics are listed Table 1:

Table 1. Faster R-CNN metric results.

Metrics	Accuracy
Accuracy	92.5%
Precision	88.6%
Recall	100%
AUC	91.1%
F1-Score	94%

Our prototype had an accuracy of 92.5%, precision of 88.6%, recall of 100%, f1-score of 94% and area under the curve (AUC) of 91.1%. The precision is low because of the sober cases misclassified as inebriated. Our recall is very high because there were no misclassifications on inebriated cases. We believe this is a good performance on predicting inebriation in humans.

6.2 YOLO

For functional requirements, we looked at our pipeline's performance in both face detection and inebriation classification. Our YOLO pipeline performed well, showing state-of-the-art results. All face images were successfully detected and classified. Figure 6 shows a sample of face images that were both detected and classified by our YOLO pipeline, including augmented images.

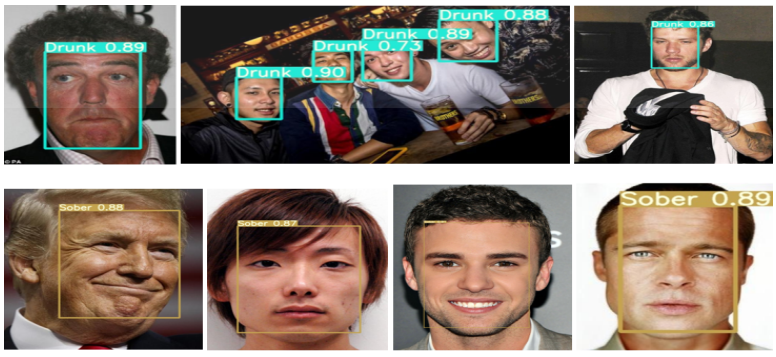

Fig. 6. *Face images detected and classified by YOLO.*

After obtaining results from our pipeline, we obtained the confusion matrix shown in Figure 7:

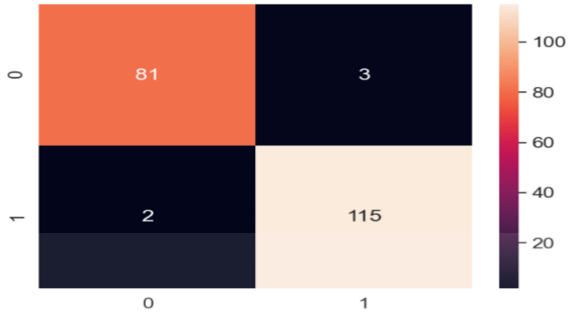

Fig. 7. *:The Confusion Matrix for the YOLOv5 pipeline.*

We used 201 face images for inferences. 81 individuals were correctly predicted as sober and 115 individuals were correctly predicted as inebriated. However, 3 individuals were wrongly predicted as inebriated and 2 individuals were wrongly predicted as sober. The 5 misclassifications are shown in Figure 8.

Fig. 8. *The misclassification cases for YOLO.*

All the 5 misclassification cases are augmented face images. Although our model performed well on most augmented images, we believe it struggled with them as well. Also, 60 percent of the misclassified cases also have the correct classification detected as well, as shown in Figure 8.

We calculated our accuracy, precision, recall and f1-score from the confusion matrix. These metrics are listed Table 2:

Table 2. YOLOv5 metric results.

Metrics	Accuracy
Accuracy	97.5%
Precision	97.9%
Recall	97.5%
AUC	98.3%
F1-Score	97.4%

Our prototype had an accuracy of 97.5%, precision of 97.9%, recall of 97.5%, f1-score of 97.4% and area under the curve (AUC) of 98.3%. Our precision and recall are very high because of low misclassification rate on both classes. The low misclassification rates on sober and inebriated cases means our F1 score is very high. These are great results because a high F1-score means the system can be used in various environments and use cases that might require high precision, high recall or both. We believe this is a state-of-the-art performance on recognising inebriation in humans.

Table 3 compares our methods against each other to show which pipeline had the better results. In our experiment we also implemented traditional pipelines namely: local binary patterns with support vector machines classifier (LBP-SVMs), local binary patterns with gradient boosted trees classifier (LBP-GBT), local binary patterns with random forests clasifier (LBP-RF), histogram of gradients with support vector machines classifier (HOG-SVM), histogram of gradients with gradient boosted trees (HOG-GBT) and histogram of gradients with random forests classifier (HOG-RF). We also compare these methods for completeness.

Table 3. A comparison of YOLOv5 vs Faster R-CNN and the traditional pipelines.

Method	Accuracy	Precision	Recall	F1-Score	AUC
YOLOv5	**97.5%**	**97.9%**	97.5	**97.4%**	**98.3%**
Faster R-CNN	92.5%	88.6%	**100%**	94%	91.1%
LBP-SVM	79.6%	78.4%	89.7%	83.7%	77.6%
LBP-GBT	78.1%	76.2%	90.1%	82.3%	75.7%
LBP-RF	79.1%	77.45%	90.1%	83.4%	76.8%
HOG-SVM	76.3%	73.8%	84.6%	78.9%	71.5%
HOG-GBT	72.6%	71.2%	88.9%	69.4%	79.1%
HOG-RF	72.6%	69.4%	94.9%	80.1%	68.3%

YOLOv5 and Faster R-CNN perfomed better than the traditional methods. The traditional methods struggled with augmented images. There's silimarities in the misclassification cases. Traditional pipelines had a low precision because of a high misclassification of sober individuals as inebriated. Although both YOLOv5 and Faster R-CNN had high performance, YOLOv5 performed extremely better and produced state-of-the-art results in inebriation recognition in humans. YOLOv5 had a lower misclassification rate than Faster R-CNN.

Table 4 provides comparison on the performance of our experiments against similar systems in literature. Our YOLOv5 pipeline achieved higher accuracy, F1-score and AUC. Our Faster R-CNN had the highest recall. Lee et al. [5] achieved a higher precisionbut their accuracy, f1-score and AUC metrics were not provided on their study. We believe our YOLOv5 pipeline had the best results based on the Table 5 comparison.

Table 4. Comparing our model with similar classifiers in literature.

Method	Data Sampling	Accuracy	Precision	Recall	F1-Score	AUC
YOLOv5	**920**	**97.5%**	97.9	97.5	**97.4%**	**98.3%**
Faster R-CNN	**920**	92.5%	88.6%	**100%**	94%	91.1%
Mehta et al. [7]	166	88.39%	85%	99%	-	-
Lee et al. [5]	890	-	**98%**	93%	-	-
Neagoe and Diaconescu [9]	400	95.75%	-	-	-	-
Menon et al. [8]	41	87%	-	-	-	-
Bhango and van der Haar [2]	153	84.31%	84.38%	71.05%	77.14%	83%

7 Conclusion

Excessive alcohol consumption lead to inebriation. Alcohol abuse is a serious social issue in need of solutions. According to a study by WHO, vehicle accidents will become the 5th highest cause of death if inebriated driving is not mitigated [31]. Inversely, 70% of the world population can be protected by handling drunk driving effectively. Drinking too much alcohol has a heavy impact on our health. Heavy alcohol consumption meddles with the delicate balance of neurotransmitters.

Our research shows that there is enough feature separability between sober and inebriated individuals to separate them. Our experiment showed that both our YOLOv5 and Faster R-CNN implementations on ineberiation recognition produced state-of-the-art results. We compared results which proved our experiments to be superior to the ones in literature. Traditional biometrics algorithms

struggled to separate inebriation and sober individuals. They especially struggled with precision – recognising sober individuals as sober. In contrast, our deep learning algorithms are high performing at recognising inebriation in humans. Although Faster R-CNN struggled with precision, the recognition performance of both Faster R-CNN and YOLOv5 methods was very high.

To the best of our knowledge, no publicly available datasets on inebriated and sober face images that can be used to classify inebriation exists. Making our dataset publicly available for public use will play a role in facilitating the growth of inebriation recognition using deep learning methods. YOLOv5 and Faster R-CNN struggled with recognising augmented face images as opposed to normal images. Most of our misclassification cases were related to augmented images. However, adding augmented images for training vastly improved the deep learning pipelines' overall classifier performance.

For future work, we believe a scientifically proven dataset consisting of inebriated and sober individuals will be of immense help for benchmarking inebriation recognition methods. Different types of alcohol have varying effects when consumed. Our experiments compared methods that recognise inebriation, but not the cause of inebriation. An algorithm that can detect the type of intoxicating substance that caused an individual to be inebriated will be a big contribution in literature.

References

1. Arnold, Z., Larose, D., Agu, E.: Smartphone inference of alcohol consumption levels from gait. In: 2015 International Conference on Healthcare Informatics. pp. 417–426 (2015)
2. Bhango, Z., van der Haar, D.: A model for inebriation recognition in humans using computer vision. In: Abramowicz, W., Corchuelo, R. (eds.) Business Information Systems. pp. 259–270. Springer International Publishing, Cham (2019)
3. Goffredo, M., Bouchrika, I., Carter, J.N., Nixon, M.S.: Performance analysis for gait in camera networks. In: Proceedings of the 1st ACM Workshop on Analysis and Retrieval of Events/Actions and Workflows in Video Streams. pp. 73–80. AREA '08, ACM, New York, NY, USA (2008), http://0-doi.acm.org.ujlink.uj.ac.za/10.1145/1463542.1463555
4. Gregor, S., Hevner, A.: Positioning and presenting design science research for maximum impact. MIS Quarterly 37, 337–356 (06 2013)
5. Lee, J., Choi, S., Lim, J.: Detection of high-risk intoxicated passengers in video surveillance. In: 2018 15th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS). pp. 1–6 (2018)
6. Mariakakis, A., Parsi, S., Patel, S.N., Wobbrock, J.O.: Drunk user interfaces: Determining blood alcohol level through everyday smartphone tasks. In: Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems. p. 1–13. CHI '18, Association for Computing Machinery, New York, NY, USA (2018), https://doi.org/10.1145/3173574.3173808
7. Mehta, V., Katta, S.S., Yadav, D.P., Dhall, A.: Dif dataset of perceived intoxicated faces for drunk person identification. In: 2019 International Conference on Multimodal Interaction. p. 367–374. ICMI '19, Association for Computing Machinery, New York, NY, USA (2019), https://doi.org/10.1145/3340555.3353754

8. Menon, S., J., S., S.K., A., Nair, A.P., S., S.: Driver face recognition and sober drunk classification using thermal images. In: 2019 International Conference on Communication and Signal Processing (ICCSP). pp. 0400–0404 (2019)
9. Neagoe, V.E., Diaconescu, P.: An ensemble of deep convolutional neural networks for drunkenness detection using thermal infrared facial imagery. In: 2020 13th International Conference on Communications (COMM). pp. 147–150 (2020)
10. Neagoe, V.E., Carata, S.V.: Drunkenness diagnosis using a neural network-based approach for analysis of facial images in the thermal infrared spectrum. pp. 165–168 (06 2017)
11. Oesterle, H., Becker, J., Hess, T., Karagiannis, D., Krcmar, H., Loos, P., Mertens, P., Oberweis, A., Sinz, E.: Memorandum on design-oriented information systems research. http://www.alexandria.unisg.ch/Publikationen/71089 20 (01 2011)
12. Wang, W.F., Yang, C.Y., Wu, Y.F.: Svm-based classification method to identify alcohol consumption using ecg and ppg monitoring. Personal and Ubiquitous Computing 22 (04 2018)
13. Wu, C.K., Tsang, K.F., Chi, H.R.: A wearable drunk detection scheme for healthcare applications. In: 2016 IEEE 14th International Conference on Industrial Informatics (INDIN). pp. 878–881 (2016)
14. Wu, C., Tsang, K., Chi, H., Hung, F.: A precise drunk driving detection using weighted kernel based on electrocardiogram. Sensors 16, 659 (05 2016)
15. Wu, Y.c., Xia, Y.q., Xie, P., Ji, X.w.: The design of an automotive anti-drunk driving system to guarantee the uniqueness of driver. Proceedings - 2009 International Conference on Information Engineering and Computer Science, ICIECS 2009 62 (12 2009)