INSTITUTO SUPERIOR DE ENGENHARIA DO PORTO
MESTRADO EM ENGENHARIA INFORMÁTICA




Sistema de reconhecimento de expressões faciais para deteção de stress

JOSÉ PAULO DE SOUSA ALMEIDA
Outubro de 2020

POLITÉCNICO
DO PORTO

Instituto Superior de Engenharia do Porto

Facial Expression Recognition System

Stress Detection

José Paulo de Sousa Almeida

A dissertation submitted in partial fulfilment of the requirements for the degree of Master of Computer Engineering, Specialisation Area of Information and Knowledge Systems

Supervisor: Doutora Fátima Rodrigues

Porto, October 2020

“Discipline does not fail me, I am the one who fail discipline.
Discipline Equals Freedom”

Jocko Willink

Abstract

Stress is the body's natural reaction to external and internal stimuli. Despite being something natural, prolonged exposure to stressors can contribute to serious health problems. These reactions are reflected not only physiologically, but also psychologically, translating into emotions and facial expressions.

Once this relationship between the experience of stressful situations and the demonstration of certain emotions in response was understood, it was decided to develop a system capable of classifying facial expressions and thereby creating a stress detector.

The proposed solution consists of two main blocks. A convolutional neural network capable of classifying facial expressions, and an application that uses this model to classify real-time images of the user's face and thereby verify whether or not it shows signs of stress.

The application consists in capturing real-time images from the webcam, extract the user's face, classify which facial expression he expresses, and with these classifications assess whether or not he shows signs of stress in a given time interval. As soon as the application determines the presence of signs of stress, it notifies the user.

For the creation of the classification model, was used transfer learning, together with fine-tuning. In this way, we took advantage of the pre-trained networks VGG16, VGG19, and Inception-ResNet V2 to solve the problem at hand. For the transfer learning process, were also tried two classifier architectures.

After several experiments, it was determined that VGG16, together with a classifier made up of a convolutional layer, was the candidate with the best performance at classifying stressful emotions. Having presented an MCC of 0.8969 in the test images of the KDEF dataset, 0.5551 in the Net Images dataset, and 0.4250 in the CK +.

Keywords: Stress, Stress Detection, Emotion, Facial Expression Classification, Convolutional Neural Networks

Resumo

O stress é uma reação natural do corpo a estímulos externos e internos. Apesar de ser algo natural, a exposição prolongada a *stressors* pode contribuir para sérios problemas de saúde. Essas reações refletem-se não só fisiologicamente, mas também psicologicamente. Traduzindo-se em emoções e expressões faciais.

Uma vez compreendida esta relação entre a experiencia de situações stressantes e a demonstração de determinadas emoções como resposta, decidiu-se desenvolver um sistema capaz de classificar expressões faciais e com isso criar um detetor de stress.

A solução proposta é constituida por dois blocos fundamentais. Uma rede neuronal convolucional capaz de classificar expressões faciais e uma aplicação que utiliza esse modelo para classificar imagens em tempo real do rosto do utilizador e assim averiguar se este apresenta ou não sinais de stress.

A aplicação consiste em captar imagens em tempo real a partir da webcam, extrair o rosto do utilizador, classificar qual a expressão facial que este manifesta, e com essas classificações avaliar se num determinado intervalo temporal este apresenta ou não sinais de stress. Assim que a aplicação determine a presença de sinais de stress, esta irá notificar o utilizador.

Para a criação do modelo de classificação, foi utilizado *transfer learning*, juntamente com *fine-tuning*. Desta forma tirou-se partido das redes pre-treinadas VGG16, VGG19, e Inception-ResNet V2 para a resolução do problema em mãos. Para o processo de *transfer learning* foram também experimentadas duas arquiteturas de classificadores.

Após várias experiencias, determinou-se que a VGG16, juntamente com um classificador constituido por uma camada convolucional era a candidata com melhor desempenho a classificar emoções stressantes. Tendo apresentado um MCC de 0,8969 nas imagens de teste do conjunto de dados KDEF, 0,5551 no conjunto de dados Net Images, e 0,4250 no CK+.

Palavras-chave: *Stress*, Deteção de Stress, Emoção, Classificação de Expressões Faciais, Rede Neuronal Convolucional

Acknowledgement

I want to start by thanking my supervisor, Dr Fátima Rodrigues, first, for giving me the opportunity to work on this project, and second, for all the help and teachings.

A second thanks go to my family, who have always supported me throughout this journey, and who have always been there to motivate and encourage me to overcome all adversities.

I also want to thank all my colleagues and friends who accompanied me during this phase of my life, for all the shared friendship and all the memorable moments we lived together. Without them, this journey would not have half the fun.

I cannot forget, a special thanks to the Instituto Superior de Engenharia do Porto, as well as to all its professors, because thanks to your dedication and commitment, I was able to learn and grow for a lifetime of opportunities to carry out engineering works.

Contents

List of Figures

List of Tables

List of Acronyms

AAM	Active Appearance Model
AU	Action Unit
CNN	Convolution Neural Network
CRISP-DM	Cross Industry Standard Process for Data Mining
DNN	Deep Neural Network
EU-OSHA	European Agency for Safety and Health at Work
FACS	Facial Action Coding System
GAP	Global Average Pooling
K-NN	K-Nearest Neighbour
LDA	Linear Discriminant Analysis
MCC	Matthews Correlation Coefficient
NIR	Near-Infrared
rPPG	Remote Photoplethysmography
SDM	Supervised Descent Method
SIFT	Scale-invariant Feature Transform
SVM	Support Vector Machines
USA	United States of America

1 Introduction

This chapter presents the project context, the problem to be addressed, and the objectives to achieve with this document.

1.1 Context

Every day people communicate with each other. Not only verbally, but also from gestures and facial expressions. Often these gestures and facial expressions are automatic, and the transmitter does not even realize that he is executing them.

This unintentional information is the primary way to know the transmitter's mood in a non-invasive way. For example, when a person smiles at the same time as the lower eyelid show wrinkles below it, it means they are happy, just like many other emotions.

If a facial expression is expressed by an acquaintance or if it is included in a conversation, it can seem quite intuitive to humans to interpret those emotions. However, when the context is not known, it is quite complex to determine precisely what a facial expression or a gesture means.

Like the other most common emotions, stress can be manifested from facial expressions, gestures, and even from the voice.

Demanding jobs are a significant cause of stress in people. Situations like frequent exposure to danger, short deadlines, or rigorous tasks are some of the originators. Different people will have different rupture points, depending on their ability to deal with these external stressors. Some will show signs of stress earlier, others more expressive signs, but all will react to stress.

According to an opinion pool made in 2014 to workers from all over Europe, 53% considered stress as one of the main health and safety risks they face in their workplace. In this same study, 27% also stated that in the last 12 months they suffered from stress, depression, or anxiety due to work (TNS Political & Social, 2014).

Constant exposure to stress is already known as a source of health problems, from cardiovascular illnesses to depression and exhaustion (also known as burnout). In addition to health problems, stress can also have consequences for employers. They may see declines in the productivity of their workers, as well as increases in absenteeism and presenteeism.

According to studies by the European Agency for Safety and Health at Work (EU-OSHA), work-related depression costs Europe about 600 billion euros annually (Cosmar et al., 2014).

About 61% of European establishments participating in the 2019 EU-OSHA study (*ESENER 3*, 2019) reported that a reluctance to talk openly about these issues seems to be the main difficulty for addressing psychosocial risks.

Not only in Europe but also in the United States of America (USA) the mental illness is on the rise, indicating that hundreds of thousands of Americans live with serious psychological distress(Thompson, 2017).

In a survey by the American Psychological Association in 2018 to adults living in the USA, 74% indicated that in the last month, stress had impacted their lives at least once. Almost half of these adults say they lay awake at night (45%), overeat or eat unhealthy due to stress (American Psychological Association, 2018).

Another relevant point is digitization. Eight out of ten companies have personal computers, laptops or mobile computing devices in the work environment, which are used by their workers(*ESENER 3*, 2019).

1.2 Problem

Stress is difficult to detect, largely because it is something progressive that comes from an accumulation of extreme situations. As such, behavioural changes will also be gradual and subtle. Even more difficult due to the lack of openness from workers to talk about their psychosocial problems.

Due to its late detection, there are a large number of cases of depressions, burnouts, loss of productivity, among others.

1.3 Objectives

The purpose of this dissertation is to create a system capable of detecting signs of stress from images of people's facial expressions. For this, the main objectives to be achieved will be:

- Study and understanding the influence of stress on facial expressions;

- Analysis of images taken from stressful situations and the extraction of the main characteristics influenced by stress;
- Evaluate and determine the best machine learning technique, along with the best set of characteristics for stress detection;
- Create a system capable of detecting signs of stress.

1.4 Expected Results

Is expected a system capable of running on the background, that through user's face images, detects if that individual is showing signs of stress. In a positive case, notify the user, so he can act upon that stress.

1.5 Methodology

The methodology to be adopted is the Cross Industry Standard Process for Data Mining (Wirth & Hipp, 2000), CRISP-DM for short. This methodology provides guidelines for the Data Mining process, breaking it down into six phases, as shown in Figure 1.

Using the CRISP-DM, will be taken a cyclical approach where the collected images will be analysed and pre-processed for corrections and disposal of images not relevant to the solution to be developed.

When the images are ready, will then be built classification models, using Convolution Neural Networks. These, in turn, will be evaluated using hold-out techniques. Based on the knowledge taken from this iteration, it may be necessary to repeat the whole process until are attained the ideal classification model.

Once satisfied with the built models, they will be deployed, so it can be used in use cases and consequently validate the solution's operation.

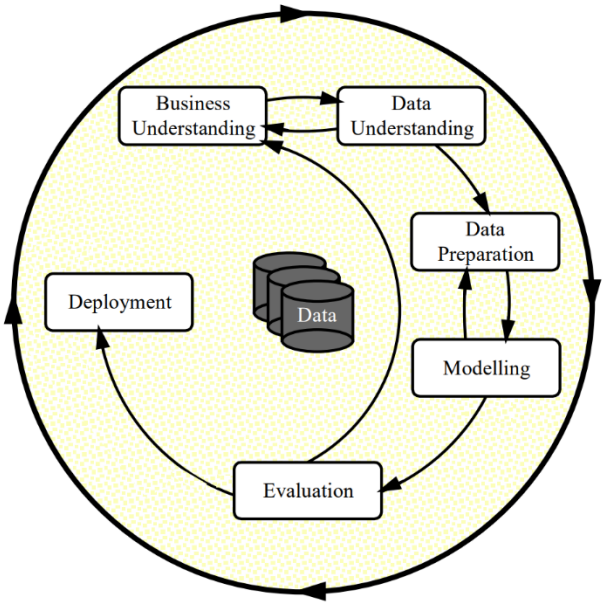

Figure 1 – Phases of the CRISP-DM Process Model for Data Mining. Source: (Wirth & Hipp, 2000)

1.6 Document Structure

This document is divided into five chapters. The first chapter presents the context in which this project is inserted, what is the problem intended to solve, what objectives to achieve, as well as the expected results and the approach to adopt.

In the second chapter, will be studied the key concepts such as emotions, facial expressions and stress. Will then be addressed the techniques and technologies commonly used in Computer Vision, followed by the datasets to be used for this project and a state of the art of other existing approaches.

In the third chapter, it will be described the design of the solution and the decisions taken to implement the program.

The fourth chapter will describe the entire process of creating, training and evaluating the facial expression classification model that will be part of the final system.

The fifth chapter will close the document with the conclusions of the project, pointing the achieved objectives, the limitations of the developed project, and some viewpoints for future work.

2 Context and State of the Art

For a better understanding of the context, it will be presented in this chapter an explanation of the theoretical concepts around stress and facial expressions. In a second stage, will be shown an overview of some techniques used in computer vision, followed with a more detailed explanation on artificial neural networks, pre-trained networks, and datasets.

At the end of this chapter will be presented some existing solutions and proposals in stress detection, finalizing with a comparative analysis of the previously exposed solutions.

2.1 Facial Expressions

In one of his books, Paul Ekman indicates the human face as a multi-message system which provides three types of signals: “static (such as skin colour), slow (such as permanent wrinkles), and rapid (such as raising the eyebrows)” (Ekman & Friesen, 2003). In this study, rapid signals are the most valuable. These are the ones that express emotions and moods and are called facial expressions.

2.1.1 Universal Facial Expressions of Emotion

Paul Ekman wrote an article (Ekman, 1970) where he exposes some evidence of the universality of the facial expressions of emotion. That is, regardless of culture, country, race, or religion, certain emotions are manifested through the same expressions. In a more recent study (Ekman, 2016), Ekman conducted a questionnaire to the world’s leading emotion scientists and more than 75% of them agreed on the universality of 5 emotions (anger, fear, disgust, sadness and happiness). In this same study, scientists also agreed on the relationship between emotions and moods.

2.1.2 Macro and Micro-expressions

An expression can have many points of variation, such as the intensity and the time with which it is expressed. These are distinguished in macro-expressions and micro-expressions. (Shreve et al., 2009).

Macro-expressions are the most common expressions that untrained people can easily distinguish. These are facial expressions that are displayed in full and that typically last longer than one second (Ekman & Friesen, 2003). In (Revina & Emmanuel, 2018), are presented several systems for recognition of facial expressions, more specifically, macro-expressions.

Micro-expressions can be described as short-lived facial expressions. They can be as short as 1/5 to 1/25 of a second and occur mainly when someone is trying to hide a genuine emotion. Another characteristic of the micro-expressions is that they can occur only in one part of the face (Ekman & Friesen, 2003; Porter & ten Brinke, 2008).

However, due to micro-expressions being so fast, it is quite a time-consuming task. It requires specialized people to analyse videos frame by frame in order to be able to classify micro-expressions. Considering these disadvantages of micro-expressions, several studies have already been carried out to try to automate its detection and classification (Polikovsky et al., 2009, p.; Shreve et al., 2009, 2011, p.; Wu et al., 2011, p.; Xu et al., 2017).

2.1.3 Facial Action Coding System

As a way to facilitate the study of facial expression of emotion and micro-expressions, Paul Ekman and Wallace Friesen (Ekman & Friesen, 1976) created a system to taxonomize human facial movements. Facial Action Coding System (FACS) encodes all facial movements with a unique code, called Action Units (AUs).

This system has the advantage of being able to interpret facial expressions, or even more subtle small signs, without the need to classify into one of the universal emotions.

Thus, it is possible to capture all the information transmitted by the human face and later classify more correctly in a universal emotion or a mood.

Some work has already been done as a way to automate this translation of images for Action Units. For example, in the (Marian Stewart Bartlett et al., 1995) was applied holistic spatial analysis, feature measurement, and optic flow techniques, combined in a neuronal network that reached a generalization performance of 92%. In (M.S. Bartlett et al., 2004) resorting to an AdaBoost algorithm, were selected the best Gabor filters, which were later classified with a Support Vector Machine. This system presented an accuracy of 94.5% for the classification of 18 AU's. In the work (Hamm et al., 2011) was developed a system that uses Gabor filters and Active shape model to extract features that were later classified by AdaBoost. This system reached an accuracy of 95.9% for 15 AU's.

2.2 Stress

There is a great discussion around the definition of 'stress'. The first to present a definition was Hans Selye as being "the non-specific response of the body to any demand placed upon it" (Selye, 1950). These demands are also called stressors (Selye, 1975). A stressor can be any internal or external stimulus with adverse effects (Chrousos, 2009). Later a new definition occurred in a literature review, where they added that only when the stimulus is perceived as unpredictable and uncontrollable would we be in the presence of a stressor (Koolhaas et al., 2011).

2.2.1 General Adaptation Syndrome

Selye was also the one who proposed the "General Adaptation Syndrome" (Selye, 1946). Here he describes the adaptive nature of the stress response, and as illustrated in Figure 2, consisting of 3 stages (Selye, 1946, 1975):

1. **"Alarm Reaction"**— The definition given by Selye for this stage is to be a set of reactions for sudden stimuli for which the organism is not adapted to support. This phase is when the body triggers the fight-or-flight response, controlled by the Sympathetic Nervous System (McCorry, 2007). It promotes an increase in the production of adrenaline and norepinephrine. Such as increases in the heart rate, blood cell oxygenation and cortisol levels.

2. **"Stage of Resistance"** – This stage is when there is a higher resistance of the organism to the stressor. Because is when the body is continuously exposed to the stressor and consequently tries to develop resistances/ways of supporting it. If the exposure to the stressor ends during this stage, the body can recover without damage.

3. **"Stage of Exhaustion"** – In this last stage, there is a drop in the body's resistance to the stressor. Because of continuous exposure to the stressor, and despite having developed an adaptation to these stimuli, the body can no longer maintain such adaptation. Upon reaching this stage, there is the possibility of contracting serious health problems.

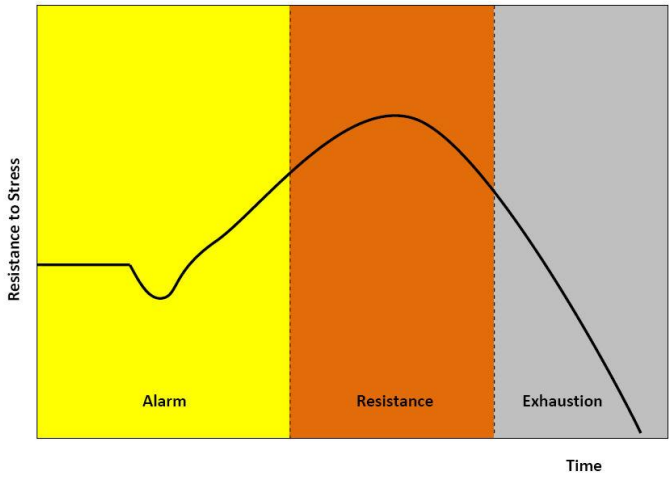

Figure 2 – A diagram of the 3 phases of the General Adaptation Syndrome. Source: (Myers, 2008)

Due to this duality of stress in which there may be an adaptation to the stressor or exhaustion, Selye proposed the terms 'distress' and 'eustress'. The former refers to poor adaptation to a stressor and the latter to a beneficial adaptation to a stressor.

2.2.2 Health Problems

The relationship between exposure to stress and the contraction of diseases has long been studied. Long-term exposure to stress, also called chronic stress, promotes the development of cardiovascular problems, obesity, cancer and its spread rate, immune disorders, and mental disorders such as depression and burnouts (Esler, 2017; Le et al., 2016; Lupien et al., 2018; Matosin et al., 2017; McEwen, 2017).

2.3 Facial Expressions of Stress

After showing the most relevant points of facial expressions and stress, it is then necessary to understand the relationship between these two and explain how stress can be detected from facial expressions.

In the study (Lerner et al., 2007), was carried out an experiment where ninety-two people were exposed to various stressors. During the experiment, were collected images of the subjects face, cardiac activity and saliva samples. The saliva samples were used to determine the levels of cortisol (stress hormone).

After analysing the results, were verified some relevant points.

Negative emotions (fear, anger and disgust) are influenced by stress. That is, the subjects being exposed to stressors expresses one of those three negative facial expressions.

It was also concluded that facial expressions vary based on the type of reaction to the stressor. If the subject considered the stressor as predictable and controllable, it would tend to express anger and disgust. Otherwise, if the stressor were unpredictable and the subject felt that had lost the control of the situation, then would express fear instead. The feeling of stress was validated by the increase in cortisol levels and cardiac activity. In unpredictable and uncontrollable situations, was when a greater increase in biological responses was detected.

It was also confirmed that the timing or intensity of facial expressions is important for predicting physiological responses, time for the anger and the intensity for fear and disgust.

In another study (Dinges et al., 2005), the relationship between the different levels of stress in facial expressions was also verified. However, they took an approach to micro-expressions, concluding that mouth and eyebrow regions had the best potential to determine stress. They used FACS to capture the various singular movements of the face.

2.4 Methodology

In this chapter, will be given a brief explanation of various techniques and methodologies related to image/video handling, as well as classification algorithms.

2.4.1 Images Pre-processing

Collected images do not always have the best quality or represent the face in the best way possible. Some of the challenges are head pose towards the camera, poor lighting, or even some obstacle between the face and the camera, such as scarfs or the person's hand.

However, with the evolution of technology, new techniques have been proposed to tackle some of these challenges. From changes in the images colour, contrast enhancement, noise removal, application of filters, normalization, reshape and resizing.

Another critical step is the identification of the region of interest, which in the context of this document will be face detection. The different face detection techniques can be distinguished into four categories:

- **Knowledge-based methods:** Methods that use human pre-defined rules to determine if it is a face;
- **Feature invariant approaches:** Look for facial structures that can stand out even with changes in head pose and luminosity;
- **Template Matching methods:** Searches for a face in the image that are similar to a pre-stored face template;
- **Appearance-based methods:** Use a statistical model for the detection of faces. This model is built from the analysis of face images training set.

The category of techniques most used today is the appearance-based methods (Zhang & Zhang, 2010). Techniques like Eigenfaces, Hidden Markov Models, Naïve Bayes, Support Vector Machines (SVM) and Artificial Neural Networks (ANN) are some examples. More can be found in (Kumar et al., 2019; Ming-Hsuan Yang et al., 2002; Zhang & Zhang, 2010).

Despite the existence of numerous algorithms for facial detection in the current days, one of the biggest drivers was Viola & Jones who with their work (Viola & Jones, 2001) managed to make facial detection practically feasible for real-world applications. His proposal consists of Haar-like feature selection, computation of the integral image, classifier learning with AdaBoost, and the attentional cascade structure.

2.4.2 Feature Extraction

Once the images have been treated and possibly corrected, it is appropriate to extract features. The features represent characteristics of the identified face, from the opening of the eyes, gauze direction, position of the eyebrows, movements of the lips, or even the heart rate.

There are different ways to extract this information, depending on the existing images. For static images, there are geometric feature-based methods and appearance-based methods.

In geometric feature-based methods, are used facial landmarks (red dots in Figure 3). From these dots, the algorithms can identify the position, direction or shape of the different components of the human face. However, one of the defects of the geometric feature-based methods is the need for accurate feature point detection techniques and the difficulty of implementing such techniques in the complex real-world background (Zhao & Zhang, 2016). Active Shape Model, Active Appearance Model (AAM), and Scale-invariant Feature Transform (SIFT) are some of the geometric feature-based methods examples. For a more detailed description, refer to (Zhao & Zhang, 2016).

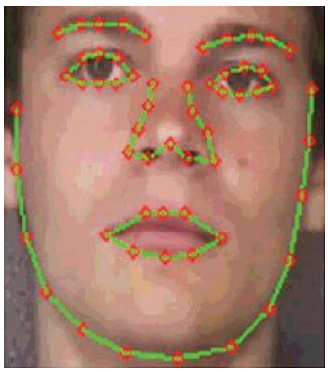

Figure 3 – The shape model, defined with 58 facial landmarks. Source : (Chang et al., 2006)

Appearance-based methods, unlike the previous one, do not use the contour of facial features, but directly the image. Based on the photometric appearance as the colour distribution or filter responses of the facial features, the algorithms interpret its morphology (Hansen & Ji, 2010).

Some examples of such algorithms are Local Binary Pattern and Gabor wavelet representation. More information in (Zhao & Zhang, 2016).

However, when there is a time component, are needed techniques cable of observing changes in the face over time. These changes can be movements or changes in the appearance of the face, such as changes in colour.

For movements interpretation, one of the options is the use of Optical Flow (Agarwal et al., 2016). This method evaluates the movement of each pixel in the image and assigns it a velocity vector. From the analysis of this vector, it is then possible to perceive the various movements of the facial components.

Another option is Feature Point Tracking (Pantic & Patras, 2006). This method tracks specific landmarks on the face. It obtains the various locations of these points over time, and it is then possible to determine which facial movement was performed.

There are also some variations in skin tone over time. These variations can be extracted with photoplethysmography techniques, that allows reading blood volume changes under the skin. It is possible to apply this technique with ordinary colour cameras that, from the variations of the skin colour of the human face, measure cardiac activity. This specific variation that uses images from a camera is called Remote Photoplethysmography (rPPG). In the work of (Wang et al., 2015) can be obtained a more detailed explanation.

2.4.3 Classification

Once the features of the images/videos are extracted, it is then possible to approach the problem as a common machine learning problem. Classification is a type of machine learning in which, through the application of mathematical calculations and probabilistic distributions, it evaluates the input values, called features and predict the target class. This target class is a discrete value that does not imply order. The classification techniques require a set of train data where the class to which they belong is known (Krishnaiah et al., 2014).

Classification problems can be distinguished into two types, binary or multiclass. Binary classification is when the objective attribute takes on only two distinct values, for example, stress or non-stress. Multiclass problems are when there are more than two possible classification options, such as happiness, sadness, disgust, surprise, anger and fear (Krishnaiah et al., 2014).

Numerous algorithms have already been proposed for solving classification problems. SVM, K-Nearest Neighbour (K-NN), Naïve Bayes, Decision Tree, Random Forest and ANN are some examples. In the works (Hemmatian & Sohrabi, 2019; Krishnaiah et al., 2014; Zhao & Zhang, 2016) can be found more detailed explanations.

2.5 Artificial Neural Network

Artificial Neural Networks are a sub-area of the discipline of Machine Learning, which, like the human brain, consists of a network of neurons that communicate with each other. However, in the case of ANN, neurons (or nodes), are stratified in several layers, where each connection between neurons has a weight.

The layers can be of the type input, hidden and output, as shown in Figure 4.

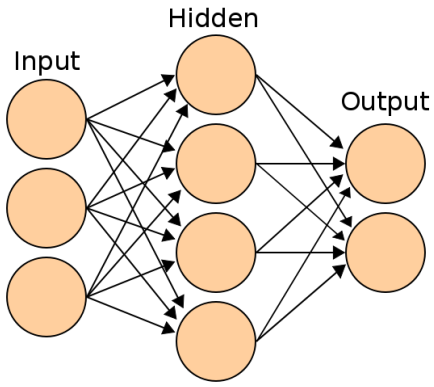

Figure 4 – Artificial Neural Network Schematic. Source: (Cburnett, 2006)

In the input layer is where will enter raw data, which can be anything from numerical values, to images and videos. Hidden layers are where functions will be applied, which will influence the propagation of information from layer to layer. These functions, together with the connection weights, aim to give more or less importance to each relationship between nodes. Finally, the output layer will always be the last one in the network. It is responsible for returning a response in the form of classification or forecast.

ANNs can assume two different architectures, feedforward and recurrent. In the former, the information only proceeds in the direction of input to output. In the latter, the connections may form feedback loops, assigning memory to the network.

It is also possible to create multi-layered artificial neural networks, known by the name of Deep Neural Networks (DNN). An example of DNN is the Convolution Neural Networks (CNN).

2.5.1 Convolution Neural Network

Convolution is a mathematical operation that uses a filter, or kernel, and applies him to the input data in order to obtain a transformed feature map (*Convolution*, 2018). This type of DNN is widely used for the analysis of images and videos, allowing to achieve state of the art performances (Karpathy et al., 2014).

In addition to the convolution layers that allow feature learning, CNNs can also consist of pooling layers that are used to reduce the size of feature maps (subsampling), while allowing

the network to be less affected by rotations and translations of the objects to classify. Normalization layers that apply techniques such as batch normalization or dropout, and aim to improve the stability and performance of the network (Ioffe & Szegedy, 2015). And fully connected layers in which the nodes are connected to all other nodes in the previous layer (*Convolutional Neural Network (CNN)*, 2018). In Figure 5 is presented a diagram of a CNN.

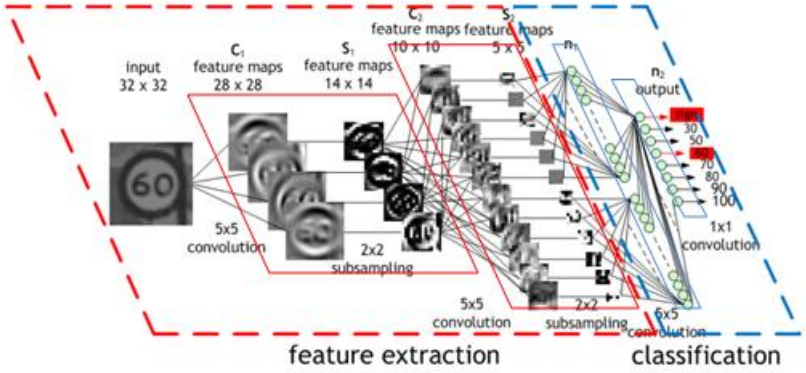

Figure 5 - An example of a CNN that receives an image of a traffic signal as input and classifies that image as "sign" and "60". Source: (*Convolutional Neural Network (CNN)*, 2018)

In the following subsections, will be presented some of the key concepts of Convolution Neural Networks.

2.5.2 Optimizer

In deep learning, the standard used optimizers are implementations of gradient descent algorithms. Gradient descent is an algorithm that seeks to minimize a loss function by updating the model parameters in the opposite direction from the gradient of that same loss function (Ruder, 2017). The main parameter of this algorithm is the learning rate, which will indicate the magnitude of the updates to be made to the model weights in the backpropagation process.

Currently, there are already many evolutions to gradient descent, such as the Stochastic gradient descent (SGD), Mini-batch Gradient Descent, RMSProp, or Adam.

Some of the optimizers indicated above implement techniques complementary to gradient descent, such as Momentum, Nesterov Accelerated Gradient, and adaptive learning rate.

2.5.3 Learning Rate

Learning rate is a hyperparameter from the optimization algorithm, that determines the magnitude of the changes made to the model's weights during the backpropagation process. The higher the learning rate, the more significant the changes are.

This hyperparameter commonly has positive values less than one and greater than zero, commonly multiple values of 10.

However, the value to be attributed to the learning rate depends on each case, with higher values generally being assigned at the beginning of training, and reducing his value with techniques such as early stopping or schedulers, when reaching plateaus.

By reducing the learning rate during training, most of the time, it is possible to get out of local minima and improve the model's performance. However, when it drops to very low values of learning rate, the changes made to the weights are so small that it will be difficult to observe improvements in performance. It may even begin to overfit the training data, reducing the model's generalization power. Therefore, the learning rate is one of the most important hyperparameters that is crucial to tune well.

2.5.4 Loss Function

The training of a neural network involves that, at the end of each epoch, its performance is evaluated according to a specific objective, which is intended to maximize or reduce.

Typically, with neural networks, the functions that evaluate the model's performance, seek to minimize the error, so they are commonly referred to as loss functions that calculate the model's loss (Brownlee, 2019).

As suggested in (Chollet & Allaire, 2018a), for classification problems the most suitable is the calculation of the cross-entropy. Cross-Entropy "is a quantity from the field of Information Theory that measures the distance (...) between the ground-truth distribution and your predictions.". That is, the farther from the correct value the forecast made by the model is, the greater the cross-entropy will be. François Chollet also recommends that for multiclass and single-label classification problems, the most suitable loss function is categorical_crossentropy.

2.5.5 Metrics

At the beginning of the problem definition, it is necessary to define what is meant by success, and as such, determine the metrics to measure that success.

Since the dataset (KDEF), which will be used to train the neural network, has the same number of samples for each of the emotions, we are in the presence of a balanced-classification problem. Therefore accuracy and area under the receiver operating characteristic curve (ROC AUC) are common metrics to measure these problems (Chollet & Allaire, 2018b).

2.5.6 Data Augmentation

Overfitting is one of the main difficulties in creating models. The models adapt too much to the training data and lose the ability to predict observations that were not present in the training

data. That is, since it has too few samples to learn from, the model will not be able to generalize to new information.

If it is not possible to obtain new data for the dataset, can be applied data augmentation techniques. These will generate more data from existing data, using random transformations which, depending on the type of existing data, will generate credible artificial data.

Standardly, the data augmentation is performed during the training process, continually feeding the model in training with different information, encouraging him to generalize better to the various aspects of the data.

In computer vision, data augmentation techniques can involve numerous transformations such as rotation, translation, magnification, flipping, cropping, Gaussian Noise, contrast change, luminosity, and so on.

However, it is necessary to evaluate the problem at hand and determine which transformations will bring plausible aspects to the model. For example, imagine the classification of handwritten numbers, techniques such as horizontal flipping and vertical flipping are not advisable, as these types of changes would alter the meaning of the image creating invalid data.

With the use of data augmentation techniques, the model during training will never see the same images twice. However, since these images are intercorrelated because they come from the same set of original images, it is necessary to be aware of some aspects of the real world that were not present in the original dataset. Despite data augmentation techniques, these aspects may have not yet been understood by the model (Chollet & Allaire, 2018c).

2.5.7 Dropout

Dropout is a regularization technique for neural networks developed in 2014 by Geoff Hinton and is students (Srivastava et al., 2014), that improve neural networks performance by reducing overfitting. This technique consists of "randomly dropping out (setting to zero) a number of output features of the layer during training" (Chollet & Allaire, 2018b). By randomly dropping out units and their connections during training, will prevent units from co-adapting, and with that, overfit to the training data (Srivastava et al., 2014).

This technique requires that the parameter "dropout rate" be specified, with a value between 0 and 1. During the training phase, this value will determine the fraction of the neurons that are dropped out. During the test time, the neurons are not dropped out, but instead, the layer's output values are scaled down by a factor equal to the dropout rate (Chollet & Allaire, 2018b).

Dropout in deep neural networks is usually done in the classifier layers before or/and between the dense layers. For the dropout rate, the recommended values by Geoff Hinton, in his paper, is about 0.8 to 0.5. If there is too much dropout (low values in the dropout rate), the network may not learn and contribute to underfitting, but if there is little dropout (high values in the dropout rate), it will not effectively counter overfitting (Srivastava et al., 2014).

It is important to note that some implementations of this algorithm, Keras is an example, have reversed the dropout rate. Where higher dropout rate values contribute to a greater number of neurons that are dropped out, and reduced dropout rate values to a smaller number of neurons. The values from the original paper to this new implementation can be calculated by the formula (1) below.

$$Keras\ dropout\ rate = 1 - Paper\ dropout\ rate \quad (1)$$

2.5.8 Transfer Learning

In the context of deep Learning, transfer learning is a technique that reuses characteristics learned in solving a general problem, called source domain, as a starting point for solving another problem, target domain. With this technique, it is possible to leverage learning to solve a problem in fewer iterations than those that would be necessary without the previous knowledge.

This technique is especially advantageous, for example, in image classification problems where the dataset does not contain enough information to train a full-scale model from scratch (Chollet, 2020).

As the project of the current document, the most common in computer vision is the use of pre-trained networks, which was trained on a large benchmark dataset. Some examples of such pre-trained networks are VGG or InceptionResNet, which will be covered later.

Once these pre-trained networks are trained in broad and generic datasets, then, "the spatial feature hierarchy learned by the pre-trained network can effectively act as a generic model of our visual world, and hence its features can prove useful for many different computer vision problems, even though these new problems might involve completely different classes from those of the original task." (Chollet & Allaire, 2018c).

These pre-trained networks are divided into two parts, convolutional base, and classifier.

The convolutional base, commonly composed by stacks of convolutional and pooling layers, aims to generate features from the image. This process is called Feature Extraction.

The classifier, often composed of fully connected layers, classifies the image based on the features extracted by the convolutional base.

However, the further we progress in the neural network, the more specific the features become for the source domain. In other words, the first layers of the convolution base will extract very generic features that can be reused to solve multiple target domains. In contrast, in deeper layers the extracted features will be less and less relevant for solving other problems, reaching the classifier that will be fully tuned for the source domain and will only be useful for solving very similar problems. Thus, the classifier is rarely reused in transfer learning.

The typical transfer learning workflow in computer vision is:

- Select a pre-trained network that solves a problem similar to the one intended to be solved;
- Replace the classifier with a new one to be trained in the new dataset;
- Freeze the convolutional base and train the neural network in the new dataset.

2.5.9 Fine Tuning

However, if it is not possible to use a pre-trained network trained in a similar domain to the one we want to solve, or even if it is necessary to obtain a better performance of the final model, we can proceed to a complementary technique called fine-tuning. Fine-tuning consists of enabling the training of some of the deeper layers of the convolutional base, which are more specific to the source domain, and in this way readjusting them to make them more relevant to the new dataset.

In fine-tuning, it is necessary to determine how much of the convolutional base to train. It is essential to bear in mind two factors, the similarity between the target and source domain, as well as the size of the dataset to classify. Figure 6 shows two matrices that can help to decide the proportion of the convolutional base to train based on the factors mentioned above.

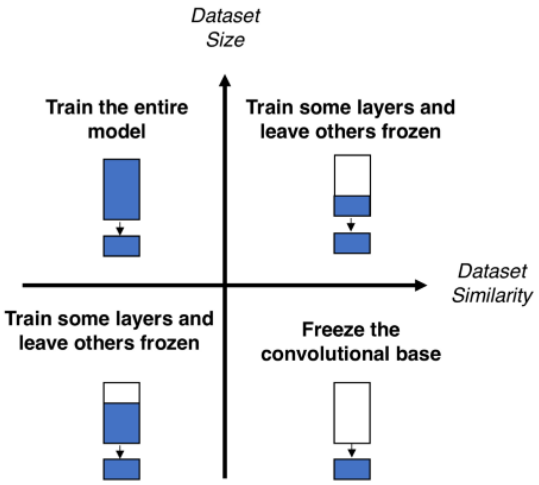

Figure 6 - Matrix representing the appropriate number of layers to be trained depending on the size and similarity between source dataset and target dataset. Source: (Marcelino, 2018)

Analyzing the matrix can be seen that there are three possible approaches.

1. If there is little data on which to train the new model and the datasets are similar, then training only the new classifier should be enough to achieve acceptable accuracy, not

being necessary to apply fine-tuning.

2. If we have a large set of data, but this is different from the one that trained the pre-trained network, then, the features extracted by the pre-trained network will not be of much use, since there is enough data to create a powerful feature extractor from scratch.
   Nevertheless, the pre-trained network architecture may still be relevant to the resolution of the problem and the previous weights as initialization for the new model. However, this approach will require a significant amount of processing power since the entire network will be trained.

3. In intermediate cases, only part of the base convolution should be trained to maintain the most general knowledge of the superficial layers and readjust the weights of the deeper layers that have more specific feature extraction patterns. The more the datasets are similar, then more layers can be left frozen, as they have knowledge that is still relevant to the problem, thus saving processing capacity.
   In case the datasets are different, it will be necessary to train a more significant number of layers, readjusting the more specific layers to the previous problem.

However, when applying fine-tuning, it is necessary to pay attention to three nuances. First, before applying fine-tuning to the convolutional base, it is imperative to train the randomly initialized classifier with the entire convolutional base frozen. "If the classifier wasn't already trained, then the error signal propagating through the network during training would be too large, and the representations previously learned by the layers being fine-tuned would be destroyed." (Chollet & Allaire, 2018c).

Second, the more layers are enabled to train, the greater the risk of overfitting the model to the dataset. To combat this tendency, can be used data-augmentation, dropout, or even a more extensive dataset. Preferably all the three.

Third, it is vital to use a very low learning rate, so the changes made to the weights during the backpropagation process are not too significant, which could damage what was learned by the layers of the convolutional base in the source dataset.

2.6 Pre-Trained Networks

For the project implementation, were used transfer learning techniques, and as such, it was necessary to use pre-trained neural networks. The three selected neural networks were all created based on the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) (Russakovsky et al., 2015), which is based on assessing the ability of algorithms for object detection and image classification at large scale. ImageNet is a dataset consisting of more than 14 million hand-labelled images for almost 22,000 different classes.

The selected neural networks were VGG16, VGG19 and InceptionResNetV2.

VGG16 and 19 were selected because they are very simple and straightforward architecturally and because they are quite often referred to as the main networks used for transfer learning.

InceptionResNetV2 was also selected because it is mentioned in (Hung et al., 2019), presenting good results in a facial expression classification task.

The three selected networks also have the advantage of being easily available on Keras.

2.6.1 VGG16 and VGG19

K. Simonyan and A. Zisserman proposed these two pre-trained networks in (Simonyan & Zisserman, 2015), which were submitted to the 2014 ILSVRC, where they achieved the first and the second places in the localization and classification tasks respectively.

The two networks are practically the same; they differ only in the number of layers of the convolution base. In Figure 7 it is possible to see graphically the layers used for each of the configurations.

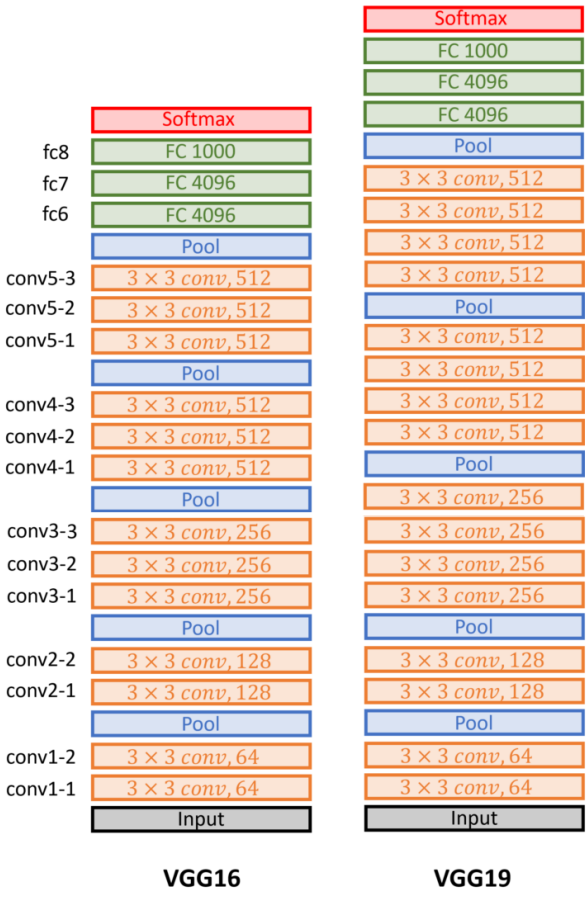

Figure 7 – VGG16 and VGG19 architecture. Source: (Matić et al., 2018)

The convolution base of these networks is made of convolutional and max-pooling layers. The convolutional layers use a 3 x 3 pixels filter, and padding and stride of 1 pixel. For the max-pooling layers were used a 2 x 2 pixels window and a stride of 2.

In all layers is used the ReLU activation function, except for the Dense layer output of the classifier that uses Softmax.

It should also be noted that these networks were trained to receive RGB images of 224 by 224 pixels.

2.6.2 Inception-ResNet V2

Inception-ResNet V2 is the result of an evolution of its predecessors ResNet and Inception, being a combination of the main characteristics of both. This was proposed by C. Szegedy and colleagues in (Szegedy et al., 2016). This model was also submitted in the 2015 ILSVRC yielding state-of-the-art performance.

At an architectural level, it is a more complex network than the VGG, wider, with filters of different sizes, and organized in different types of blocks.

From the Figure 8, it is possible to observe the condensed architecture of the Inception-ResNet V2.

By default, the Inception-ResNet V2 input waits for RGB images of 299 X 299 pixels and outputs a Dense layer with 1000 neurons and softmax as the activation function.

Internally, as shown in the image, it consists of a Stem, Inception-ResNet-A, B, and C blocks, Reduction-A and B blocks and ends with a layer of Global Average Pooling (GAP) with a dropout of 0.8 (keep 80% of neurons). This Average Pooling layer replaces the densely connected layers that many networks, like VGGs, usually use in the classifier of the networks.

Inception-ResNet modules are only made of convolution layers, while Reduction modules are made of both convolution layers and max-pooling layers in order to reduce the image size across the network.

The activation function used in the network is ReLU, with the exception of the output layer that uses Softmax and some layers in the Inception-ResNet modules that do not use activation function.

Keras' implementation for Inception-Resnet V2 varies from that described in the original paper. It uses twice the Inception-Resnet blocks and the Stem also has some changes in the number and layout of its layers. More can be found in the implementation at (Chollet, 2017).

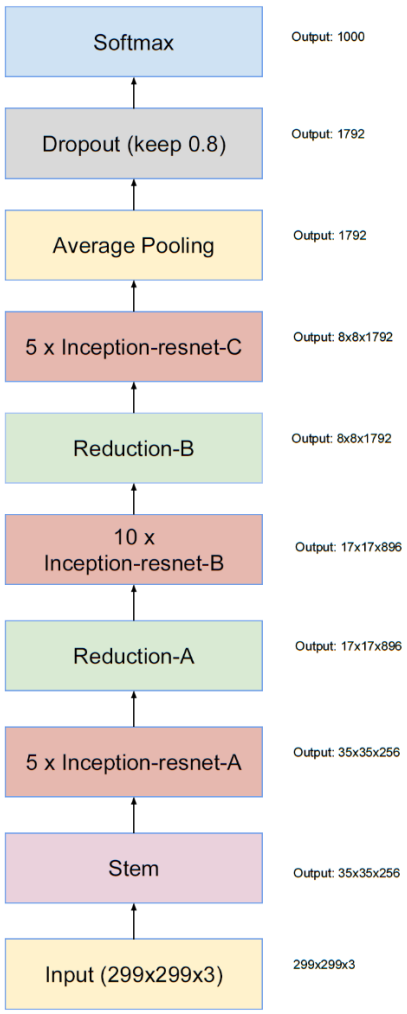

Figure 8 – Inception-ResNet V2 architecture. Source: (Szegedy et al., 2016)

2.7 Datasets

2.7.1 KDEF

Karolinska Directed Emotional Faces (KDEF) is a dataset created in 1998 by the Karolinska Institute, bringing together 4900 images of seven human facial expressions (Lundqvist et al., 1998).

Seventy amateur actors participated in the study, 35 females and 35 males, all between 20 and 30 years old. None of the subjects had glasses, beards, moustaches, earrings, or makeup.

The seven facial expressions of emotion represented during the photo sessions were, anger, disgust, fear, happiness, neutral, sadness and surprise. Before the photo session, all the actors trained the seven facial expressions that they later mimic during the session. Therefore, this dataset consists of posed, non-spontaneous facial expressions.

The images were captured in a controlled environment, where the luminosity, distance and orientation of the subjects were controlled. The photos were taken from 5 different angles: full left profile, half left profile, straight, half right profile, full right profile. After the actor represented all seven expressions, thus completing the first series, the process was repeated to capture the second series. For each emotion, were collected ten images per actor.

The images are 562 * 762 pixels in size.

2.7.1.1 KDEF adapted to the problem

Given the purpose of this project, which will use the user's computer webcam to obtain their facial expressions, it was decided to use only the half-left profile, straight, and half right profile images to train the neural network. The first reason was that the full left profile and full right profile contain little information about facial expressions, which can hinder learning for the neural network. Furthermore, by default, the webcams will be pointing the user's face from the front, capturing mainly the straight and half profiles.

In other words, of the 4900 images, only 2940 were used, 420 for each emotion.

In the pre-processing phase, the images were cut around the face, to remove part of the background, so that it did not contribute with noise to the learning phase. Finally, the images were all scaled to 299 by 299 pixels.

In Figure 9 can be seen three examples for each of the emotions along the columns and each of the three poses along the lines.

Figure 9 – Representation of facial expressions present in the KDEF dataset.
The first column represents the emotion anger, with the subject F25. The second column, disgust with subject M13. The third column, fear with subject F07. The fourth column, happiness with subject M11. The fifth column, neutral with the subject F30. Sixth column, sadness with the subject M32. The seventh column, surprise with subject F26.
The first line represents the straight pose, the second line represents the half left profile and the third the half right profile.

2.7.2 CK+

The CK+ dataset is the result of an extension to the CK created in 2000 by Cohn and Kanade that aimed to promote research into automatically detecting individual facial expressions (Lucey et al., 2010). They collected sequences of images from a neutral facial expression to the apex of one of seven emotions (anger, contempt, disgust, fear, happiness, sadness, and surprise). The image collection process involved 210 adults, between 18 and 50 years old, 69% female, 81% Euro-American, 13% Afro-American, and 6% of other groups.

The collected images were later classified with one of the seven emotions. The final image of every sequence that represented the emotions apex was encoded in AUs based on the FACS.

However, after a selection process, only 327 sequences from the 593 made it to the dataset. The number of sequences present for each emotion can be found in Table 1.

Table 1 – Number of sequences per emotion on CK+ dataset.

EMOTION	Nº of sequences
Anger	45
Contempt	18
Disgust	59
Fear	25
Happiness	69
Sadness	28
Surprise	83

The collected images had a size of 640x490 or 640x480 pixels with 8-bit grey-scale or 24-bit colour values.

Despite the above, the author of the document was unable to obtain the full version of the dataset, as the website indicated in the original paper (Lucey et al., 2010) to request the dataset is no longer available. Therefore, an unofficial version (Shawon, 2018) was obtained from the Kaggle platform where it contains only three images per sequence, in a total of 981 images, in black and white with a size of 48 by 48 pixels.

However, CK+ presents a significant difference in comparison to the KDEF dataset. The CK+ does not contain images for the neutral emotion and contains a different facial expression, contempt. In order to use the dataset, all contempt facial expression images were removed, reducing the dataset to 927 images. The dataset continued without images of the neutral facial expression.

Finally, all images were cropped around the face and resized to 299 x 299 pixels.

In Figure 10 is possible to see an example for the six facial expressions used.

Figure 10 – Representation of facial expressions used from the CK+ dataset.
From left to right, subject s55 with an angry facial expression, s74 with disgust, s132 with fear, s125 with happiness, s113 with sadness, and s130 with surprise. (©Jeffrey Cohn)

2.7.3 Net Images

Since the neural networks were trained with a dataset in a controlled environment and in which the population is mostly homogeneous, the author of the document decided to create a small set of images obtained from the internet for the seven emotions learned by the CNN.

The purpose of this small dataset is to get an idea of the extent to which the models generalized its learning to the emotions and did not adjust itself too much (overfitted) to the environment (setup) of the KDEF dataset.

Were collected twenty images for each emotion, for a total of 140 images. These come from searches on two search engines (Google and DuckDuckGo) and free stock images sites (unsplash.com; pexels.com; shutterstock.com; freepik.com). Searches were done using the expression "human <emotion> face", where <emotion> was replaced by the name of the respective emotion. For example, "human happy face".

Then, were selected images where the face was visible, and the emotion was unmistakably present. We tried to obtain very heterogeneous images, from people of different ages, different races, with and without a beard (same for glasses). Were avoided, images with watermarks, with visible image edition and in which the facial expression could be interpreted as a mix of emotions, as mentioned in the book Unmasking the face by Ekman and Friesen (Ekman & Friesen, 2003).

Once the images were collected, they were then cut around the face and resized to 299 x 299 pixels.

In Figure 11 is displayed one example for each emotion present in the Net Image dataset.

Figure 11 – Representation of facial expressions present in the Net Images dataset. From left to right, anger, disgust, fear, happiness, neutral, sadness, surprise.

2.8 Training, Validation and Test Sets

Before evaluating any model, it is necessary to ensure that the original dataset has been correctly divided into three sets: training, validation, and test. The training set will be used to train the model, after that the model will be evaluated in the validation set at the end of each epoch. Finally, once the model training is completed, it is tested in the test set to see if it has generalized enough or overfitted to the training data, validation data, or both.

Information leaks may occur during the training process, after observing the model's performance on the validation data, and by changing the model's hyperparameters, some information about the validation data leaks into the model.

Given that these changes based on the validation set can be considered as a form of learning, there is a risk that at the end of many of these iterations, the model starts to overfit to the validation data.

Therefore, as a way of assessing the model's performance more reliably on information never seen before, is created a test set that will only be used to evaluate the model's performance once fully tuned based on the validation data.

There are two main ways to divide data into these three sets: hold-out validation and K-fold validation.

2.8.1 Hold-Out Validation

Of the two, this method is the simplest and consists only of randomly separating the complete dataset into the three sets, training, validation, and testing. Commonly, the validation set and the test set are the smallest sets, with training being the largest, preferably with more than half of the data. The proportions will have to be decided based on the existing data type, distribution, and size of the dataset. It is necessary to reserve sufficient data for testing and validation that are statistically representative of the entire data. Moreover, at the same time, it is intended to have the most amount of data possible for training so that the model generalizes better.

In situations where the amount of data is reduced, these two factors can be conflicting, and as such, K-fold validation will be of a better use for solving the problem. In the other hand, in computational terms, the hold-out method is less expensive than K-fold validation.

2.8.2 K-Fold Validation

K-Fold Validation is the method most commonly used in machine learning, as this gives more reliable measurements, even with little data.

First, a test set is randomly separated and, again, its size will have to be weighted based on the dataset at hand. This will later be used to evaluate the model at the end of the K-Fold Validation process, precisely for the reasons explained above in "information leaks".

After the test set is separated, the remaining data will be used both as training data and validation data. That is, these data will be randomly divided into K folds, and K-1 folds will be used as training data and one fold as validation data. At the end of the training, the fold used for validation will be used for training in the next iteration, and a new fold will be the validation fold. In this way, all folds will be used as validation data precisely one time. This process will repeat for K times and in the end, calculated the mean of the validation score of all iterations.

The functioning of K-Fold validation can be easily understood from the figure.

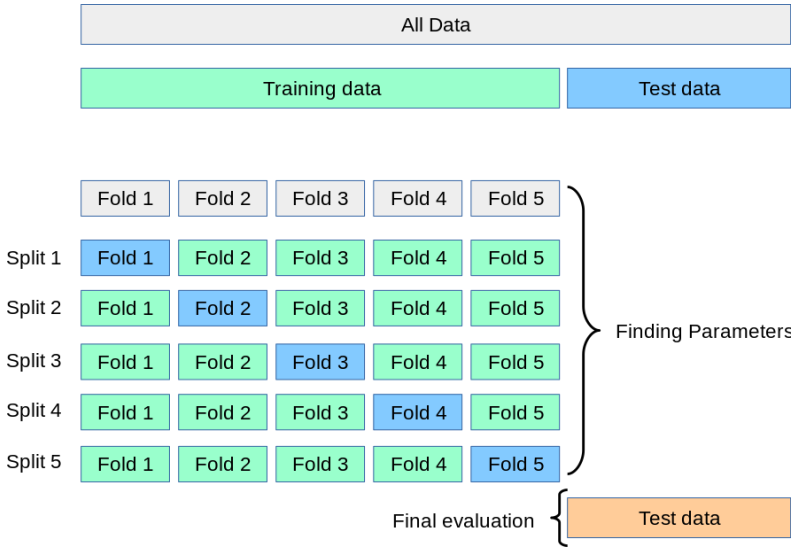

Figure 12 – K-Fold Validation overview. Source: (scikit-learn team, 2020)

In this method, there is a crucial value to define, the value of K. K will indicate how many portions of equal size will the dataset be divided (after reserving the test set). This value will have to be decided, in the same way as the size of the test set is chosen. Evaluate the existing data type, distribution, and size of the dataset.

There are some evaluation methods in which the K value is fixed or predetermined. As in the Leave-one-out Cross-Validation in which the validation set will only contain one example, and therefore the dataset will have to be divided into as many folds as there are instances.

However, when this method is applied to deep learning, it is hugely computationally and time expensive. For each split that can be seen in the image, it is necessary to train the model for n epochs which in some cases can take hours or even days. In other words, it would be necessary

to multiply these hours by the number of K folds, and every time changes are made to the hyperparameters it would be required to re-train the network for n epochs * k folds.

2.9 Tensorflow and Keras

Currently, the most used platforms for building neural networks are TensorFlow and Keras (Piatetsky, 2019). TensorFlow is a machine learning engine, that was created by a team from Google, called Google Brain. The first version of TensorFlow was launched in 2015 and at the date of this document is already in version 2.3.0.

Keras is a high-level API for TensorFlow that aims to facilitate the resolution of machine learning problems. Keras was initially created by François Chollet to offer a library capable of providing users with a development platform independent of the machine learning engine working as a backend (*Good News, Tensorflow Chooses Keras! · Issue #5050 · Keras-Team/Keras*, 2017). In addition to TensorFlow, were supported Microsoft Cognitive Toolkit, R, and Theano.

However, the version 2.2.5 of Keras was the last to support multi-backend, and since that version, Keras is only embedded in TensorFlow library (*keras-team/keras*, 2015/2020).

2.9.1 Google Colab

Google Colaboratory or "Colab" for short, is an online platform that allows the execution of python code directly in the browser, free of charge, with access to GPU and TPU processing without the need for configurations. Not only executes python code but it currently supports the latest version of TensorFlow.

In addition to the free version, there is also a paid version that allows access to more powerful GPU and TPU, the possibility of longer runs, as well as access to machines with twice the CPUs, memory, and disk space.

For the realization of this project, was used the free version that allowed access to Nvidia K80s GPUs, 12 GB of RAM, 68 GB of disk space and runs of up to 12 hours.

2.10 Existing Solutions

Automatic stress detection has been studied for many years. From some intrusive approaches, such as saliva or blood tests, heart activity, body temperature, Electroencephalography, galvanic skin response. To less intrusive approaches, with the collection of images.

In this section, will be analysed only non-intrusive solutions, using videos of people performing stressful tasks. From these videos, several features are then extracted and then used to create classifying models.

In the end, will be made a conclusion, summarising a comparison between the solutions.

2.10.1 Detecting Emotional Stress From Facial Expressions For Driving Safety

This study (Gao et al., 2014) was carried out in 2014 by researchers from the École Polytechnique Fédérale in Lausanne, Switzerland, whose main objective is the detection of stress while driving from images collected by a camera mounted inside the dashboard.

It should be noted that this study, assumes it is in the presence of stress if anger and disgust are detected above a specific time limit.

2.10.1.1 Data

The researchers themselves collected the images used. Using a near-infrared (NIR) camera, which captured at 25 frames per second (fps) and with a resolution of 1280 x 1024 pixels. This resulted in two sets of images, “Set1” and “Set2”.

The “Set1” was recorded inside an office, where the camera was positioned on a table and oriented horizontally with the subject’s face. None of the subjects was a professional actor, and a total of 21 were filmed.

In “Set2”, the images were collected inside a car, with the camera mounted inside the dashboard, behind the steering wheel. The camera was with a slightly up-tilted view-angle towards the driver’s face. A schematic of the camera setup can be seen in Figure 13. In this set, only 12 of the previous subjects were filmed.

In both sets, the subjects were asked to express the expression of stress for 1 minute.

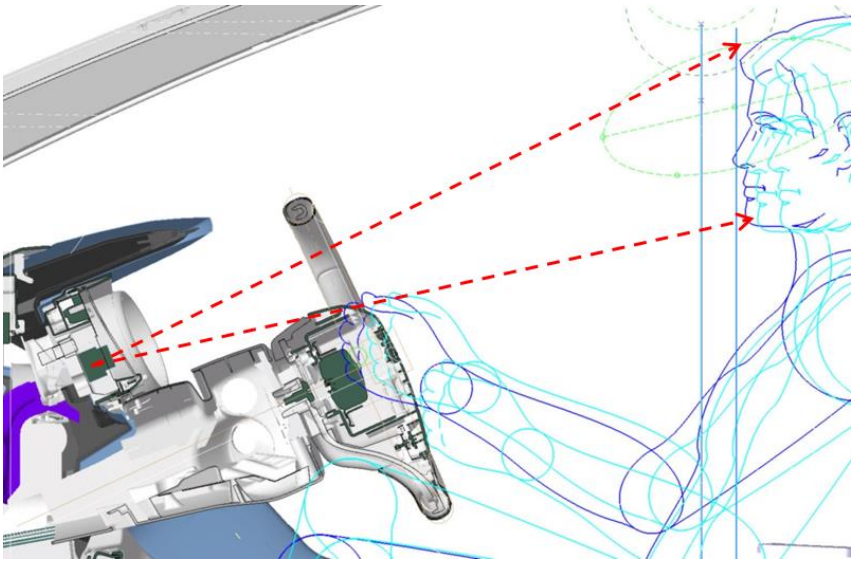

Figure 13 - The camera setup inside the car. On the left, inside the dashboard, the green block represents the NIR-camera, and the red arrows the viewing angle. Source: (Gao et al., 2014)

2.10.1.2 Implementation

The process consists of collecting the driver's images, perform face detection and face tracking, followed by the feature extraction, emotional detection and stress classification.

In the Face Detection step, it was implemented with Viola & Jones algorithm (Viola & Jones, 2004), and the face tracking with Supervised Descent Method (SDM). The SDM was configured to track 49 facial landmarks.

Two approaches were proposed for the feature extraction:

- The first approach treats the image in a holistic way. From the coordinates of the eyes, the images were centred and normalized. Finally, is applied an algorithm to extract local descriptors using block-based discrete cosine transform (Ekenel & Stiefelhagen, 2005);
- The second approach, extracted the local descriptors around the tracked facial landmarks, using scale-invariant feature transform (Chu et al., 2013). The images were centred and normalized with a 3D Cylindrical Head Model (Xiao et al., 2003).

For the recognition of emotions, they implemented a classifier with SVM, which was tuned with 5-fold cross-validation. They used two public data sets of facial expressions, the FACES (Ebner et al., 2010) and the Radboud (Langner et al., 2010). An extra classifier was created for each subject. These additional classifiers were trained like the others, but with the addition of images of the subjects' own facial expressions.

Finally, in the step of determining the presence of stress, was used the classifier described in the previous paragraph. If it detected the presence of disgust and anger emotions above a certain percentage of the frames analysed, then it is classified that the driver is under stress.

2.10.1.3 Results

Of the approaches performed, the one that obtained the best results was the second; however, with the use of the emotion classifier adapted to the subject with his own facial expressions.

With an accuracy of 90.5% (F-measure: 0.871, recall: 0.860, precision: 0.882) for "Set1" and 85% (F-measure: 0.815, recall: 0.735, precision: 0.914) for "Set2".

2.10.2 Automatic human stress detection based on webcam photoplethysmographic signals

Conducted by researchers at the University of Lorraine, France, in 2015, this study (Maaoui et al., 2015) aimed to develop a system capable of detecting stress from a computer's webcam using rPPG signals.

2.10.2.1 Data

To carry out the study, the researchers used images collected in a previous study (Bousefsaf et al., 2013), where 12 students from their laboratory participated. The sessions consisted of successions of relaxing videos and Stroop colour word tests to induce stress in the subject.

The images were collected with a conventional webcam at a resolution of 320 x 240 pixels and a frequency of 30 frames per second (fps).

2.10.2.2 Implementation

The first step was the automatic face detection, using a cascade of boosted classifier on each frame, based on (Viola & Jones, 2001).

Then a skin detection was performed to extract the skin pixels, where it is possible to obtain the PPG signals. They changed the colour space of the images from RGB to L * u * v * to help reduce fluctuations due to light variations. The u * component was then extracted, as it is the most important for rPPG signals. Finally, the PPG signals were converted into a sinusoidal wave, which represents the heart rate (HR).

Seven features were extracted from this wave: mean value of HR signal, standard deviation of the HR, first derivative of HR, root mean square of the successive differences, low-frequency band, high-frequency band and the ratio between low and high-frequency band.

In order to distinguish stress states from relax states, were used two classification algorithms: SVM with RBF kernel and Linear Discriminant Analysis (LDA).

2.10.2.3 Results

The SVM with RBF kernel was the algorithm that obtained the best results, with 94.40% accuracy, against 91.10% of the LDA

2.10.3 Stress and anxiety detection using facial cues from videos

A group of Greek researchers developed a system (Giannakakis et al., 2016) capable of detecting stress/anxiety emotional states through video-recorded facial cues. In addition to detecting stress, their work also includes detection of anxiety. According to them, the manifestation of these states is identical.

2.10.3.1 Data

For the collection of images, the researchers created an experimental procedure capable of inducing affective states (neutral, relaxed and stressed/anxious). This procedure consisted of four phases. Each phase aims to expose the subject to different types of stressors. The first phase is a social exposure, the second an emotional recall, the third stressful images and mental tasks, such as the Stroop test, and the fourth stressful videos.

From this procedure, where participated 23 adults, were obtained 276 videos at 50 fps and with a resolution of 526 x 696 pixels.

2.10.3.2 Implementation

They started by improving the contrast of the images with histogram equalization, followed by the application of the Active Appearance Models algorithm for the detection of the face and its landmarks.

The features were extracted using Active Appearance Models, Optical Flow, and rPPG. The AAM was used to obtain eye and head movements, the Optical Flow for mouth activities, and the rPPG for heart rate.

After the most relevant features were prepared and selected, were applied and tested the K-NN, Generalized Likelihood Ratio, SVM, Naïve Bayes and AdaBoost algorithms to create the stress/anxiety classifier.

The various classifiers created were adjusted and tested with 10-fold cross-validation.

2.10.3.3 Results

In the end, the best classification accuracy, of 91.68%, was obtained with AdaBoost for the phase of social exposure. However, the algorithm that achieved the best average accuracy over the four phases was K-NN with 87.72%, followed by AdaBoost with 85.95%.

2.10.4 Towards Independent Stress Detection: a Dependent Model using Facial Action Units

This study (Viegas et al., 2018) published in 2018, describes the proposal for a system capable of detecting signs of stress based on Facial Action Units of videos collected in a previous study (Maxion & Lau, 2018) that aimed to determine the differences between neutral and stressed typing.

2.10.4.1 Data

Five people participated in the data collection, which resulted in 5 hours of video at 30 fps and 1920 x 1080 pixels resolution.

The experimental protocol used to collect the images consisted of 30 minutes of rest, for the subject to enter a neutral state, then the subject provided a neutral typing sample. Once completed, the subject was submitted to a 15-minute stressor task. After that, he was asked to provide a stress typing sample. Image collection ended with a rest period to return to a neutral state and provide a final neutral typing sample.

2.10.4.2 Implementation

Since the objective of the study was the detection of stress only from facial Action Units, they used the OpenFace toolbox (Baltrušaitis et al., 2016) for the extraction of 17 different Action Units.

After analysing the data obtained from the videos, they defined three different classification problems. The first, a six-class problem, one for each phase of the experimental protocol. The second, a four-class problem, considering all phases of writing to be the same class. And a binary classification problem distinguishing between stress phases and non-stress phases.

Once the classification problems were defined, were used the Random Forest, LDA, Gaussian Naïve Bayes and Decision Tree algorithms to implement the classifiers. For the implementation

of the classifiers, were taken two approaches, one applying a leave-one-subject-out approach to obtain a person independent model and five other classifiers using 5-fold cross-validation to obtain a person dependent model for each subject.

2.10.4.3 Results

Regardless of the approach, the Random Forest was the best algorithm. For the person independent classification, were obtained 41%, 49% and 75% of accuracy for the problems of six-classes, four-classes and two-classes respectively. For the person dependent classifiers were obtained 83%, 83% and 93%.

2.10.5 Comparative Analysis

Of the four cases analysed, all of them needed to create an experimental procedure where subjects are stress-induced, and are collected videos. For, as mentioned by Viegas et al., doesn't seem to exist any dataset publicly available with videos of people during stress states. The videos can be summarised qualitatively and quantitatively through Table 2.

Table 2 – Comparative table of videos used in each solution

	FPS	**Resolution**	**Number of Subjects**	**Number of videos**	**Minutes of vídeo**
(Gao et al., 2014)	25	1280 x 1024	21	33	33
(Maaoui et al., 2015)	30	320 x 240	12	n/a	n/a
(Giannakakis et al., 2016)	50	526 x 696	23	276	333
(Viegas et al., 2018)	30	1920 x 1080	5	5	300

Except for the last, all used Viola & Jones algorithm (Viola & Jones, 2004, 2001) for face detection. Other pre-processing techniques used were 3D Cylindric Head Model for pose normalization, Histogram Equalization for contrast enhancement and colour space change from RGB to L * u * v * to reduce fluctuations due to light variations.

About the features used, each followed a different approach. In Gao et al. (2.10.1) collected local descriptors with SIFT; In Maaoui et al. (2.10.2) used only the heart activity, extracted with rPPG; In Viegas et al. (2.10.4) the facial Action Units was the only feature used, extracted with the OpenFace toolbox. The Giannakakis et al. (2.10.3) was the study with the biggest number of features, with eye and head movements extracted with AAM, mouth activity with Optical Flow and heart rate with rPPG.

For the detection of stress, all opted to implement a classifier. The different algorithm used was SVM, LDA, K-NN, Generalized Likelihood Ratio, Naïve Bayes, AdaBoost, Random Forest and Decision Trees.

Each study had different results with different algorithms; on Table 3 it is presented the best accuracy obtained with each solution.

Table 3 – Comparative table for the accuracy obtained in each of the studies.[1]

	Best Accuracy	**Used Algorithm**
Gao et al. (2.10.1)	90.50%	SVM
Maaoui et al. (2.10.2)	94.40%	SVM
Giannakakis et al. (2.10.3)	91.68% / 87.72%	AdaBoost / K-NN
Viegas et al. (2.10.4)	93.00%	Random Forest

[1] In the line Giannakakis et al. the first accuracy refers to the best value for the best of the phases and the second is the average accuracy of all phases.

3 Solution Design

The system to be developed will have only one use case. Detect and notify the user that he shows signs of stress.

The program will run in the background, monitoring the user's facial expressions, and when the program determines that the user is showing signs of stress, the program will notify the user of that fact.

For the construction of such a system, it was proposed to create four modules, as can be seen in Figure 14.

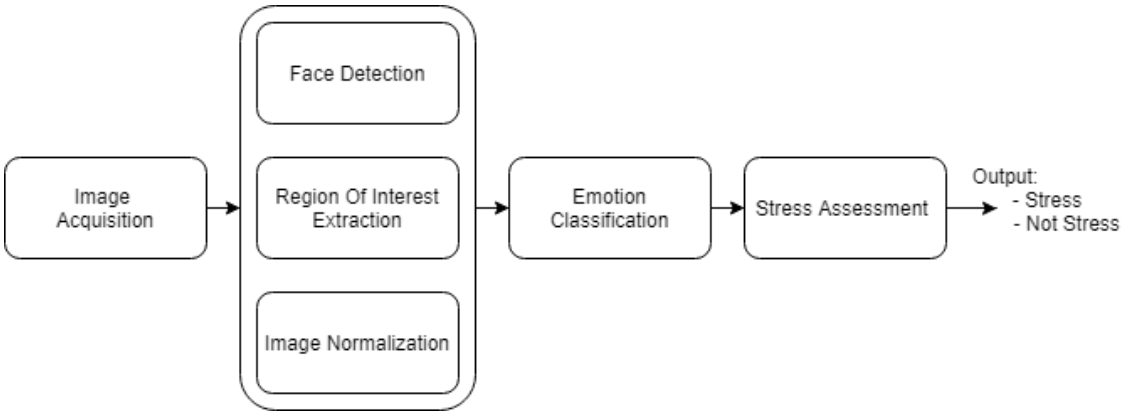

Figure 14 – Overview of the modules composing the stress detection system.

The operation of the envisioned modules can be described by the following sequence diagram, shown in Figure 15.

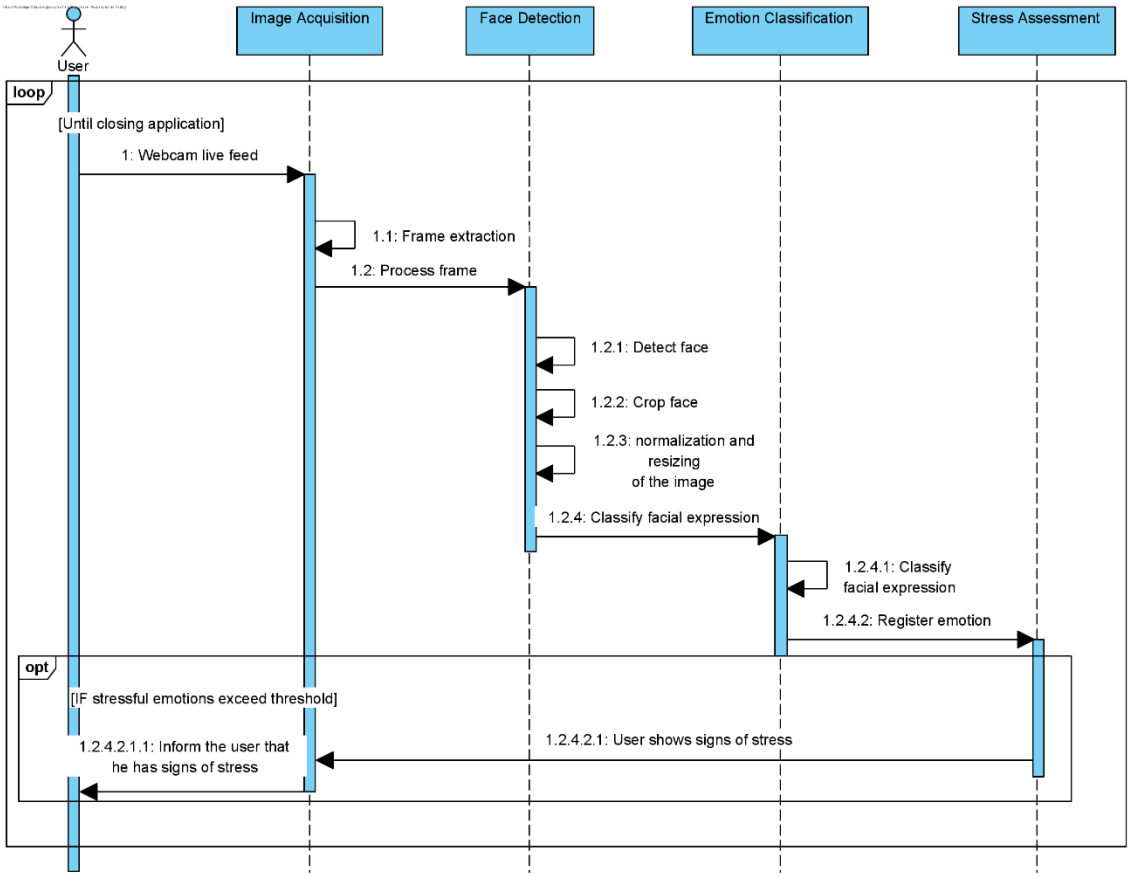

Figure 15 – Sequence diagram of system operation.

3.1 Image Acquisition Module

The first module is responsible for the image acquisition through the webcam of the computer. This module collects real-time footages of the user's face and periodically extracts a frame to send to the next module. It was decided only to collect frames periodically, first so as not to overload the processor and second because with an adequate frequency it is still possible to capture all the facial expressions represented by the user, even if they are very brief. The interval between each extraction will be given by a parameter, that will also determine the frequency of the classifications.

For the implementation of this module, was used the open-source computer vision library, OpenCV.

3.2 Face Detection Module

Once the frame is collected, it is processed by the OpenCV class, CascadeClassifier, which, using a Haar-like feature selection technique, will detect the face and return the coordinates of the face in the frame.

From these coordinates, the face is cut and resized to 299 by 299 pixels. Then, the image is normalized, by dividing the value of all pixels by 255 so that all have values between 0 and 1. Similar to the images used to create the classification model.

3.3 Emotion Classification Module

The normalized face image is then provided to the classification model that returns a list of seven probability scores, one for each emotion. These probability scores indicate the likelihood that the image represents the respective facial expression. These values are due to the activation function Softmax of the last neural net layer that translates the output into probabilities, and the sum of the output will always be 1.

At the end of this module, the emotion with the highest probability is then selected and given as input to the next module, stress assessment.

3.4 Stress Assessment Module

Whenever the model makes a classification, it will be recorded by the stress assessment module, that in turn will only distinguish between non-stressful or stressful emotions—being the anger, disgust and fear the stressful emotions, as documented in section 2.3.

This record will only take effect within a time window, being the size of that window parameterizable. For example, if we define this window as only 30 seconds, the assessment for the presence of signs of stress will be made only with the classifications from the last 30 seconds. The number of classifications within a 30 seconds window, will be dependent on the parameter indicated in the image acquisition module, that will determine the time interval between each frame extraction, and consequently each classification.

The distinction between a non-stressful or a stressful situation will be made based on a threshold. This threshold is the last module parameter that will indicate the percentage of stressful emotion needed to determine that the user shows signs of stress. For example, if the time window is 30 seconds and the threshold is 75%, it means that if in the last 30 seconds 75% or more of the classifications are for stressful emotions, then will be determined that the user shows signs of stress.

Once the module determines that the user shows signs of stress, it will be displayed a notification alerting for that fact.

In the Figure 16 can be seen an example of the notification.

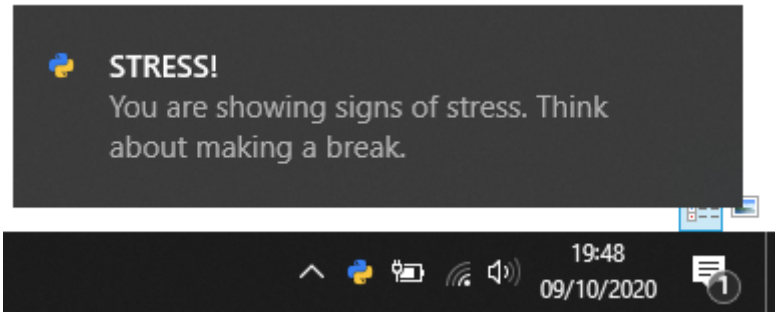

Figure 16 – Notification that will be displayed to the user in case of signs of stress.

4 Experimentation

For this project, was follow the CRISP-DM methodology, it provides guidelines for data mining and machine learning process in general, breaking it down into six phases, as shown in Figure 1.

It started with a Bussiness Understanding process where was comprehended what stress was and how it relates to facial expressions. After making this association, were collected datasets classified with the seven universal facial expressions, and made the necessary changes. Once the data was prepared, were constructed and fine-tuned models for classifying facial expressions. These models were then evaluated in the various datasets collected, and the best was then integrated into the final system.

4.1 Business and Data Understanding

As presented in the chapter "Facial Expressions of Stress" (2.3), there is a direct correlation between some facial expressions and stressful situations. As presented in the study (Lerner et al., 2007), in situations of stress, subjects presented three facial expressions, here called stressful emotions or negative emotions — fear, anger, and disgust.

Given this relationship, there were two possibilities. Address the problem with data classified as stress or non-stress (preferable approach), or with data classified in the seven facial expressions. Unfortunately, datasets classified directly as stress or non-stress are not easily available, and the dataset from (Dinges et al., 2005) was requested but the request was not answered. Therefore, was followed the second approach and was requested the KDEF dataset.

KDEF is a dataset classified into the seven universal emotions, described in more detail in section 2.7.1. Although complete and well uniformed, KDEF is a very homogeneous dataset, where the subjects are all of the same age group, same race, without any facial modification such as glasses or beard, and the images were all captured in a controlled environment.

As such, in order to counteract this homogeneity, it was decided to obtain the CK + and the Net Images datasets, in order to evaluate the created models with more heterogeneous data.

With the use of these two datasets, it is expected to obtain a more realistic assessment of the models and hopefully closer to the real world.

4.2 Data Preparation

Before using any of the three datasets, they went through pre-processing. This pre-processing included cropping the images around the face and resizing the result to 299 by 299 pixels.

As for the evaluation method to be used, was chosen Hold-Out instead of K-Fold due to the time and processing capacity required for the latter. Since were going to be tested three pre-trained neural networks and for each of them it would be necessary to fine-tune hyperparameters and change the number of layers to be trained, given the equipment and time available, it was decided to use Hold-Out.

NetImages and CK + remained in a single set. However, the KDEF that was used to train the models had to be partitioned into three sets, since was used the hold-out method. It was divided into 80% for training data, 10% for validation data and the remaining 10% for test data.

For the partitioning of the KDEF dataset, was taken into account what was presented in the work (Viegas et al., 2018), where neural networks showed an ability to adapt to people's faces, or even to the way they express their emotions. Therefore, for the separation of this dataset in training, validation and test data, subjects were always taken into account. That is, the divisions were made in such a way that images of a specific person only existed in one of the sets.

Since the subjects are catalogued with codes, they were randomly selected for one of the subsets. In this way, when the model is tested with data never seen before, it will be evaluated, not only with different images but with different persons.

Another aspect taken into account during the data preparation process, was the size of the KDEF dataset. There is not a large amount of data, so, during the training process was also applied data augmentation to the training images.

This data augmentation consisted of:

- Rotations up to 20 degrees;
- 10% and 15% translations for width and height, respectively;
- Brightness changes between 0.2 and 1 (where 0 would mean a wholly darkened image, 1 the original image's brightness, and values greater than 1 extra brightness until the image turns white.);
- Zoom-out up to 10% and zoom-in up to 20%;

- Horizontal flips.

In the data augmentation process, no modifications were made to the images in the validation or test set, as suggested by F. Chollet in his book (Chollet & Allaire, 2018c).

4.3 Modelling

The first step taken in creating the models was the definition of classifiers architecture. Were tried two different approaches, one classifier based on a global average pooling layer and a second with a convolution layer. The two architectures can be seen in Figure 17.

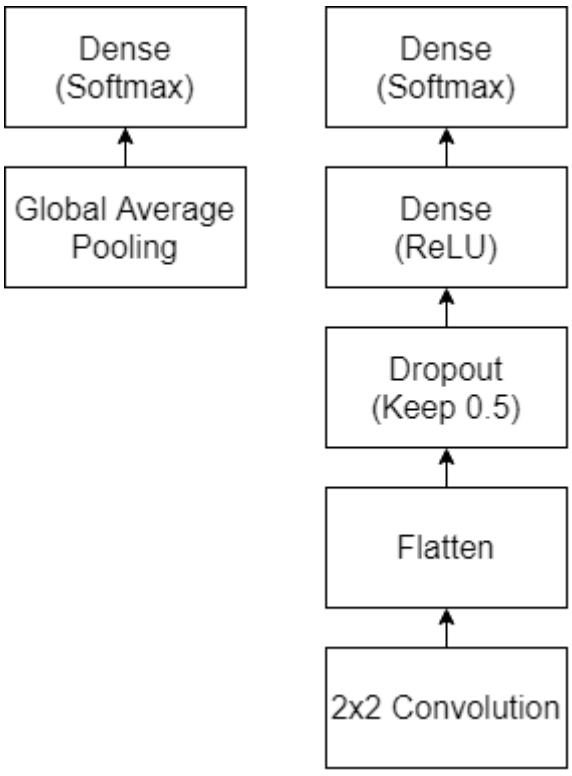

Figure 17 – Classifier architecture of the two tried approaches.

For the Global Average Pooling approach, was used only a GAP layer together with a fully connected layer with seven neurons and softmax as the activation function. Since this layer only returns the average of all pixels for each of the filters, there are no hyperparameters to define.

The second approach is made up of the convolution layer, followed by a flatten layer, that reshapes all the filters in a single array of one dimension, a 50% dropout to reduce overfitting along with two fully connected layers. The activation function ReLU composes the first fully connected layer and the last, which serves as the output layer, uses a Softmax function with seven neurons. For this approach, there are a set of hyperparameters that need to be selected. Therefore, were trained a set of different classifiers in order to select the most suitable values.

Essentially, we tried to determine the best value for the number of filters in the convolution layer and the number of neurons for the penultimate densely connected layer.

Giving four different configurations:

- **Configuration A:** 64 filters with 256 neurons;
- **Configuration B:** 64 filters with 512 neurons;
- **Configuration C:** 128 filters with 256 neurons;
- **Configuration D:** 128 filters with 512 neurons;

In order to understand the best configuration for each of the pre-trained networks, were tested the four configurations with each one of them.

For the training of these classifiers, it was necessary to select an optimizer. Initially, was tried the optimizer Adam, however, sporadically, he lost what he had learned so far and dropped to high loss values. Therefore, it was decided to use the Mini-batch Gradient Descent, which corresponds to Keras' SGD, with a momentum of 0.9 and Nesterov Accelerated Gradient active.

Below are the graphs with the validation accuracy of each model for each configuration.

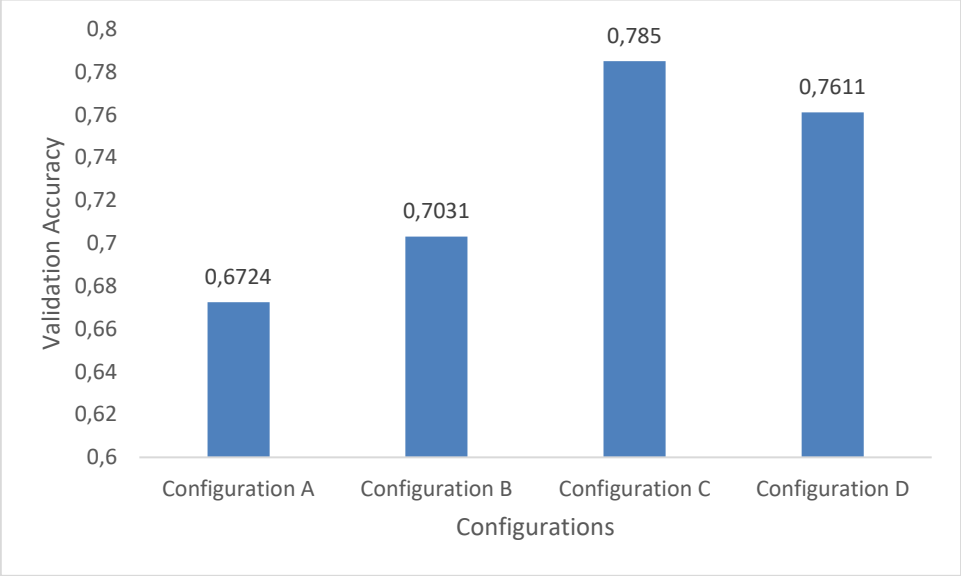

Figure 18 – Validation accuracy of the VGG16 for each of the configurations.

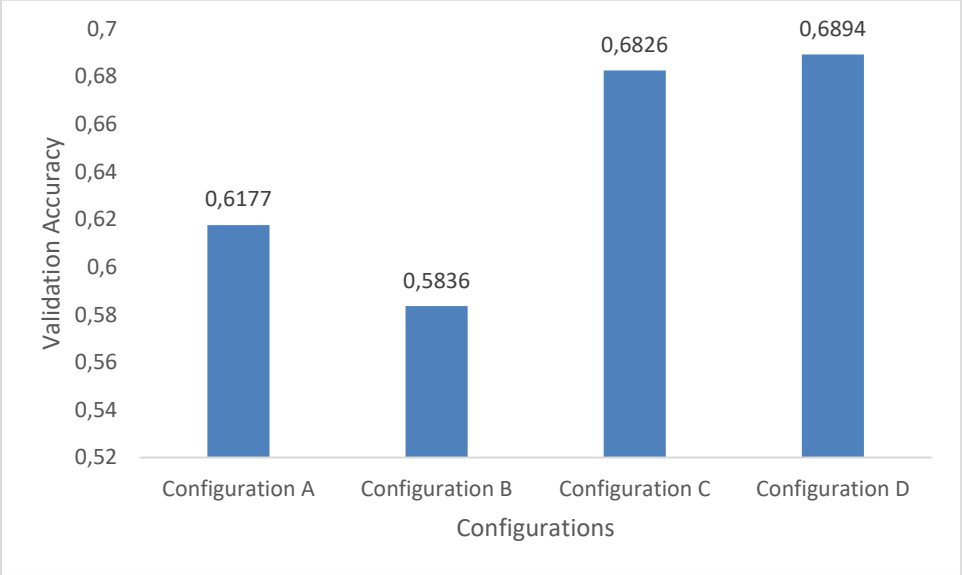

Figure 19 – Validation accuracy of the VGG19 for each of the configurations.

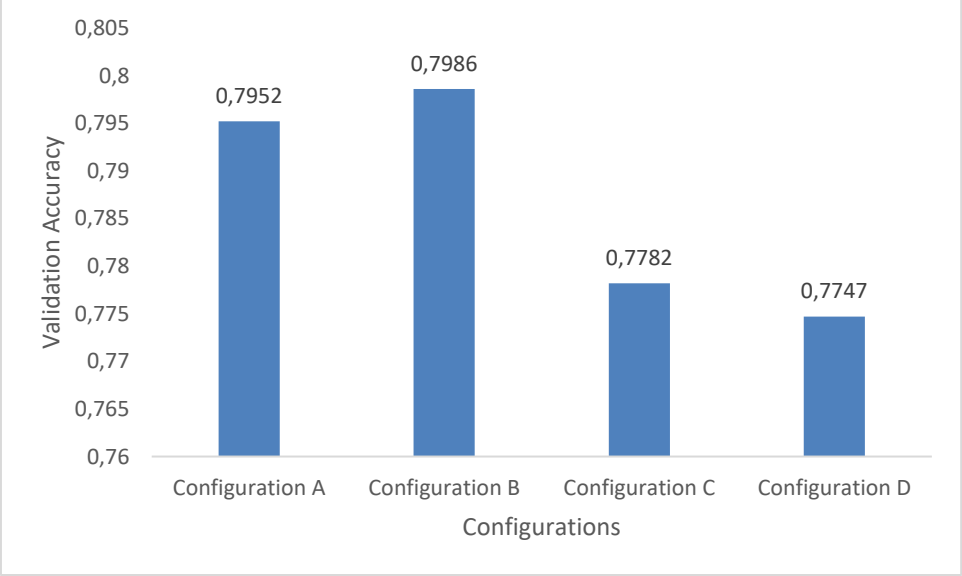

Figure 20 – Validation accuracy of the Inception-ResNet V2 for each of the configurations.

As can be seen, for VGG networks, the configurations with more filters were those that presented the highest performance, being the opposite for the Inception ResNet V2, where the configurations with 64 filters, although small, showed some advantage.

The number of neurons does not appear to be such a significant factor for performance, with only variations in the order of 3%.

Given the results obtained, were then selected the configurations with the best validation accuracy for the fine-tuning process of the final models.

The settings to be used for each of the pre-trained networks will be the following ones:

Table 4 – Selected configurations for each of the pre-trained networks.

Pre-trained network	Configuration
VGG16	C
VGG19	D
Inception ResNet V2	B

Once the best hyperparameters for the classifiers have been selected, it is then possible to proceed to fine-tuning the models that may become part of the final application.

For the fine-tuning process, it is necessary to determine the number of layers of each pre-trained network to train. To support this decision were taken into account the architecture of the pre-trained networks and the approaches described in fine-tuning section (2.5.9).

The first factor to consider was the similarity between the source domain, ILSVRC, and the target domain, KDEF. Although the ILSVRC presents images for more than 1000 different classes, none of them refers to people or their facial expressions. Therefore, it will be necessary to consider the two domains as non similar, justifying the use of fine-tuning.

Regarding the size of the target domain's dataset, we will have to consider KDEF as a reduced dataset, so it is necessary to fine-tune the majority of the pre-trained network's layers.

Since both networks are structured in blocks, the division of the layers to train followed that structure. For the VGG, were enabled the last three convolution blocks, conv5, conv4, and conv3. For the Inception-ResNet V2, were enabled the Inception-resnet-C, Reduction-B, and Inception-resnet-B.

Once the number of layers to apply fine-tuning was determined, the networks were trained with the best configurations of the convolution layer approach and for the approach with global average pooling layer.

4.4 Evaluation

After having trained and tuned the models for the best possible result in the validation data, it is then necessary to assess their performance. For this, will be used the test data from the KDEF dataset, as well as the CK + and NetImages.

Ideally, it is pursued the model that presents the best result in both three sets. However, if none of the models has that clear advantage, will be selected the one with the higher performances summation in the three datasets.

In this phase, it will be evaluated six models—three models with a classifier with a convolutional layer and another three with a GAP layer classifier.

In order to abbreviate the names of the models, will be used the following designations:

- **VGG16:** Corresponds to the pre-trained network VGG16, with the convolutional layer classifier, following the configuration C;
- **VGG19:** Corresponds to the pre-trained network VGG19, with the convolutional layer classifier, following the configuration D;
- **IRNV2:** Corresponds to the pre-trained network Inception-ResNet V2, with the convolutional layer classifier, following the configuration B;
- **VGG16 GAP:** Corresponds to the pre-trained network VGG16, with the global average pooling layer classifier;
- **VGG19 GAP:** Corresponds to the pre-trained network VGG19, with the global average pooling layer classifier;
- **IRNV2 GAP:** Corresponds to the pre-trained network Inception-ResNet V2, with the global average pooling layer classifier;

4.4.1 Metrics, Indicators and Sources of Information

To evaluate the performance of these six models created will be used confusion matrices. These allow the representation, in a simple but complete manner, of the classifications made by the models. Once in possession of this source of information, it is then possible to extract several metrics to evaluate the models. The most commonly used metrics are accuracy (2), precision (3), recall (4), and F1 score (5).

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN} \tag{2}$$

$$Precision = \frac{TP}{TP + FP} \tag{3}$$

$$Recall = \frac{TP}{TP + FN} \tag{4}$$

$$F1\ score = 2 * \frac{Precision * Recall}{Precision + Recall} = \frac{2TP}{2TP + FP + FN} \tag{5}$$

However, even though during the training of neural networks, was used accuracy to evaluate the models, this was only possible, because they were trained and validated with a balanced dataset, KDEF. That is, for each class, there is the same number of instances. However, in this evaluation phase of the final models, will be used an unbalanced dataset, the CK +.

Thus, it is necessary to use an alternative metric, capable of evaluating both the results of balanced datasets (KDEF and Net Images), as well as unbalanced (CK +).

For this, was selected the metric Mathew Correlation Coefficient (MCC) (6), where it is recommended in (Chicco & Jurman, 2020; Jurman et al., 2012).

$$MCC = \frac{TP * TN - FP * FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \quad (6)$$

The MCC returns values between -1 and 1. Where a coefficient of 1 represents a perfect prediction, 0 an average random prediction and -1 a perfect inverse prediction.

4.4.2 Muticlass Evaluation

The models were created for a multiclass classification, where, except the CK + dataset, any of the classes is equally probable. For the evaluation of these models, were generated confusion matrices for each combination of the models with the three datasets. However, this makes a total of eighteen different confusion matrices. In order to abbreviate the document, for each dataset, will be presented a graph with the accuracy and MCC of the six models. All the confusion matrices and tables with metrics by facial expression will be present in Appendix A.

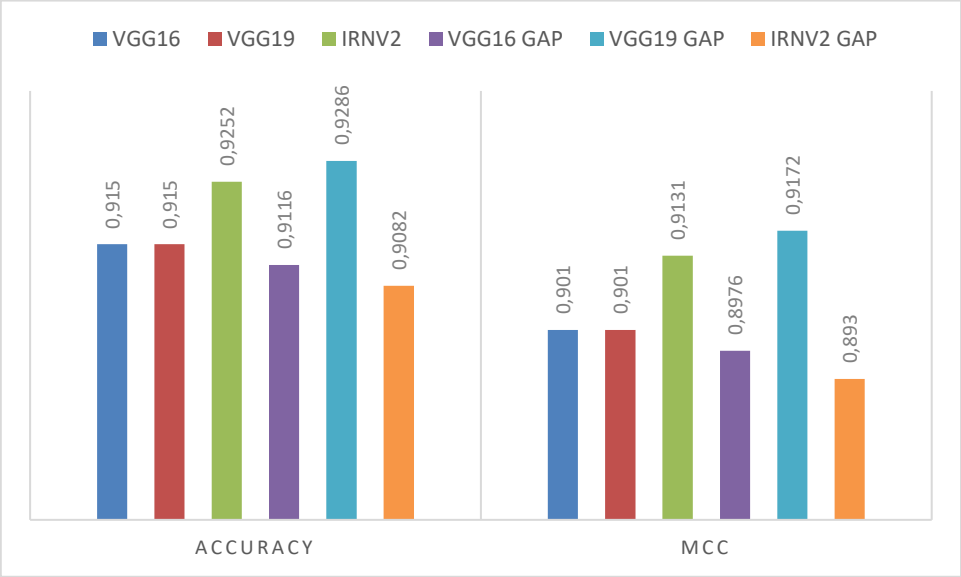

Figure 21 – KDEF Test Data multiclass evaluation.

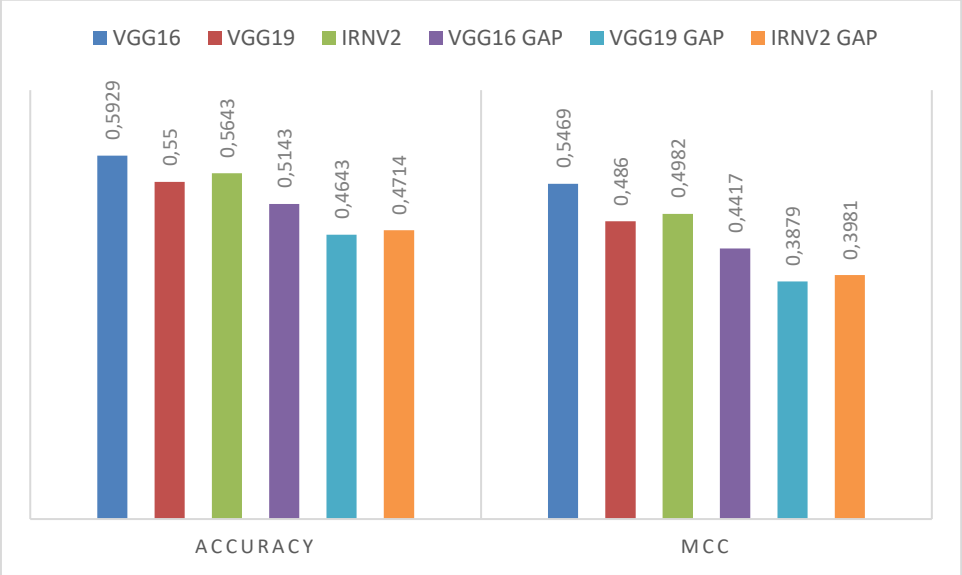

Figure 22 – Net Images multiclass evaluation.

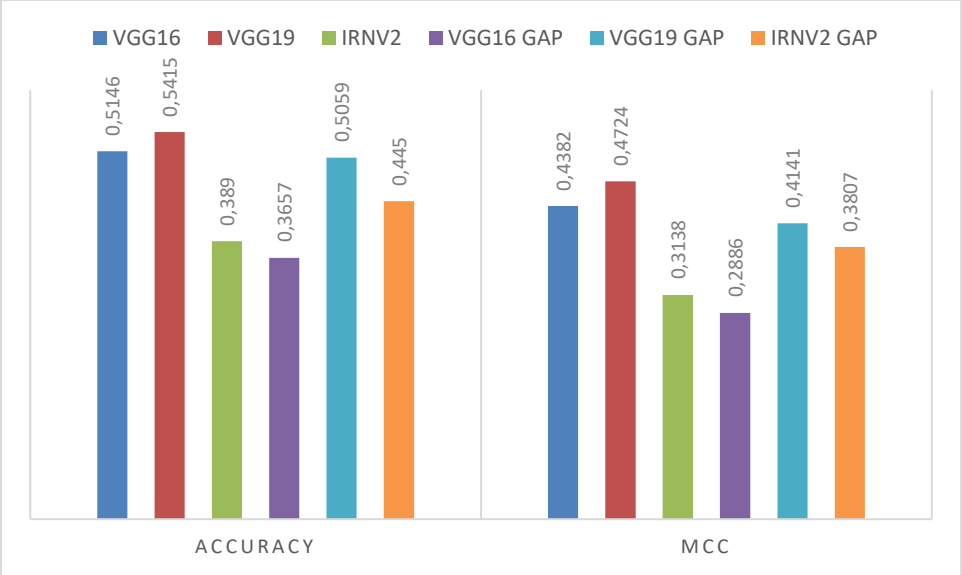

Figure 23 – CK+ multiclass evaluation.

From the data collected, it is now necessary to determine which model performed the best. As referred previously, it is pursued the model that presents the best result in both three sets. However, that case did not occur. Therefore, it will be calculated the sum of the models MCCs. The model that presents the higher value is the one with the best performance across all datasets.

Table 5 – Calculations for the best model in a multiclass evaluation.

Models	KDEF MCC	Net Images MCC	CK+ MCC	Sum
VGG16	0.901	0.5469	0.4382	1.8861
VGG19	0.901	0.486	0.4724	1.8594
IRNV2	0.9131	0.4982	0.3138	1.7251
VGG16 GAP	0.8976	0.4417	0.2886	1.6279
VGG19 GAP	0.9172	0.3879	0.4141	1.7192
IRNV2 GAP	0.893	0.3981	0.3807	1.6718

As can be seen in Table 5, the VGG16 was the model that showed the higher sum, it can then be concluded that this is the best model at classifying facial expressions.

In order to demonstrate the VGG16 multiclass classification performance in more detail, are shown in Figure 24, Figure 25, and Figure 26, the model confusion matrices for each of the datasets. Are also presented the Table 6, Table 7, and Table 8, with the metrics, precision, recall, and f1 score for each facial expression, together with the weighted average for each of those metrics.

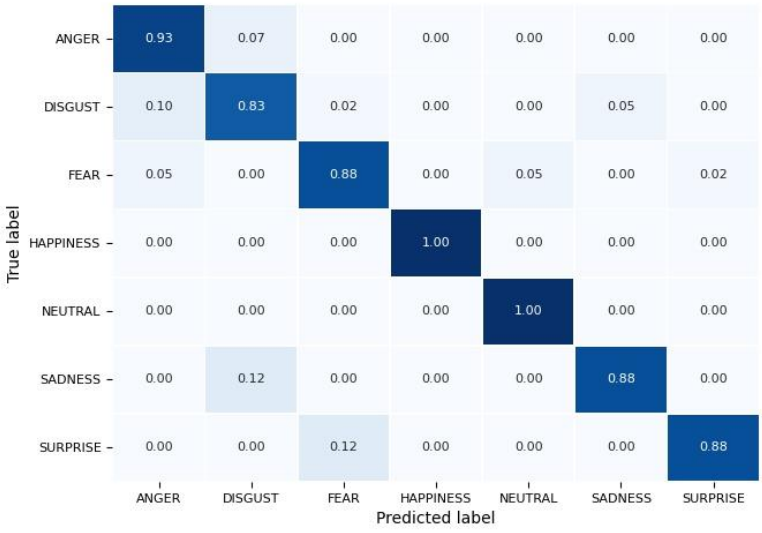

Figure 24 – Confusion matrix of the VGG16 model for the KDEF test data.

Table 6 – Metrics of the VGG16 model for the KDEF tet data.

	Precision	Recall	F1 Score	Support
Anger	0.8667	0.9286	0.8966	42
Disgust	0.814	0.8333	0.8235	42
Fear	0.8605	0.8810	0.8706	42
Happiness	1.0000	1.0000	1.0000	42
Neutral	0.9545	1.0000	0.9767	42
Sadness	0.9487	0.8810	0.9136	42
Surprise	0.9737	0.8810	0.9250	42
Weighted Average	0.9169	0.9150	0.9151	294

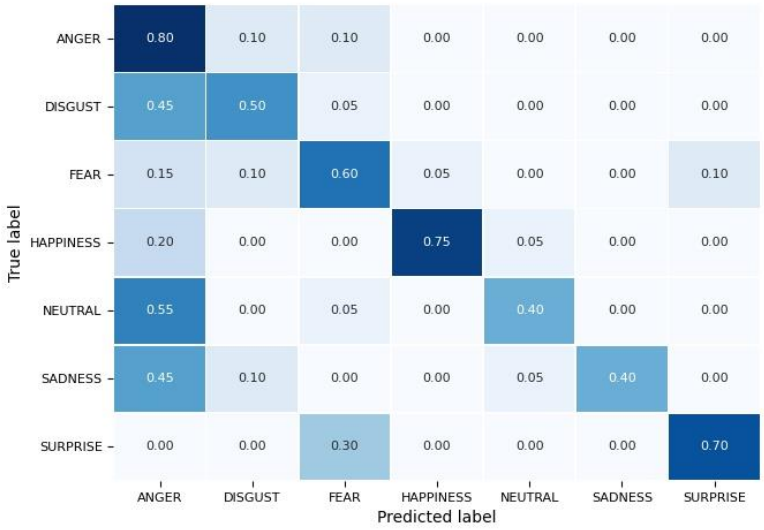

Figure 25 – Confusion matrix of the VGG16 model for the Net Images dataset.

Table 7 – Metrics of the VGG16 model for the Net Images dataset.

	Precision	Recall	F1 Score	Support
Anger	0.3077	0.8000	0.4444	20
Disgust	0.6250	0.5000	0.5556	20
Fear	0.5455	0.6000	0.5714	20
Happiness	0.9375	0.7500	0.8333	20
Neutral	0.8000	0.4000	0.5333	20
Sadness	1.0000	0.4000	0.5714	20
Surprise	0.8750	0.7000	0.7778	20
Weighted Average	0.7272	0.5929	0.6125	140

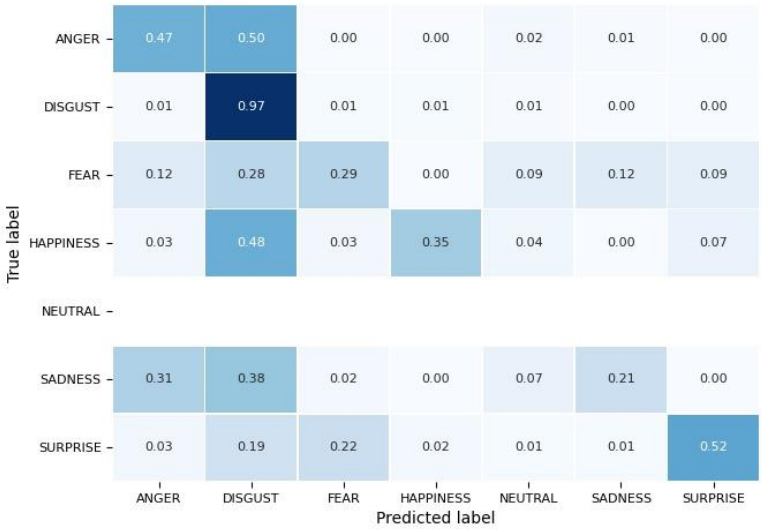

Figure 26 – Confusion matrix of the VGG16 model for the CK+ dataset.

Table 8 – Metrics of the VGG16 model for the CK+ dataset.

	Precision	**Recall**	**F1 Score**	**Support**
Anger	0.5565	0.4741	0.5120	135
Disgust	0.3904	0.9661	0.5561	177
Fear	0.2529	0.2933	0.2716	75
Happiness	0.9114	0.3478	0.5035	207
Neutral	0.0000	0.0000	0.0000	0
Sadness	0.6000	0.2143	0.3158	84
Surprise	0.8609	0.5221	0.6500	249
Weighted Average	0.6652	0.5146	0.5184	927

4.4.3 Binary Evaluation

However, the performance of this facial expression classification model does not directly represent the performance of the final system itself. This is because the final system addresses a binary classification problem, distinguishing between non-stressful and stressful emotions. As such, we will have to verify how these models perform in distinguishing between these two types of emotions. Even there, we will have to take into account that this is an approximation to reality with a relationship between facial expressions and stress, and the real performance of the final product may vary even more.

For this binary evaluation, were translated the model classifications into non-stressful and stressful and again, were created the confusion matrices.

As in the previous section, will be exhibit three graphs with the accuracy and MCC, with the addition of the precision, recall and f1 score for the six models.

All the confusion matrices and tables with metrics by stress and non-stres will be present in Appendix B.

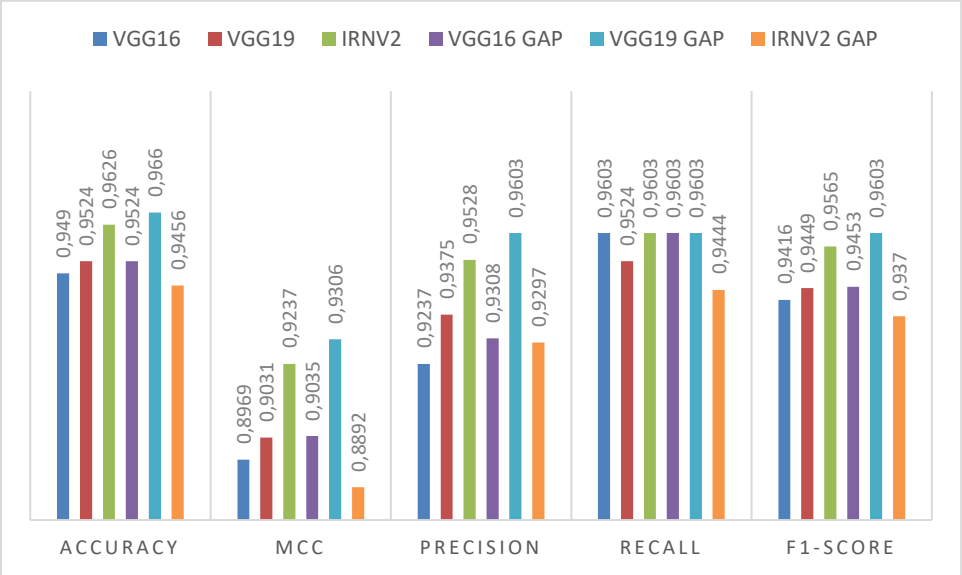

Figure 27 – KDEF Test Data binary evaluation.

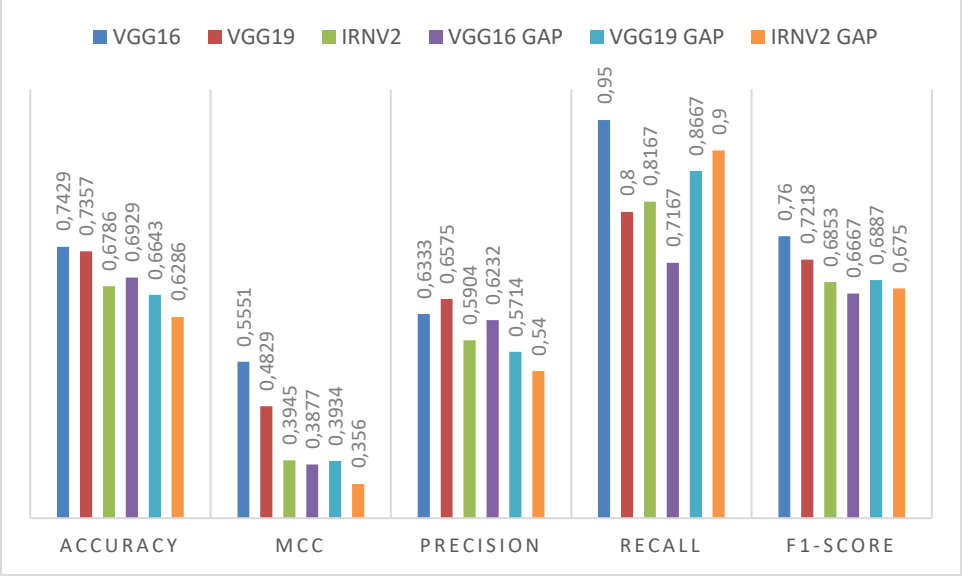

Figure 28 – Net Images binary evaluation.

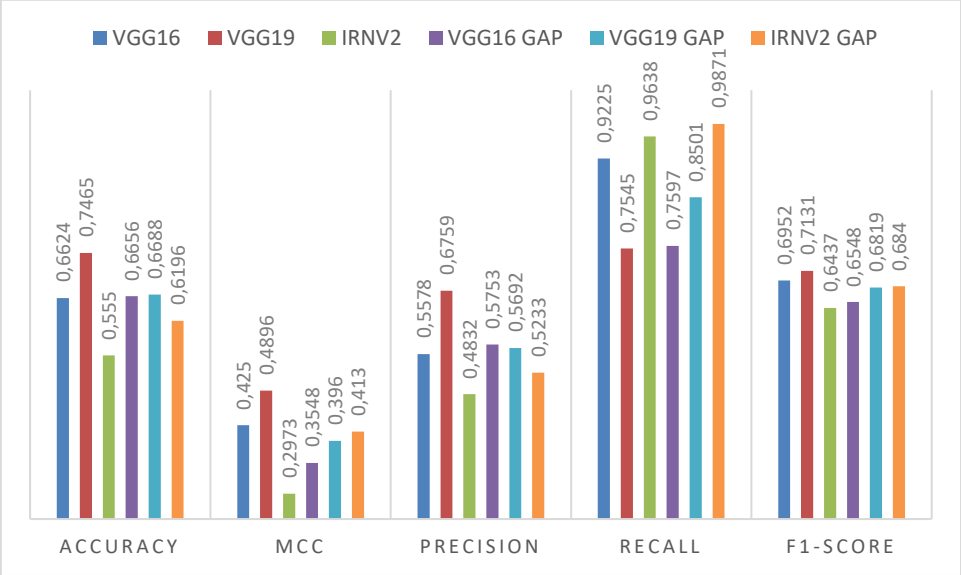

Figure 29 – CK+ binary evaluation.

Again, no model presented a superior advantage over all the others in the three datasets, so it will be necessary to calculate the sum of the models MCCs.

Table 9 – Calculations for the best model in a binary evaluation.

Models	KDEF MCC	Net Images MCC	CK+ MCC	Sum
VGG16	0.8969	0.5551	0.4250	1.8770
VGG19	0.9031	0.4829	0.4896	1.8756
IRNV2	0.9237	0.3945	0.2973	1.6155
VGG16 GAP	0.9035	0.3877	0.3548	1.6460
VGG19 GAP	0.9306	0.3934	0.3960	1.7200
IRNV2 GAP	0.8892	0.3560	0.4130	1.6582

As can be seen in Table 9, the VGG16 was the model that presented the higher sum for the binary classification. However, the difference with the VGG19 is relatively small, and it is justified to untie these two models. Since it is intended to detect the maximum possible stressful emotions, that is, maximize the True Positives, it is then necessary to look at the recall values of VGG16 and VGG19.

Table 10 – Comparison of the VGG16 and VGG19 binary recall.

Models	KDEF recall	Net Images recall	CK+ recall	Sum
VGG16	0.9603	0.9500	0.9225	2.8328
VGG19	0,9524	0.8000	0.7545	2.5069

As can be seen in Table 10, the VGG16 model is the one with the highest recall sum in the three datasets, and as such, it is the one that maximizes the true positives.

Therefore, it can be concluded that the VGG16, with the convolutional layer classifier, is the best candidate for the stress detection program.

In the Figure 30, Figure 31, and Figure 32, are displayed the VGG16's binary confusion matrices for each of the datasets, together with the Table 11, Table 12, and Table 13 with the metrics precision, recall and f1 score.

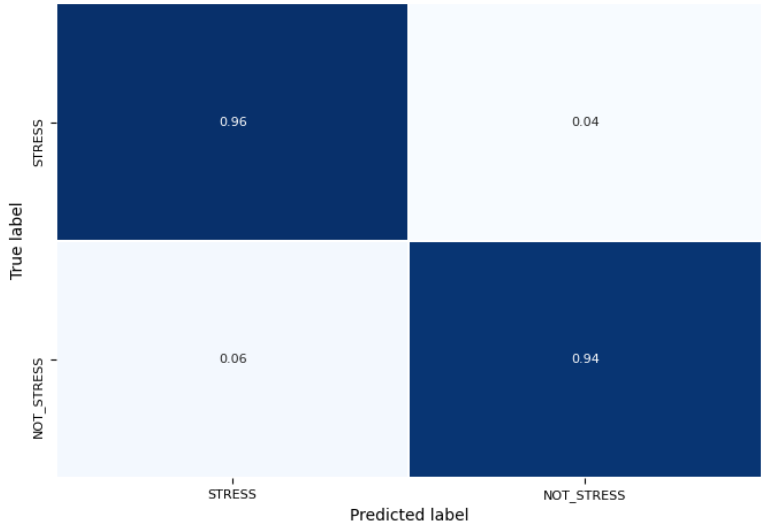

Figure 30 – Binary confusion matrix of the VGG16 model for the KDEF test data.

Table 11 – Metrics from the binary classification of the VGG16 model for the KDEF test data.

	Precision	**Recall**	**F1 Score**	**Support**
Stress	0.9237	0.9603	0.9416	126
Not stress	0.9693	0.9405	0.9547	168
Weighted Average	0.9498	0.9490	0.9491	294

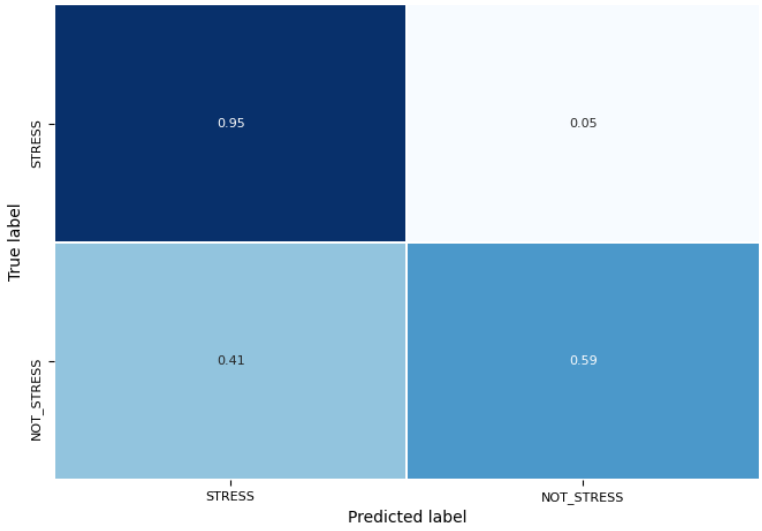

Figure 31 – Binary confusion matrix of the VGG16 model for the Net Images dataset.

Table 12 – Metrics from the binary classification of the VGG16 model for the Net Images dataset.

	Precision	**Recall**	**F1 Score**	**Support**
Stress	0.6333	0.9500	0.7600	60
Not stress	0.9400	0.5875	0.7231	80
Weighted Average	0.8086	0.7429	0.7389	140

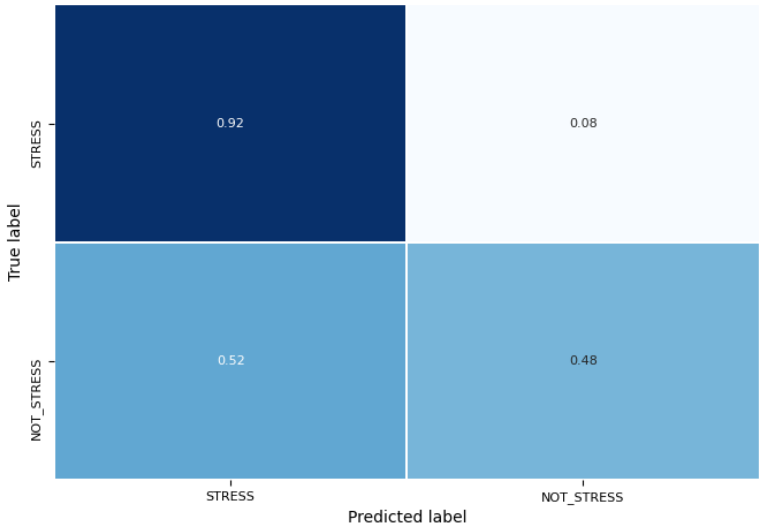

Figure 32 – Binary confusion matrix of the VGG16 model for the CK+ dataset.

Table 13 – Metrics from the binary classification of the VGG16 model for the CK+ dataset.

	Precision	**Recall**	**F1 Score**	**Support**
Stress	0.5578	0.9225	0.6952	387
Not stress	0.8955	0.4759	0.6215	540
Weighted Average	0.7545	0.6624	0.6523	927

5 Conclusion

In this chapter, are presented the conclusions about the project developed, namely the objectives achieved, the limitations found and the improvements that can be made in the future.

5.1 Achieved Goals

This project aimed to develop a system capable of detecting signs of stress from facial expressions.

The developed solution allows to capture real-time images of the user's face and, using a facial expression classifier, assess whether or not the user presents signs of stress. If so, it allows notifying the user of this fact.

Thus having been achieved the main objective of the project. However, the developed system has some limitations that can be mitigated with some future works.

5.2 Limitations

The work developed was based on some associations and adaptations in order to overcome some limitations.

The main limitations are the lack of a dataset directly classified as stress or non-stress. Due to the lack of such dataset, the system was developed based on a relationship between facial expressions and stress, in which the frequency of certain facial expressions determines the presence of signs of stress.

Despite this adaptation, the final system was not evaluated systematically for the classification of stress signs. It is therefore not possible to determine this capacity with complete certainty.

Another limitation was the small size and homogeneity of the dataset used for training the models.

5.3 Future Work

In the perspective of the document's author, this project has two paths to follow to combat the first limitation presented above. Continue with the association between facial expressions and stress, and in partnership with experts, experiment inducing volunteers into stressful situations and validate if the system correctly classifies those stressful situations. Or, obtain a dataset classified as stress or not-stress and develop a new classification model for that new dataset.

For the second limitation, could be obtained other datasets, more heterogeneous and join with the already existing KDEF in order to create a larger dataset with more variation factors so that the model generalizes better for the recognition of facial expressions.

One possible increment to the project would be the migration of the classification module to a server, therefore being able to take advantage of centralized processing with graphics cards. In this way, the system would have a lesser impact on the users' computer, leaving them only with tasks to capture the images and send them to the server to be classified.

However, this increment would have to consider the possible load of the network with the constant sending of images, and the possibility of this feed being intercepted or hijacked and thus expose the user's privacy to the internet.

Bibliography

Agarwal, A., Gupta, S., & Singh, D. K. (2016). Review of optical flow technique for moving object detection. *2016 2nd International Conference on Contemporary Computing and Informatics (IC3I)*, 409–413. https://doi.org/10.1109/IC3I.2016.7917999

American Psychological Association. (2018). *STRESS IN AMERICA: Generation Z* (Stress in America™ Survey).

Baltrušaitis, T., Robinson, P., & Morency, L.-P. (2016). OpenFace: An open source facial behavior analysis toolkit. *2016 IEEE Winter Conference on Applications of Computer Vision (WACV)*, 1–10. https://doi.org/10.1109/WACV.2016.7477553

Bartlett, Marian Stewart, Viola, P. A., Sejnowski, T. J., Golomb, B. A., Larsen, J., Hager, J. C., & Ekman, P. (1995). Classifying facial action. *Proceedings of the 8th International Conference on Neural Information Processing Systems*, 823–829.

Bartlett, M.S., Littlewort, G., Lainscsek, C., Fasel, I., & Movellan, J. (2004). Machine learning methods for fully automatic recognition of facial expressions and facial actions. *2004 IEEE International Conference on Systems, Man and Cybernetics (IEEE Cat. No.04CH37583)*, *1*, 592–597 vol.1. https://doi.org/10.1109/ICSMC.2004.1398364

Bousefsaf, F., Maaoui, C., & Pruski, A. (2013). Continuous wavelet filtering on webcam photoplethysmographic signals to remotely assess the instantaneous heart rate. *Biomedical Signal Processing and Control*, *8*(6), 568–574. https://doi.org/10.1016/j.bspc.2013.05.010

Brownlee, J. (2019, Janeiro 27). Loss and Loss Functions for Training Deep Learning Neural Networks. *Machine Learning Mastery*. https://machinelearningmastery.com/loss-and-loss-functions-for-training-deep-learning-neural-networks/

Cburnett. (2006). File:Artificial neural network.svg. Em *Wikipedia*. https://en.wikipedia.org/wiki/File:Artificial_neural_network.svg

Chang, Y., Hu, C., Feris, R., & Turk, M. (2006). Manifold based analysis of facial expression. *Image and Vision Computing*, *24*(6), 605–614. https://doi.org/10.1016/j.imavis.2005.08.006

Chicco, D., & Jurman, G. (2020). The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation. *BMC Genomics*, *21*. https://doi.org/10.1186/s12864-019-6413-7

Chollet, F. (2017). *Fchollet/deep-learning-models*. https://github.com/fchollet/deep-learning-models

Chollet, F. (2020, Abril 15). *Keras documentation: Transfer learning & fine-tuning* [Documentation]. Keras. https://keras.io/guides/transfer_learning/

Chollet, F., & Allaire, J. J. (2018a). Chapter 3: Getting started with neural networks. Em *Deep Learning with R* (1st Edition). Manning Publications.

Chollet, F., & Allaire, J. J. (2018b). Chapter 4: Fundamentals of machine learning. Em *Deep Learning with R* (1st Edition). Manning Publications.

Chollet, F., & Allaire, J. J. (2018c). Chapter 5: Deep learning for computer vision. Em *Deep Learning with R* (1st Edition). Manning Publications.

Chrousos, G. P. (2009). Stress and disorders of the stress system. *Nature Reviews Endocrinology*, *5*(7), 374–381. https://doi.org/10.1038/nrendo.2009.106

Chu, W.-S., De La Torre, F., & Cohn, J. F. (2013). *Selective Transfer Machine for Personalized Facial Action Unit Detection*. 3515–3522. http://openaccess.thecvf.com/content_cvpr_2013/html/Chu_Selective_Transfer_Machine_2013_CVPR_paper.html

Convolution. (2018, Julho 25). NVIDIA Developer. https://developer.nvidia.com/discover/convolution

Convolutional Neural Network (CNN). (2018, Abril 23). NVIDIA Developer. https://developer.nvidia.com/discover/convolutional-neural-network

Cosmar, M., Gründler, R., Flemming, D., Cosemans, B., & den Broek, K. V. (2014). *Calculating the costs of work-related stress and psychosocial risks – A literature review* [Literature Review]. European Agency for Safety and Health at Work. https://osha.europa.eu/en/publications/calculating-cost-work-related-stress-and-psychosocial-risks

Dinges, D. F., Rider, R. L., Dorrian, J., McGlinchey, E. L., Rogers, N. L., Cizman, Z., Goldenstein, S. K., Vogler, C., Venkataraman, S., & Metaxas, D. N. (2005). Optical computer recognition of facial expressions associated with stress induced by performance demands. *Aviation, Space, and Environmental Medicine*, *76*(6 Suppl), B172-182.

Ebner, N. C., Riediger, M., & Lindenberger, U. (2010). FACES—A database of facial expressions in young, middle-aged, and older women and men: Development and validation. *Behavior Research Methods*, *42*(1), 351–362. https://doi.org/10.3758/BRM.42.1.351

Ekenel, H. K., & Stiefelhagen, R. (2005). Local appearance based face recognition using discrete cosine transform. *2005 13th European Signal Processing Conference*, 1–5.

Ekman, P. (1970). Universal facial expressions of emotion. *California Mental Health Research Digest*, *8*(4), 151–158.

Ekman, P. (2016). What Scientists Who Study Emotion Agree About. *Perspectives on Psychological Science*, *11*(1), 31–34. https://doi.org/10.1177/1745691615596992

Ekman, P., & Friesen, W. V. (1976). Measuring facial movement. *Environmental Psychology and Nonverbal Behavior*, *1*(1), 56–75. https://doi.org/10.1007/BF01115465

Ekman, P., & Friesen, W. V. (2003). *Unmasking the Face: A Guide to Recognizing Emotions from Facial Clues*. ISHK.

Esler, M. (2017). Mental stress and human cardiovascular disease. *Neuroscience & Biobehavioral Reviews*, *74*, 269–276. https://doi.org/10.1016/j.neubiorev.2016.10.011

Gao, H., Yüce, A., & Thiran, J.-P. (2014). Detecting emotional stress from facial expressions for driving safety. *2014 IEEE International Conference on Image Processing (ICIP)*, 5961–5965. https://doi.org/10.1109/ICIP.2014.7026203

Giannakakis, G., Pediaditis, M., Manousos, D., Kazantzaki, E., Chiarugi, F., Simos, P. G., Marias, K., & Tsiknakis, M. (2016). Stress and anxiety detection using facial cues from videos. *Biomedical Signal Processing and Control*, *31*, 89–101. https://doi.org/10.1016/j.bspc.2016.06.020

Good news, Tensorflow chooses Keras! · Issue #5050 · keras-team/keras. (2017, Janeiro 16). GitHub. https://github.com/keras-team/keras/issues/5050

Hamm, J., Kohler, C. G., Gur, R. C., & Verma, R. (2011). Automated Facial Action Coding System for dynamic analysis of facial expressions in neuropsychiatric disorders. *Journal of Neuroscience Methods*, *200*(2), 237–256. https://doi.org/10.1016/j.jneumeth.2011.06.023

Hansen, D. W., & Ji, Q. (2010). In the Eye of the Beholder: A Survey of Models for Eyes and Gaze. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *32*(3), 478–500. https://doi.org/10.1109/TPAMI.2009.30

Hemmatian, F., & Sohrabi, M. K. (2019). A survey on classification techniques for opinion mining and sentiment analysis. *Artificial Intelligence Review*, *52*(3), 1495–1545. https://doi.org/10.1007/s10462-017-9599-6

Hung, J. C., Lin, K.-C., & Lai, N.-X. (2019). Recognizing learning emotion based on convolutional neural networks and transfer learning. *Applied Soft Computing*, *84*, 105724. https://doi.org/10.1016/j.asoc.2019.105724

Ioffe, S., & Szegedy, C. (2015). Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. *arXiv:1502.03167 [cs]*. http://arxiv.org/abs/1502.03167

Jurman, G., Riccadonna, S., & Furlanello, C. (2012). A Comparison of MCC and CEN Error Measures in Multi-Class Prediction. *PLOS ONE*, *7*(8), e41882. https://doi.org/10.1371/journal.pone.0041882

Karpathy, A., Toderici, G., Shetty, S., Leung, T., Sukthankar, R., & Fei-Fei, L. (2014). Large-Scale Video Classification with Convolutional Neural Networks. *Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition*, 1725–1732. https://doi.org/10.1109/CVPR.2014.223

Keras-team/keras. (2020). [Python]. Keras. https://github.com/keras-team/keras (Original work published 2015)

Koolhaas, J. M., Bartolomucci, A., Buwalda, B., de Boer, S. F., Flügge, G., Korte, S. M., Meerlo, P., Murison, R., Olivier, B., Palanza, P., Richter-Levin, G., Sgoifo, A., Steimer, T., Stiedl, O., van Dijk, G., Wöhr, M., & Fuchs, E. (2011). Stress revisited: A critical evaluation of the stress concept. *Neuroscience & Biobehavioral Reviews*, *35*(5), 1291–1301. https://doi.org/10.1016/j.neubiorev.2011.02.003

Krishnaiah, V., Narsimha, G., & Chandra, N. S. (2014). Survey of classification techniques in data mining. *International Journal of Computer Sciences and Engineering*, *2*(9), 65–74.

Kumar, A., Kaur, A., & Kumar, M. (2019). Face detection techniques: A review. *Artificial Intelligence Review*, *52*(2), 927–948. https://doi.org/10.1007/s10462-018-9650-2

Langner, O., Dotsch, R., Bijlstra, G., Wigboldus, D. H. J., Hawk, S. T., & Knippenberg, A. van. (2010). Presentation and validation of the Radboud Faces Database. *Cognition and Emotion*, *24*(8), 1377–1388. https://doi.org/10.1080/02699930903485076

Le, C. P., Nowell, C. J., Kim-Fuchs, C., Botteri, E., Hiller, J. G., Ismail, H., Pimentel, M. A., Chai, M. G., Karnezis, T., Rotmensz, N., Renne, G., Gandini, S., Pouton, C. W., Ferrari, D., Möller, A., Stacker, S. A., & Sloan, E. K. (2016). Chronic stress in mice remodels lymph vasculature to promote tumour cell dissemination. *Nature Communications*, *7*(1), 1–14. https://doi.org/10.1038/ncomms10634

Lerner, J. S., Dahl, R. E., Hariri, A. R., & Taylor, S. E. (2007). Facial Expressions of Emotion Reveal Neuroendocrine and Cardiovascular Stress Responses. *Biological Psychiatry*, *61*(2), 253–260. https://doi.org/10.1016/j.biopsych.2006.08.016

Lucey, P., Cohn, J. F., Kanade, T., Saragih, J., Ambadar, Z., & Matthews, I. (2010). The Extended Cohn-Kanade Dataset (CK+): A complete dataset for action unit and emotion-specified expression. *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Workshops*, 94–101. https://doi.org/10.1109/CVPRW.2010.5543262

Lundqvist, D., Flykt, A., & Öhman, A. (1998). *The Karolinska Directed Emotional Faces (KDEF), CD-ROM from Department of Clinical Neuroscience, Psychology section, Karolinska Institutet, ISBN 91-630-7164-9*.

Lupien, S. J., Juster, R.-P., Raymond, C., & Marin, M.-F. (2018). The effects of chronic stress on the human brain: From neurotoxicity, to vulnerability, to opportunity. *Frontiers in Neuroendocrinology*, *49*, 91–105. https://doi.org/10.1016/j.yfrne.2018.02.001

Maaoui, C., Bousefsaf, F., & Pruski, A. (2015). Automatic human stress detection based on webcam photoplethysmographic signals. *Journal of Mechanics in Medicine and Biology*, *16*. https://doi.org/10.1142/S0219519416500391

Marcelino, P. (2018, Outubro 23). *Transfer learning from pre-trained models*. Medium. https://towardsdatascience.com/transfer-learning-from-pre-trained-models-f2393f124751

Matić, V., Jordačević, D., & Živković, S. (2018, Novembro 10). #013 CNN VGG 16 and VGG 19. *Master Data Science*. http://datahacker.rs/deep-learning-vgg-16-vs-vgg-19/

Matosin, N., Cruceanu, C., & Binder, E. B. (2017). Preclinical and Clinical Evidence of DNA Methylation Changes in Response to Trauma and Chronic Stress. *Chronic Stress*, *1*, 2470547017710764. https://doi.org/10.1177/2470547017710764

Maxion, R. A., & Lau, S. (2018). Facial Expressions During Stress and Non-Stress Conditions: A Benchmark Video Collection. *Tech. Rep. CMU-CS-18–110, Computer Science Department, Carnegie Mellon University, Pittsburgh, Pennsylvania, USA 15213*.

McCorry, L. K. (2007). Physiology of the Autonomic Nervous System. *American Journal of Pharmaceutical Education*, *71*(4). https://www.ncbi.nlm.nih.gov/pmc/articles/PMC1959222/

McEwen, B. S. (2017). Neurobiological and Systemic Effects of Chronic Stress. *Chronic Stress*, *1*, 2470547017692328. https://doi.org/10.1177/2470547017692328

Ming-Hsuan Yang, Kriegman, D. J., & Ahuja, N. (2002). Detecting faces in images: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *24*(1), 34–58. https://doi.org/10.1109/34.982883

Myers, D. G. (2008). *English: A diagram of the General Adaptation Syndrome model.* Exploring Psychology 7th ed. (Worth) page 398. https://commons.wikimedia.org/wiki/File:General_Adaptation_Syndrome.jpg

Pantic, M., & Patras, I. (2006). Dynamics of facial expression: Recognition of facial actions and their temporal segments from face profile image sequences. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, *36*(2), 433–449. https://doi.org/10.1109/TSMCB.2005.859075

Piatetsky, G. (2019, Maio). Python leads the 11 top Data Science, Machine Learning platforms: Trends and Analysis. *KDnuggets*. https://www.kdnuggets.com/python-leads-the-11-top-data-science-machine-learning-platforms-trends-and-analysis.html/

Polikovsky, S., Kameda, Y., & Ohta, Y. (2009). *Facial micro-expressions recognition using high speed camera and 3D-gradient descriptor*. 16–16. https://doi.org/10.1049/ic.2009.0244

Porter, S., & ten Brinke, L. (2008). Reading Between the Lies: Identifying Concealed and Falsified Emotions in Universal Facial Expressions. *Psychological Science*, *19*(5), 508–514. https://doi.org/10.1111/j.1467-9280.2008.02116.x

Revina, I. M., & Emmanuel, W. R. S. (2018). A Survey on Human Face Expression Recognition Techniques. *Journal of King Saud University - Computer and Information Sciences*. https://doi.org/10.1016/j.jksuci.2018.09.002

Ruder, S. (2017). An overview of gradient descent optimization algorithms. *arXiv:1609.04747 [cs]*. http://arxiv.org/abs/1609.04747

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A. C., & Fei-Fei, L. (2015). ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision*, *115*(3), 211–252. https://doi.org/10.1007/s11263-015-0816-y

scikit-learn team. (2020). *3.1. Cross-validation: Evaluating estimator performance—Scikit-learn 0.23.2 documentation* [Documentation]. scikit-learn. https://scikit-learn.org/stable/modules/cross_validation.html#cross-validation

Selye, H. (1946). THE GENERAL ADAPTATION SYNDROME AND THE DISEASES OF ADAPTATION. *The Journal of Clinical Endocrinology & Metabolism*, *6*(2), 117–230. https://doi.org/10.1210/jcem-6-2-117

Selye, H. (1950). The physiology and pathology of exposure to stress. *Acta Medica Publ.*

Selye, H. (1975). Confusion and Controversy in the Stress Field. *Journal of Human Stress*, *1*(2), 37–44.

Shawon, A. (2018, Outubro 16). *CKPLUS*. https://kaggle.com/shawon10/ckplus

Shreve, M., Godavarthy, S., Goldgof, D., & Sarkar, S. (2011). Macro- and micro-expression spotting in long videos using spatio-temporal strain. *Face and Gesture 2011*, 51–56. https://doi.org/10.1109/FG.2011.5771451

Shreve, M., Godavarthy, S., Manohar, V., Goldgof, D., & Sarkar, S. (2009). Towards macro- and micro-expression spotting in video using strain patterns. *2009 Workshop on Applications of Computer Vision (WACV)*, 1–6. https://doi.org/10.1109/WACV.2009.5403044

Simonyan, K., & Zisserman, A. (2015). Very Deep Convolutional Networks for Large-Scale Image Recognition. *arXiv:1409.1556 [cs]*. http://arxiv.org/abs/1409.1556

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *Journal of Machine Learning Research*, *15*(56), 1929–1958.

Szegedy, C., Ioffe, S., Vanhoucke, V., & Alemi, A. (2016). Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning. *arXiv:1602.07261 [cs]*. http://arxiv.org/abs/1602.07261

Third European Survey of Enterprises on New and Emerging Risks (European Survey of Enterprises on New and Emerging Risks). (2019). [Reports]. European Agency for Safety and Health at Work. https://osha.europa.eu/en/publications/third-european-survey-enterprises-new-and-emerging-risks-esener-3/view

Thompson, D. (2017, Abril 17). *More Americans suffering from stress, anxiety and depression, study finds* [News]. CBS News. https://www.cbsnews.com/news/stress-anxiety-depression-mental-illness-increases-study-finds/

TNS Political & Social. (2014). *Working Conditions* (Survey N. 398; Flash Eurobarometer). European Commission. https://ec.europa.eu/commfrontoffice/publicopinion/flash/fl_398_sum_en.pdf

Viegas, C., Lau, S.-H., Maxion, R., & Hauptmann, A. (2018). Towards Independent Stress Detection: A Dependent Model Using Facial Action Units. *2018 International Conference on Content-Based Multimedia Indexing (CBMI)*, 1–6. https://doi.org/10.1109/CBMI.2018.8516497

Viola, P., & Jones, M. J. (2004). Robust Real-Time Face Detection. *International Journal of Computer Vision*, *57*(2), 137–154. https://doi.org/10.1023/B:VISI.0000013087.49260.fb

Viola, P., & Jones, M. J. (2001). Rapid object detection using a boosted cascade of simple features. *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*, *1*, I–I. https://doi.org/10.1109/CVPR.2001.990517

Wang, W., Stuijk, S., & de Haan, G. (2015). Exploiting Spatial Redundancy of Image Sensor for Motion Robust rPPG. *IEEE Transactions on Biomedical Engineering*, *62*(2), 415–425. https://doi.org/10.1109/TBME.2014.2356291

Wirth, R., & Hipp, J. (2000). *CRISP-DM: Towards a standard process model for data mining*. 29–39.

Wu, Q., Shen, X., & Fu, X. (2011). The Machine Knows What You Are Hiding: An Automatic Micro-expression Recognition System. Em S. D'Mello, A. Graesser, B. Schuller, & J.-C. Martin (Eds.), *Affective Computing and Intelligent Interaction* (pp. 152–162). Springer. https://doi.org/10.1007/978-3-642-24571-8_16

Xiao, J., Moriyama, T., Kanade, T., & Cohn, J. F. (2003). Robust full-motion recovery of head by dynamic templates and re-registration techniques. *International Journal of Imaging Systems and Technology*, *13*(1), 85–94. https://doi.org/10.1002/ima.10048

Xu, F., Zhang, J., & Wang, J. Z. (2017). Microexpression Identification and Categorization Using a Facial Dynamics Map. *IEEE Transactions on Affective Computing*, *8*(2), 254–267. https://doi.org/10.1109/TAFFC.2016.2518162

Zhang, C., & Zhang, Z. (2010). *A Survey of Recent Advances in Face Detection*. https://www.microsoft.com/en-us/research/publication/a-survey-of-recent-advances-in-face-detection/

Zhao, X., & Zhang, S. (2016). A Review on Facial Expression Recognition: Feature Extraction and Classification. *IETE Technical Review*, *33*(5), 505–517. https://doi.org/10.1080/02564602.2015.1117403

Appendix A – Multiclass evaluation

VGG 16 multiclass evaluation

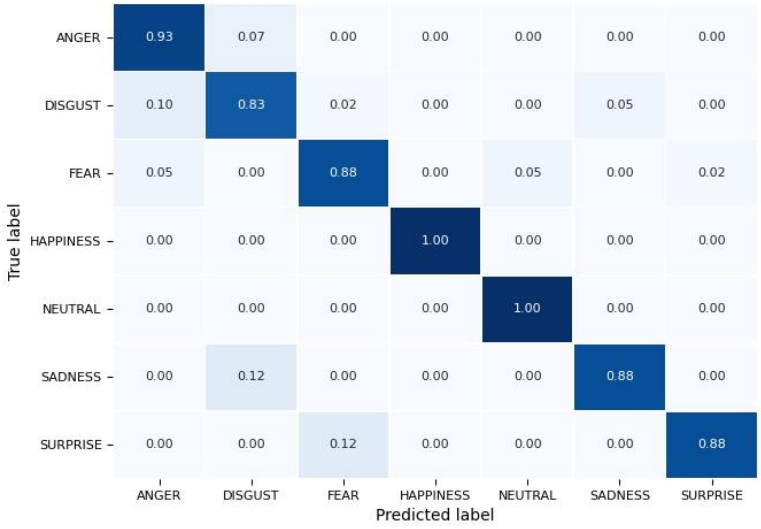

Figure 33 – Multiclass confusion matrix of the VGG16 model for the KDEF test data.

Table 14 – Multiclass metrics of the VGG16 model for the KDEF test data.

	Precision	**Recall**	**F1 Score**	**Support**
Anger	0.8667	0.9286	0.8966	42
Disgust	0.814	0.8333	0.8235	42
Fear	0.8605	0.8810	0.8706	42
Happiness	1.0000	1.0000	1.0000	42
Neutral	0.9545	1.0000	0.9767	42
Sadness	0.9487	0.8810	0.9136	42
Surprise	0.9737	0.8810	0.9250	42
Weighted Average	0.9169	0.9150	0.9151	294

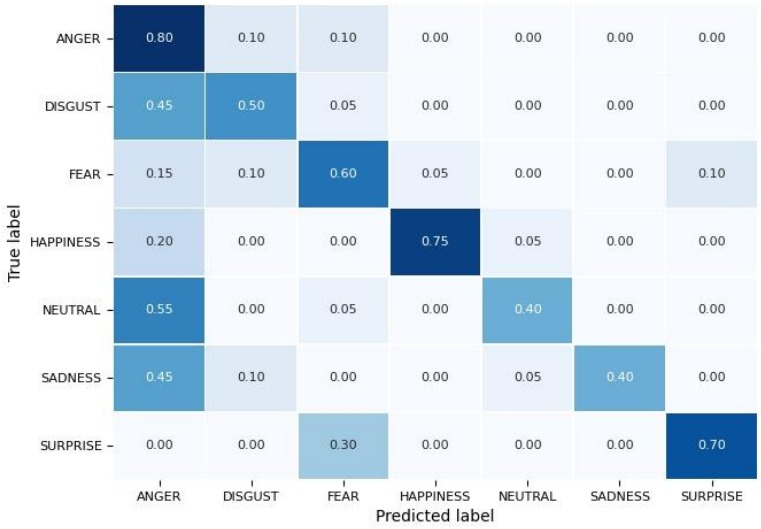

Figure 34 – Multiclass confusion matrix of the VGG16 model for the Net Images dataset.

Table 15 – Multiclass metrics of the VGG16 model for the Net Images dataset.

	Precision	**Recall**	**F1 Score**	**Support**
Anger	0.3077	0.8000	0.4444	20
Disgust	0.6250	0.5000	0.5556	20
Fear	0.5455	0.6000	0.5714	20
Happiness	0.9375	0.7500	0.8333	20
Neutral	0.8000	0.4000	0.5333	20
Sadness	1.0000	0.4000	0.5714	20
Surprise	0.8750	0.7000	0.7778	20
Weighted Average	0.7272	0.5929	0.6125	140

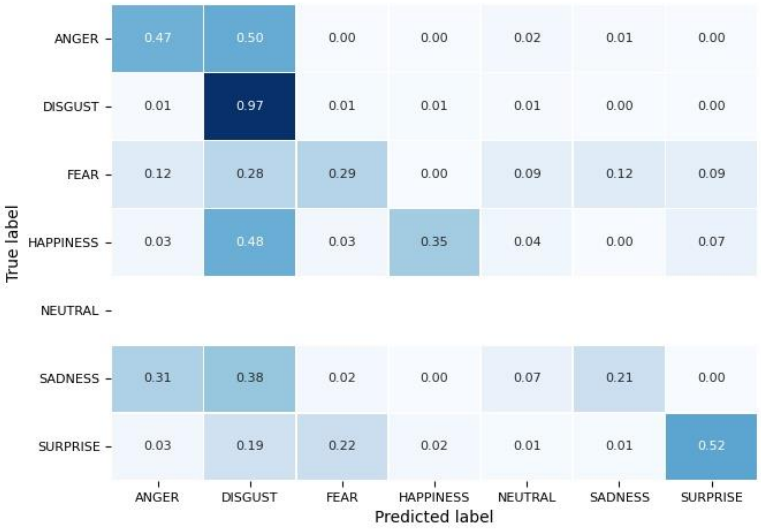

Figure 35 – Multiclass confusion matrix of the VGG16 model for the CK+ dataset.

Table 16 – Multiclass metrics of the VGG16 model for the CK+ dataset.

	Precision	**Recall**	**F1 Score**	**Support**
Anger	0.5565	0.4741	0.5120	135
Disgust	0.3904	0.9661	0.5561	177
Fear	0.2529	0.2933	0.2716	75
Happiness	0.9114	0.3478	0.5035	207
Neutral	0.0000	0.0000	0.0000	0
Sadness	0.6000	0.2143	0.3158	84
Surprise	0.8609	0.5221	0.6500	249
Weighted Average	0.6652	0.5146	0.5184	927

VGG 19 multiclass evaluation

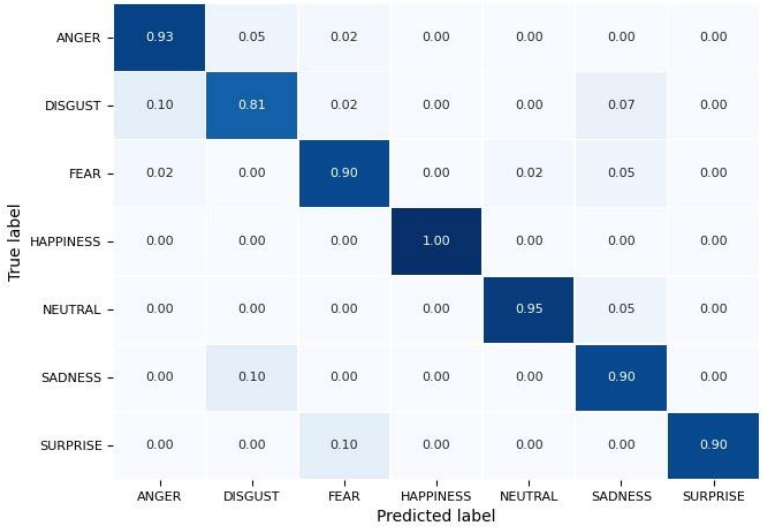

Figure 36 – Multiclass confusion matrix of the VGG19 model for the KDEF test data.

Table 17 – Multiclass metrics of the VGG19 model for the KDEF test data.

	Precision	**Recall**	**F1 Score**	**Support**
Anger	0.8864	0.9286	0.9070	42
Disgust	0.8500	0.8095	0.8293	42
Fear	0.8636	0.9048	0.8837	42
Happiness	1.0000	1.0000	1.0000	42
Neutral	0.9756	0.9524	0.9639	42
Sadness	0.8444	0.9048	0.8736	42
Surprise	1.0000	0.9048	0.9500	42
Weighted Average	0.9172	0.9150	0.9153	294

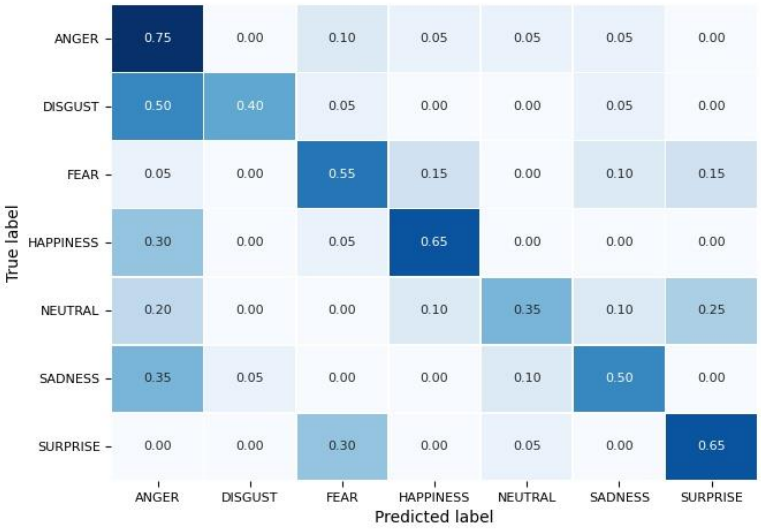

Figure 37 – Multiclass confusion matrix of the VGG19 model for the Net Images dataset.

Table 18 – Multiclass metrics of the VGG19 model for the Net Images dataset.

	Precision	**Recall**	**F1 Score**	**Support**
Anger	0.3488	0.7500	0.4762	20
Disgust	0.8889	0.4000	0.5517	20
Fear	0.5238	0.5500	0.5366	20
Happiness	0.6842	0.6500	0.6667	20
Neutral	0.6364	0.3500	0.4516	20
Sadness	0.6250	0.5000	0.5556	20
Surprise	0.6190	0.6500	0.6341	20
Weighted Average	0.6180	0.5500	0.5532	140

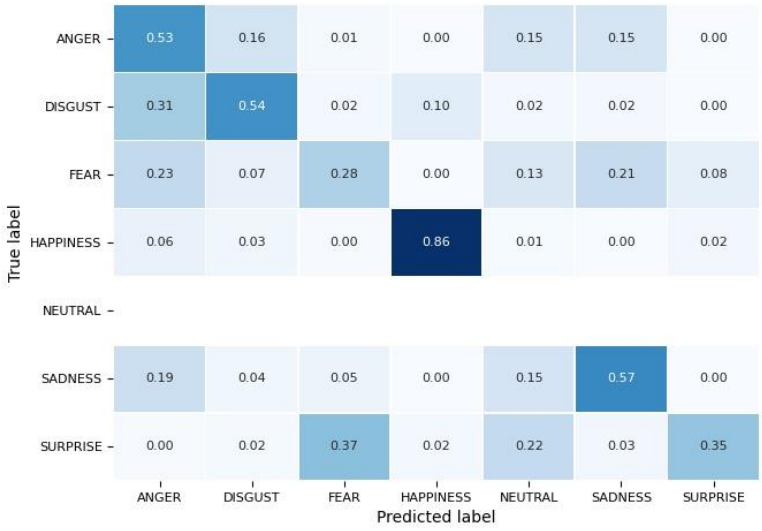

Figure 38 – Multiclass confusion matrix of the VGG19 model for the CK+ dataset.

Table 19 – Multiclass metrics of the VGG19 model for the CK+ dataset.

	Precision	**Recall**	**F1 Score**	**Support**
Anger	0.4138	0.5333	0.4660	135
Disgust	0.7059	0.5424	0.6134	177
Fear	0.1721	0.2800	0.2132	75
Happiness	0.8905	0.8647	0.8775	207
Neutral	0.0000	0.0000	0.0000	0
Sadness	0.5106	0.5714	0.5393	84
Surprise	0.8866	0.3454	0.4971	249
Weighted Average	0.6922	0.5415	0.5806	927

Inception-ResNet V2 multiclass evaluation

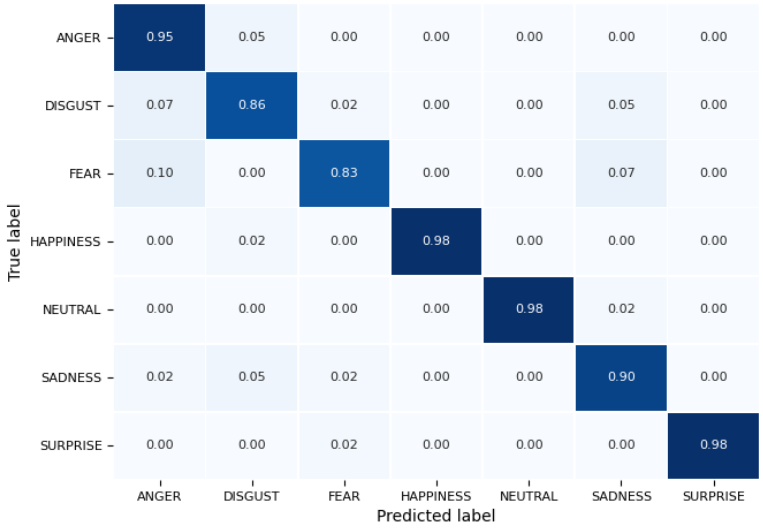

Figure 39 – Multiclass confusion matrix of the Inception-ResNet V2 model for the KDEF test data.

Table 20 – Multiclass metrics of the Inception-ResNet V2 model for the KDEF test data.

	Precision	**Recall**	**F1 Score**	**Support**
Anger	0.8333	0.9524	0.8889	42
Disgust	0.8780	0.8571	0.8675	42
Fear	0.9211	0.8333	0.8750	42
Happiness	1.0000	0.9762	0.9880	42
Neutral	1.0000	0.9762	0.9880	42
Sadness	0.8636	0.9048	0.8837	42
Surprise	1.0000	0.9762	0.9880	42
Weighted Average	0.9280	0.9252	0.9256	294

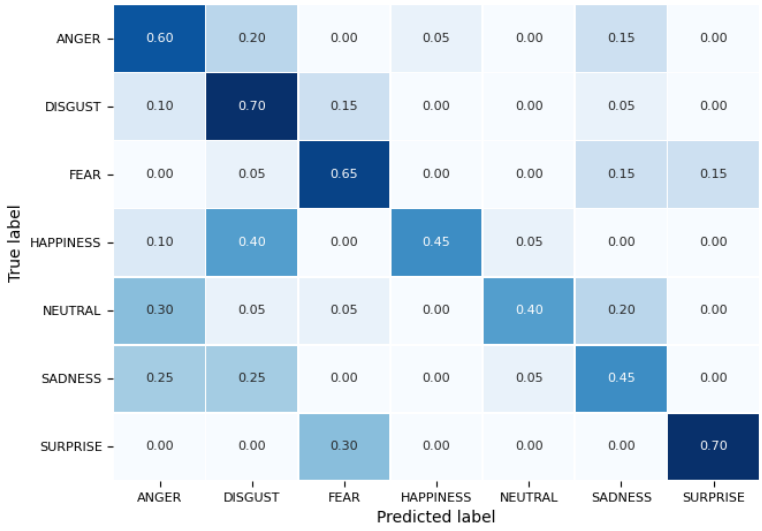

Figure 40 – Multiclass confusion matrix of the Inception-ResNet V2 model for the Net Images dataset.

Table 21 – Multiclass metrics of the Inception-ResNet V2 model for the Net Images dataset.

	Precision	**Recall**	**F1 Score**	**Support**
Anger	0.4444	0.6000	0.5106	20
Disgust	0.4242	0.7000	0.5283	20
Fear	0.5652	0.6500	0.6047	20
Happiness	0.9000	0.4500	0.6000	20
Neutral	0.8000	0.4000	0.5333	20
Sadness	0.4500	0.4500	0.4500	20
Surprise	0.8235	0.7000	0.7568	20
Weighted Average	0.6296	0.5643	0.5691	140

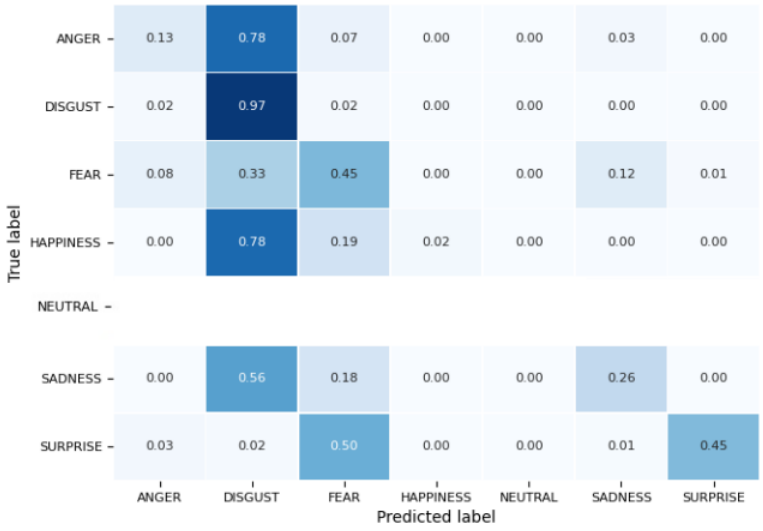

Figure 41 – Multiclass confusion matrix of the Inception-ResNet V2 model for the CK+ dataset.

Table 22 – Multiclass metrics of the Inception-ResNet V2 model for the CK+ dataset.

	Precision	**Recall**	**F1 Score**	**Support**
Anger	0.5152	0.1259	0.2024	135
Disgust	0.3333	0.9661	0.4957	177
Fear	0.1504	0.4533	0.2259	75
Happiness	1.0000	0.0242	0.0472	207
Neutral	0.0000	0.0000	0.0000	0
Sadness	0.5789	0.2619	0.3607	84
Surprise	0.9911	0.4458	0.6150	249
Weighted Average	0.6928	0.3884	0.3508	927

VGG16 GAP multiclass evaluation

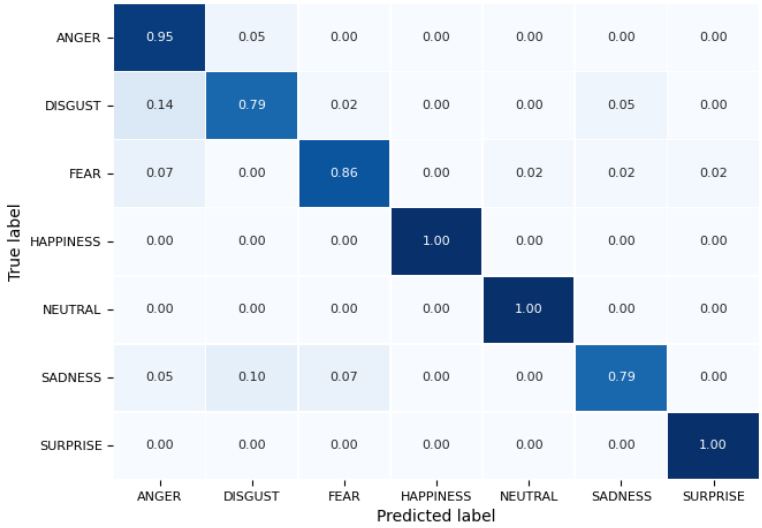

Figure 42 – Multiclass confusion matrix of the VGG16 GAP model for the KDEF test data.

Table 23 – Multiclass metrics of the VGG16 GAP model for the KDEF test data.

	Precision	**Recall**	**F1 Score**	**Support**
Anger	0.7843	0.9524	0.8602	42
Disgust	0.8462	0.7857	0.8148	42
Fear	0.9000	0.8571	0.8780	42
Happiness	1.0000	1.0000	1.0000	42
Neutral	0.9767	1.0000	0.9882	42
Sadness	0.9167	0.7857	0.8462	42
Surprise	0.9767	1.0000	0.9882	42
Weighted Average	0.9144	0.9116	0.9108	294

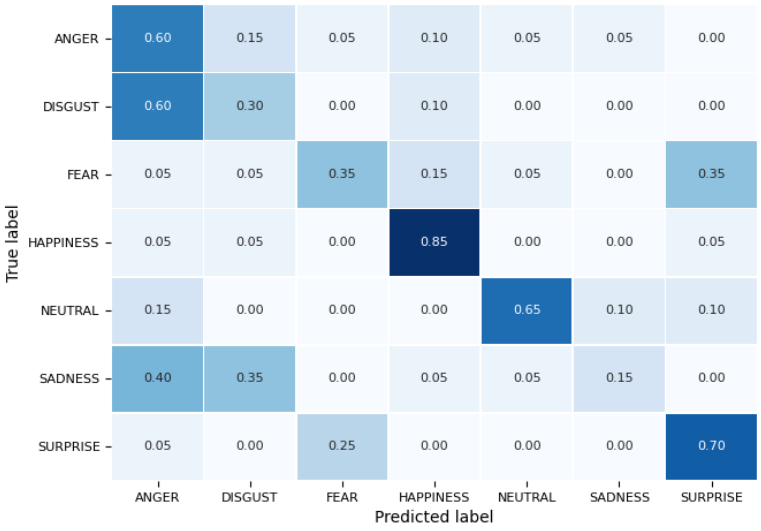

Figure 43 – Multiclass confusion matrix of the VGG16 GAP model for the Net Images dataset.

Table 24 – Multiclass metrics of the VGG16 GAP model for the Net Images dataset.

	Precision	**Recall**	**F1 Score**	**Support**
Anger	0.3158	0.6000	0.4138	20
Disgust	0.3333	0.3000	0.3158	20
Fear	0.5385	0.3500	0.4242	20
Happiness	0.6800	0.8500	0.7556	20
Neutral	0.8125	0.6500	0.7222	20
Sadness	0.5000	0.1500	0.2308	20
Surprise	0.5833	0.7000	0.6364	20
Weighted Average	0.5376	0.5143	0.4998	140

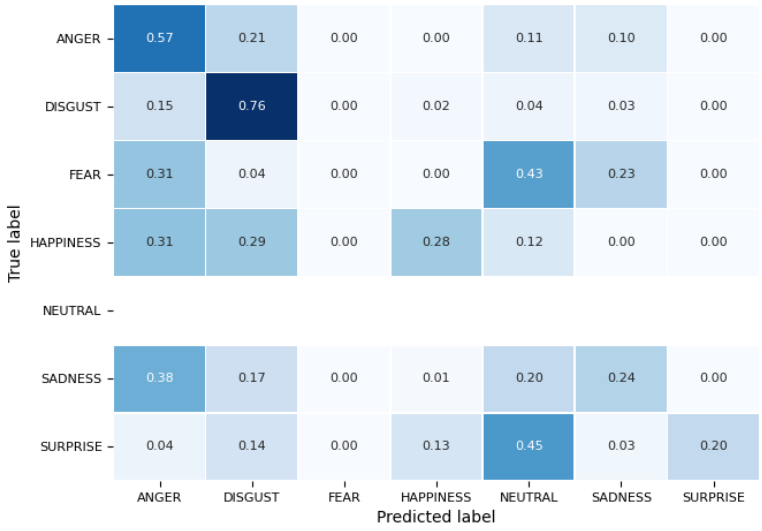

Figure 44 – Multiclass confusion matrix of the VGG16 GAP model for the CK+ dataset.

Table 25 – Multiclass metrics of the VGG16 GAP model for the CK+ dataset.

	Precision	**Recall**	**F1 Score**	**Support**
Anger	0.3277	0.5704	0.4162	135
Disgust	0.4891	0.7627	0.5960	177
Fear	0.0000	0.0000	0.0000	75
Happiness	0.6105	0.2802	0.3841	207
Neutral	0.0000	0.0000	0.0000	0
Sadness	0.3125	0.2381	0.2703	84
Surprise	1.0000	0.1968	0.3289	249
Weighted Average	0.5744	0.3657	0.3730	927

VGG19 GAP multiclass evaluation

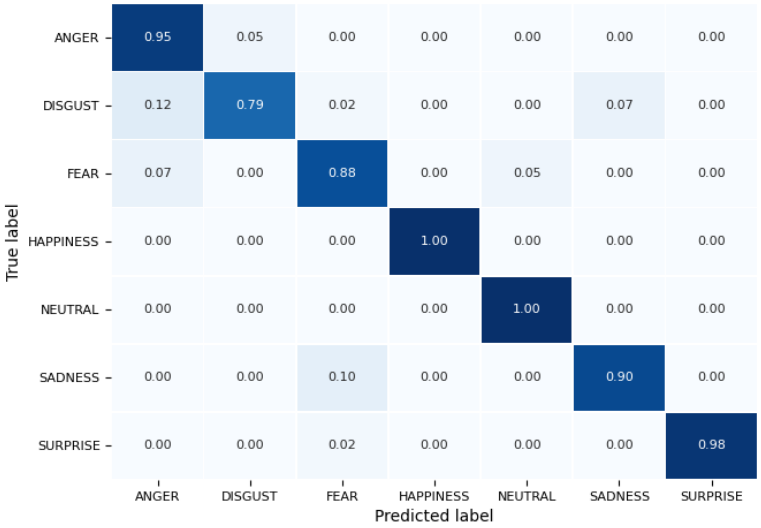

Figure 45 – Multiclass confusion matrix of the VGG19 GAP model for the KDEF test data.

Table 26 – Multiclass metrics of the VGG19 GAP model for the KDEF test data.

	Precision	**Recall**	**F1 Score**	**Support**
Anger	0.8333	0.9524	0.8889	42
Disgust	0.9429	0.7857	0.8571	42
Fear	0.8605	0.8810	0.8706	42
Happiness	1.0000	1.0000	1.0000	42
Neutral	0.9545	1.0000	0.9767	42
Sadness	0.9268	0.9048	0.9157	42
Surprise	1.0000	0.9762	0.9880	42
Weighted Average	0.9311	0.9286	0.9281	294

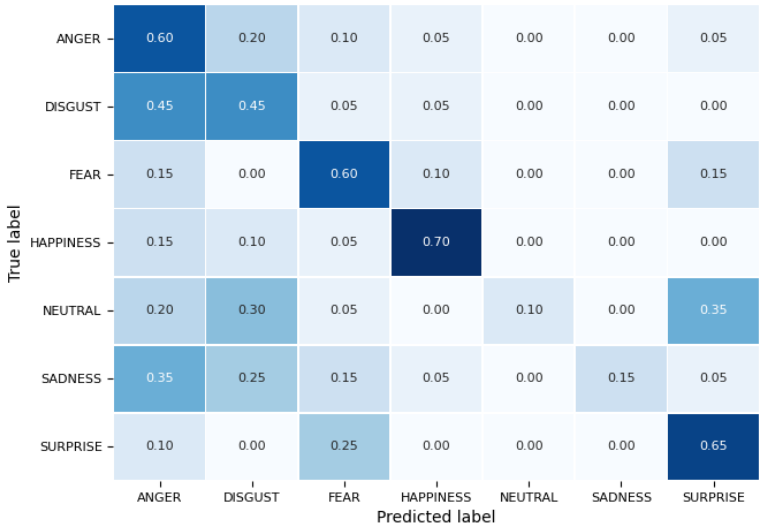

Figure 46 – Multiclass confusion matrix of the VGG19 GAP model for the Net Images dataset.

Table 27 – Multiclass metrics of the VGG19 GAP model for the Net Images dataset.

	Precision	**Recall**	**F1 Score**	**Support**
Anger	0.3000	0.6000	0.4000	20
Disgust	0.3462	0.4500	0.3913	20
Fear	0.4800	0.6000	0.5333	20
Happiness	0.7368	0.7000	0.7179	20
Neutral	1.0000	0.1000	0.1818	20
Sadness	1.0000	0.1500	0.2609	20
Surprise	0.5200	0.6500	0.5778	20
Weighted Average	0.6261	0.4643	0.4376	140

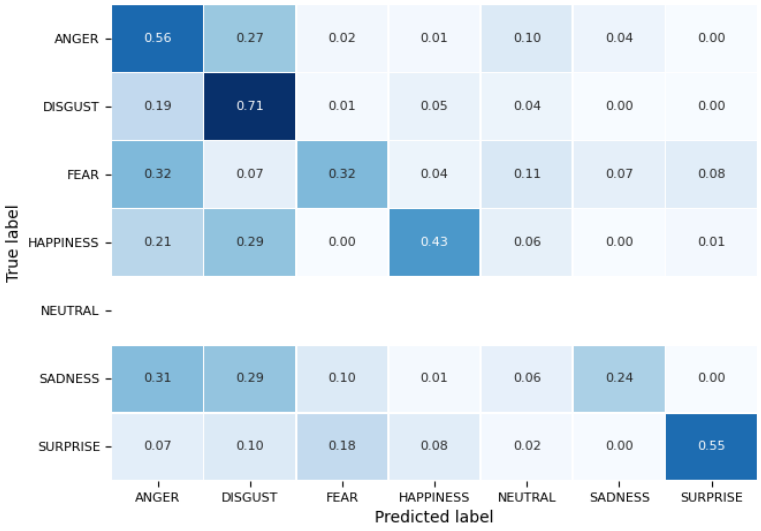

Figure 47 – Multiclass confusion matrix of the VGG19 GAP model for the CK+ dataset.

Table 28 – Multiclass metrics of the VGG19 GAP model for the CK+ dataset.

	Precision	**Recall**	**F1 Score**	**Support**
Anger	0.3425	0.5556	0.4237	135
Disgust	0.4565	0.7119	0.5563	177
Fear	0.2892	0.3200	0.3038	75
Happiness	0.7154	0.4251	0.5333	207
Neutral	0.0000	0.0000	0.0000	0
Sadness	0.6452	0.2381	0.3478	84
Surprise	0.9379	0.5462	0.6904	249
Weighted Average	0.6306	0.5059	0.5286	927

Inception-ResNet V2 GAP multi class evaluation

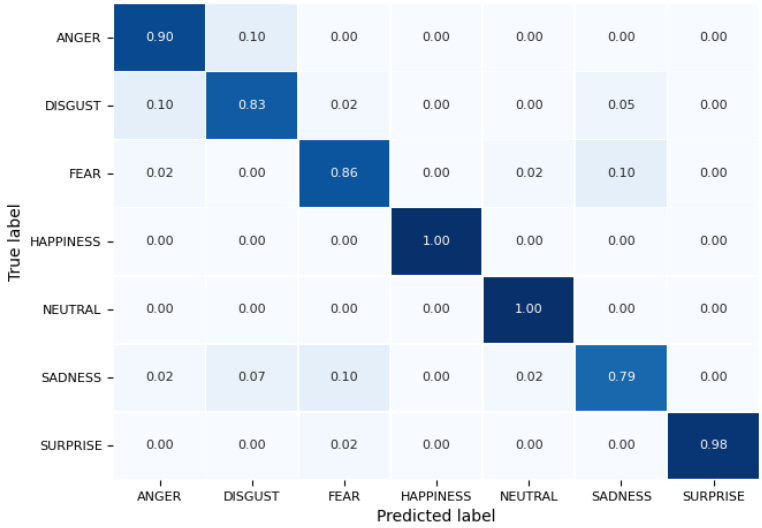

Figure 48 – Multiclass confusion matrix of the Inception-ResNet V2 GAP model for the KDEF test data.

Table 29 – Multiclass metrics of the Inception-ResNet V2 GAP model for the KDEF test data.

	Precision	**Recall**	**F1 Score**	**Support**
Anger	0.8636	0.9048	0.8837	42
Disgust	0.8333	0.8333	0.8333	42
Fear	0.8571	0.8571	0.8571	42
Happiness	1.0000	1.0000	1.0000	42
Neutral	0.9545	1.0000	0.9767	42
Sadness	0.8462	0.7857	0.8148	42
Surprise	1.0000	0.9762	0.9880	42
Weighted Average	0.9078	0.9082	0.9077	294

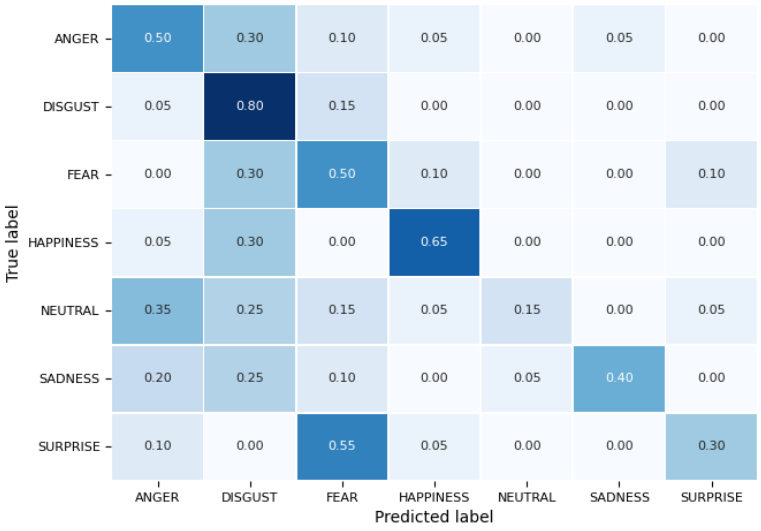

Figure 49 – Multiclass confusion matrix of the Inception-ResNet V2 GAP model for the Net Images dataset.

Table 30 – Multiclass metrics of the Inception-ResNet V2 GAP model for the Net Images dataset.

	Precision	**Recall**	**F1 Score**	**Support**
Anger	0.4000	0.5000	0.4444	20
Disgust	0.3636	0.8000	0.5000	20
Fear	0.3226	0.5000	0.3922	20
Happiness	0.7222	0.6500	0.6842	20
Neutral	0.7500	0.1500	0.2500	20
Sadness	0.8889	0.4000	0.5517	20
Surprise	0.6667	0.3000	0.4138	20
Weighted Average	0.5877	0.4714	0.4623	140

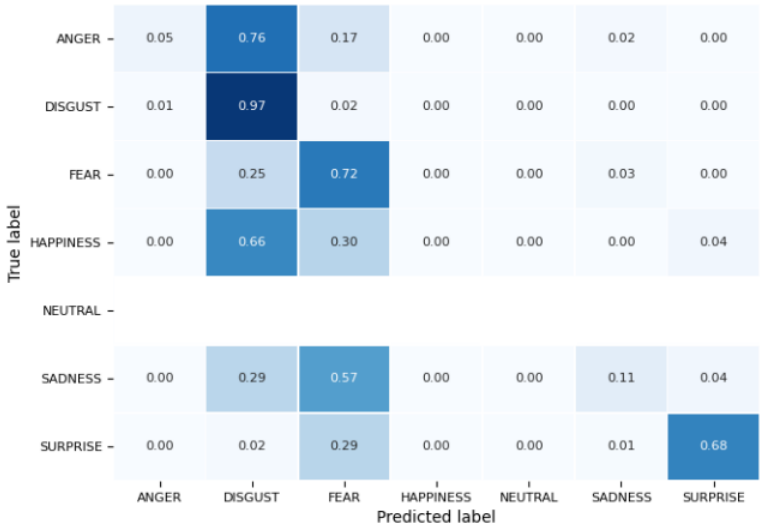

Figure 50 – Multiclass confusion matrix of the Inception-ResNet V2 GAP model for the CK+ dataset.

Table 31 – Multiclass metrics of the Inception-ResNet V2 GAP model for the CK+ dataset.

	Precision	**Recall**	**F1 Score**	**Support**
Anger	0.8750	0.0519	0.0979	135
Disgust	0.3755	0.9718	0.5417	177
Fear	0.2045	0.7200	0.3186	75
Happiness	0.0000	0.0000	0.0000	207
Neutral	0.0000	0.0000	0.0000	0
Sadness	0.5625	0.1071	0.1800	84
Surprise	0.9392	0.6827	0.7907	249
Weighted Average	0.5189	0.4444	0.3722	927

Appendix B – Binary evaluation

VGG16 binary evaluation

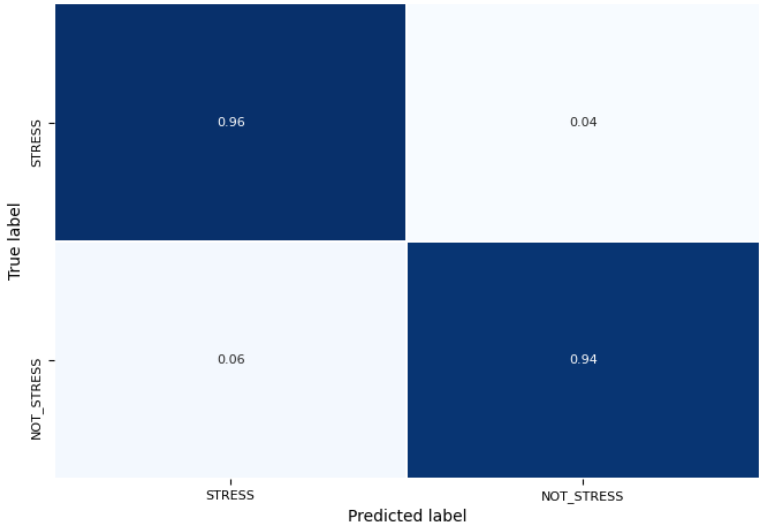

Figure 51 – Binary confusion matrix of the VGG16 model for the KDEF test data.

Table 32 – Binary metrics of the VGG16 model for the KDEF test data.

	Precision	**Recall**	**F1 Score**	**Support**
Stress	0.9237	0.9603	0.9416	126
Not stress	0.9693	0.9405	0.9547	168
Weighted Average	0.9498	0.9490	0.9491	294

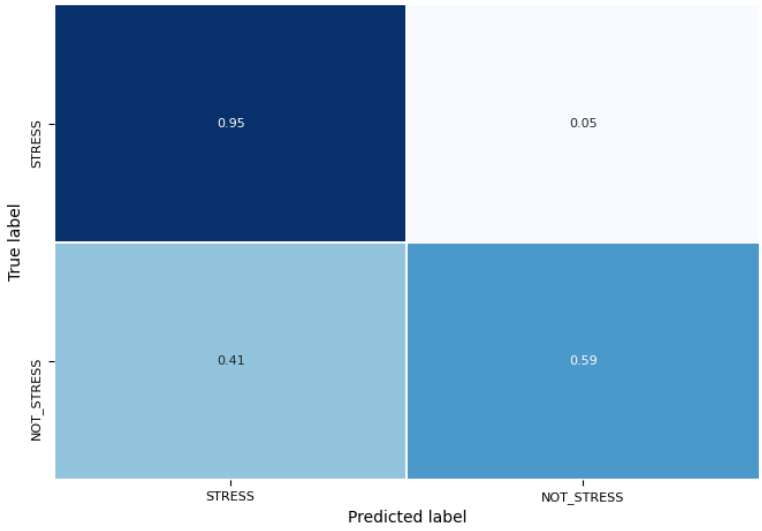

Figure 52 – Binary confusion matrix of the VGG16 model for the Net Images dataset.

Table 33 – Binary metrics of the VGG16 model for the Net Images dataset.

	Precision	**Recall**	**F1 Score**	**Support**
Stress	0.6333	0.9500	0.7600	60
Not stress	0.9400	0.5875	0.7231	80
Weighted Average	0.8086	0.7429	0.7389	140

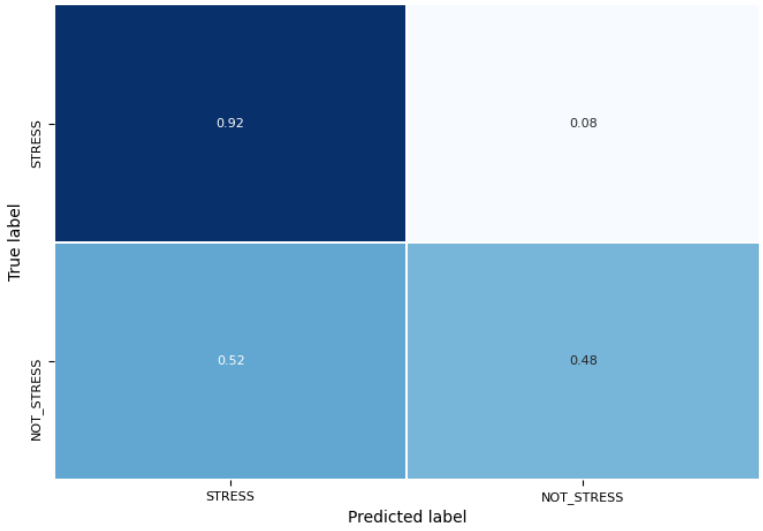

Figure 53 – Binary confusion matrix of the VGG16 model for the CK+ dataset.

Table 34 – Binary metrics of the VGG16 model for the CK+ dataset.

	Precision	**Recall**	**F1 Score**	**Support**
Stress	0.5578	0.9225	0.6952	387
Not stress	0.8955	0.4759	0.6215	540
Weighted Average	0.7545	0.6624	0.6523	927

VGG19 binary evaluation

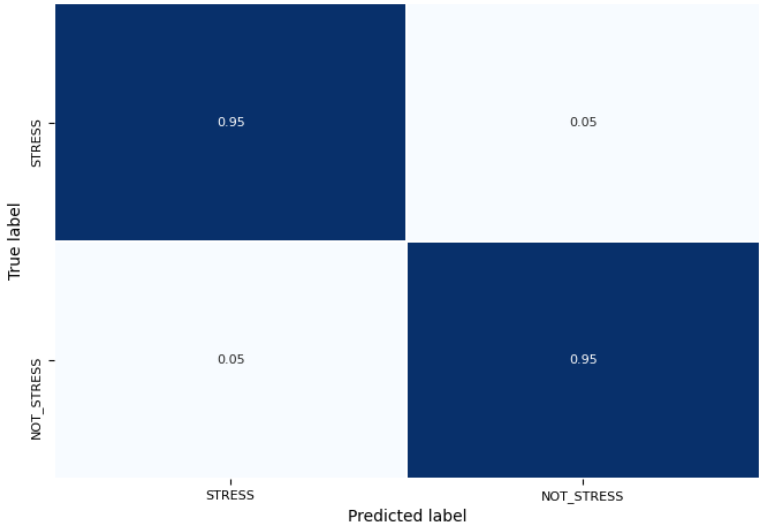

Figure 54 – Binary confusion matrix of the VGG19 model for the KDEF test data.

Table 35 – Binary metrics of the VGG19 model for the KDEF test data.

	Precision	**Recall**	**F1 Score**	**Support**
Stress	0.9375	0.9524	0.9449	126
Not stress	0.9639	0.9524	0.9581	168
Weighted Average	0.9526	0.9524	0.9524	294

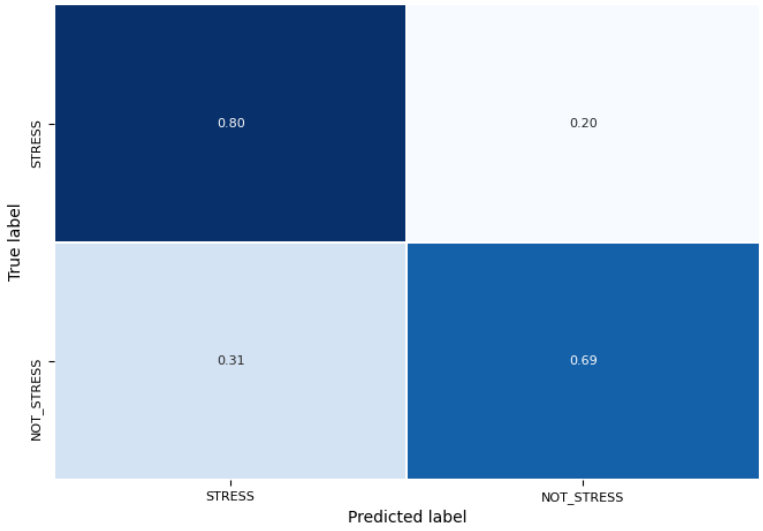

Figure 55 – Binary confusion matrix of the VGG19 model for the Net Images dataset.

Table 36 – Binary metrics of the VGG19 model for the Net Images dataset.

	Precision	**Recall**	**F1 Score**	**Support**
Stress	0.6575	0.8000	0.7218	60
Not stress	0.8209	0.6875	0.7483	80
Weighted Average	0.7509	0.7357	0.7369	140

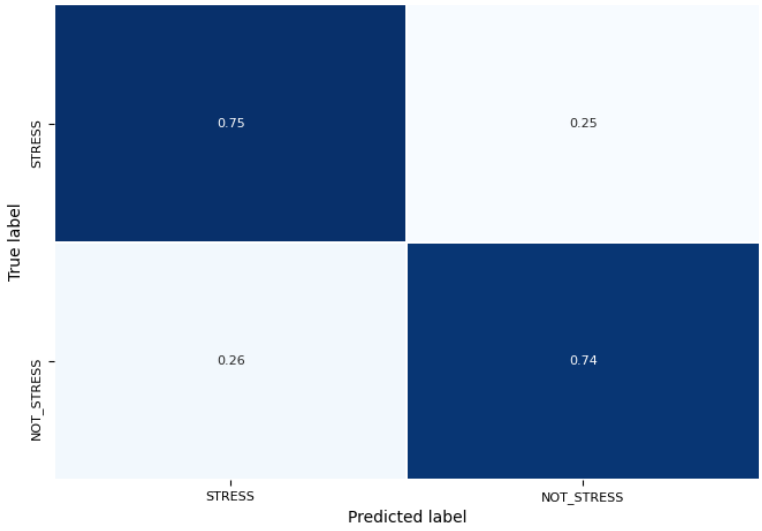

Figure 56 – Binary confusion matrix of the VGG19 model for the CK+ dataset.

Table 37 – Binary metrics of the VGG19 model for the CK+ dataset.

	Precision	**Recall**	**F1 Score**	**Support**
Stress	0.6759	0.7545	0.7131	387
Not stress	0.8081	0.7407	0.7729	540
Weighted Average	0.7529	0.7465	0.7479	927

Inception-ResNetV2 binary evaluation

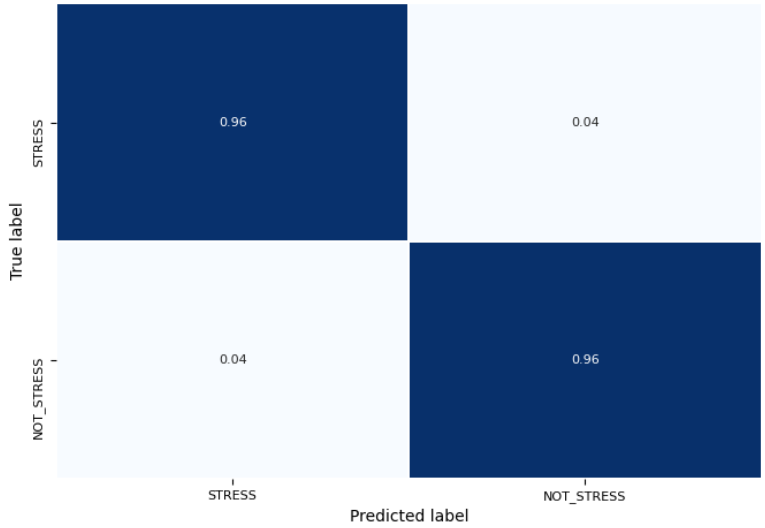

Figure 57 – Binary confusion matrix of the Inception-ResNet V2 model for the KDEF test data.

Table 38 – Binary metrics of the Inception-ResNet V2 model for the KDEF test data.

	Precision	**Recall**	**F1 Score**	**Support**
Stress	0.9528	0.9603	0.9565	126
Not stress	0.9701	0.9643	0.9672	168
Weighted Average	0.9626	0.9626	0.9626	294

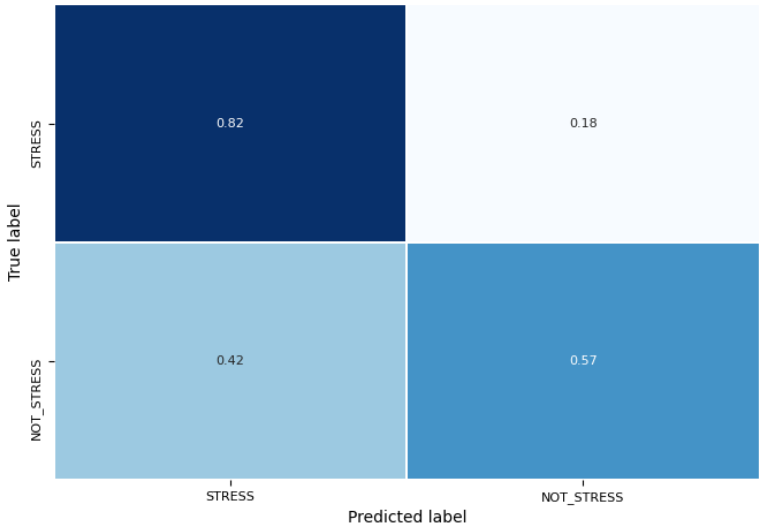

Figure 58 – Binary confusion matrix of the Inception-ResNet V2 model for the Net Images dataset.

Table 39 – Binary metrics of the Inception-ResNet V2 model for the Net Images dataset.

	Precision	**Recall**	**F1 Score**	**Support**
Stress	0.5904	0.8167	0.6853	60
Not stress	0.8070	0.5750	0.6715	80
Weighted Average	0.7142	0.6786	0.6774	140

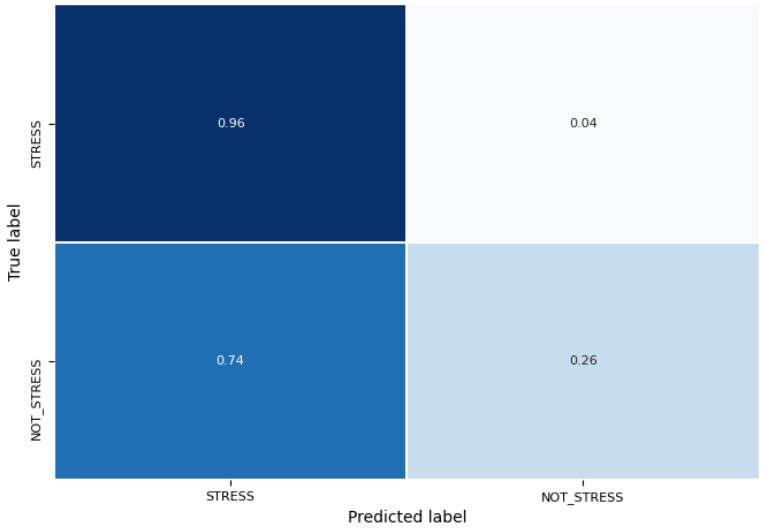

Figure 59 – Binary confusion matrix of the Inception-ResNet V2 model for the CK+ dataset.

Table 40 – Binary metrics of the Inception-ResNet V2 model for the CK+ dataset.

	Precision	**Recall**	**F1 Score**	**Support**
Stress	0.4832	0.9638	0.6437	387
Not stress	0.9103	0.2625	0.4075	540
Weighted Average	0.7321	0.5550	0.5060	927

VGG16 GAP binary evaluation

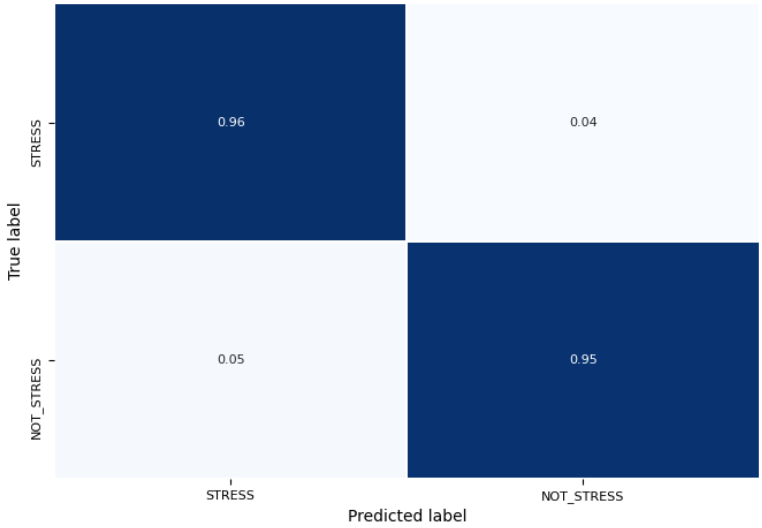

Figure 60 – Binary confusion matrix of the VGG16 GAP model for the KDEF test data.

Table 41 – Binary metrics of the VGG16 GAP model for the KDEF test data.

	Precision	**Recall**	**F1 Score**	**Support**
Stress	0.9308	0.9603	0.9453	126
Not stress	0.9695	0.9464	0.9578	168
Weighted Average	0.9529	0.9524	0.9525	294

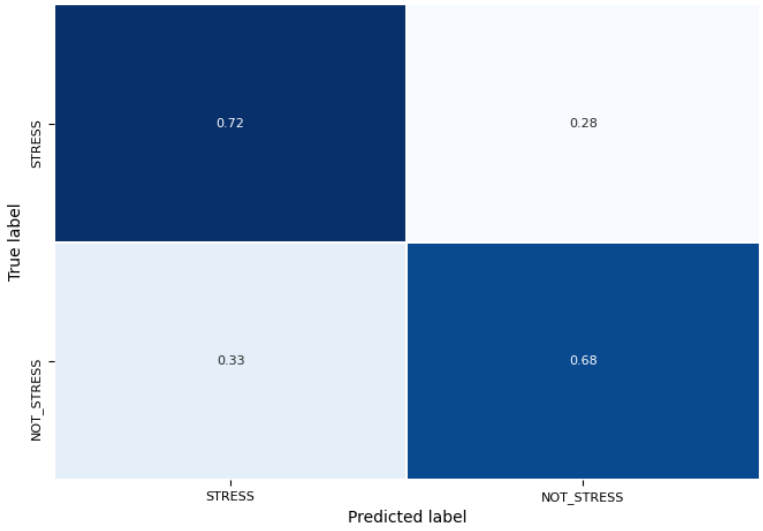

Figure 61 – Binary confusion matrix of the VGG16 GAP model for the Net Images dataset.

Table 42 – Binary metrics of the VGG16 GAP model for the Net Images dataset.

	Precision	**Recall**	**F1 Score**	**Support**
Stress	0.6232	0.7167	0.6667	60
Not stress	0.7606	0.6750	0.7152	80
Weighted Average	0.7017	0.6929	0.6944	140

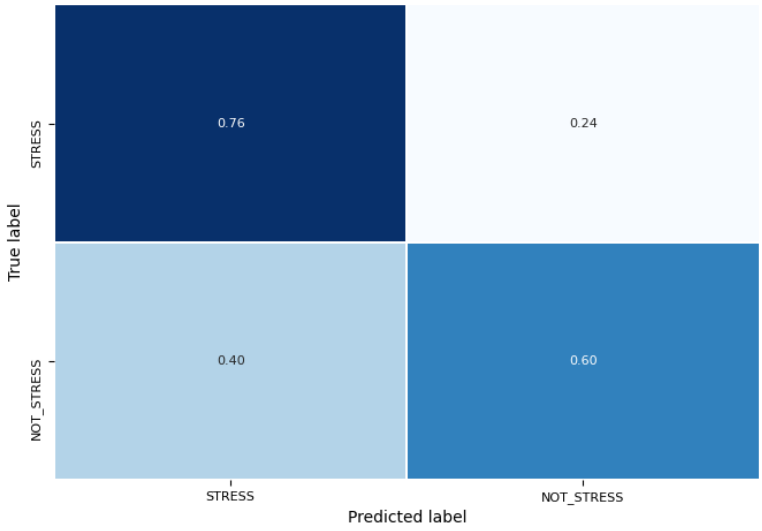

Figure 62 – Binary confusion matrix of the VGG16 GAP model for the CK+ dataset.

Table 43 – Binary metrics of the VGG16 GAP model for the CK+ dataset.

	Precision	**Recall**	**F1 Score**	**Support**
Stress	0.5753	0.7597	0.6548	387
Not stress	0.7764	0.5981	0.6757	540
Weighted Average	0.6925	0.6656	0.6670	927

VGG19 GAP binary evaluation

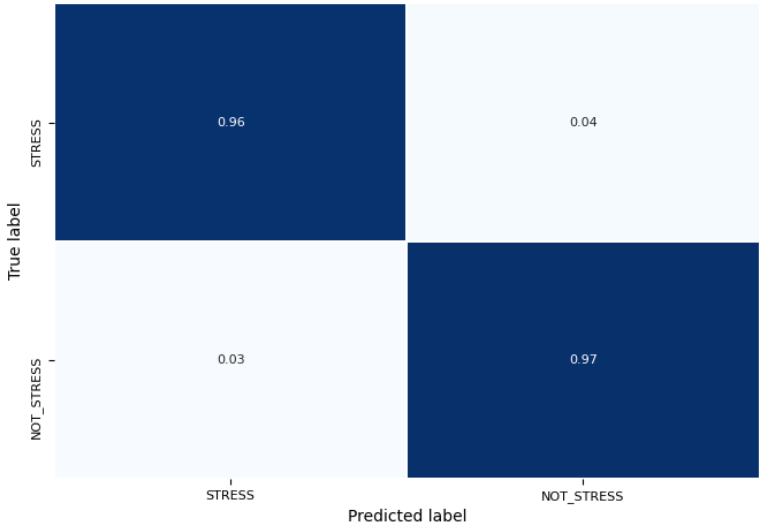

Figure 63 – Binary confusion matrix of the VGG19 GAP model for the KDEF test data.

Table 44 – Binary metrics of the VGG19 GAP model for the KDEF test data.

	Precision	**Recall**	**F1 Score**	**Support**
Stress	0.9603	0.9603	0.9603	126
Not stress	0.9702	0.9702	0.9702	168
Weighted Average	0.9660	0.9660	0.9660	294

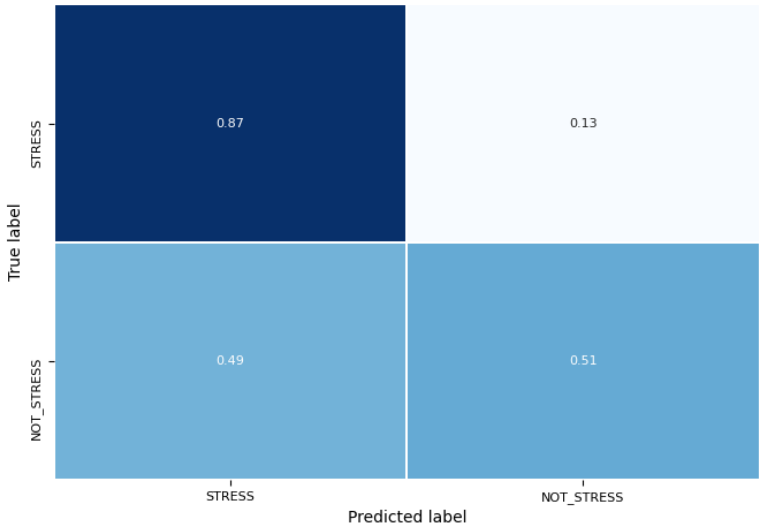

Figure 64 – Binary confusion matrix of the VGG19 GAP model for the Net Images dataset.

Table 45 – Binary metrics of the VGG19 GAP model for the Net Images dataset.

	Precision	**Recall**	**F1 Score**	**Support**
Stress	0.5714	0.8667	0.6887	60
Not stress	0.8367	0.5125	0.6357	80
Weighted Average	0.7230	0.6643	0.6584	140

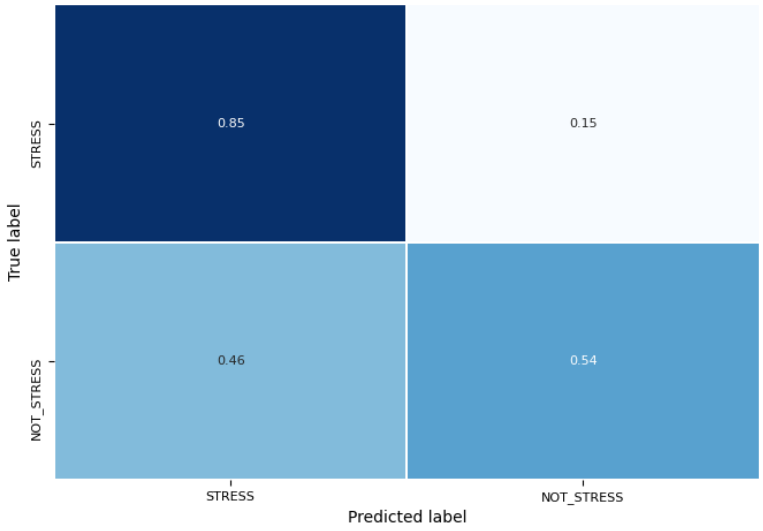

Figure 65 – Binary confusion matrix of the VGG19 GAP model for the CK+ dataset.

Table 46 – Binary metrics of the VGG19 GAP model for the CK+ dataset.

	Precision	**Recall**	**F1 Score**	**Support**
Stress	0.5692	0.8501	0.6819	387
Not stress	0.8338	0.5389	0.6547	540
Weighted Average	0.7233	0.6688	0.6660	927

Inception-ResNet V2 GAP binary evaluation

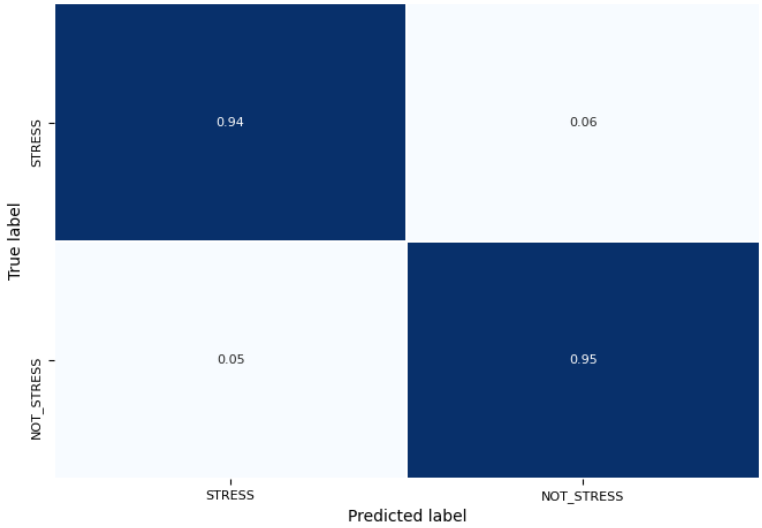

Figure 66 – Binary confusion matrix of the Inception-ResNet V2 GAP model for the KDEF test data.

Table 47 – Binary metrics of the Inception-ResNet V2 GAP model for the KDEF test data.

	Precision	**Recall**	**F1 Score**	**Support**
Stress	0.9297	0.9444	0.9370	126
Not stress	0.9578	0.9464	0.9521	168
Weighted Average	0.9458	0.9456	0.9456	294

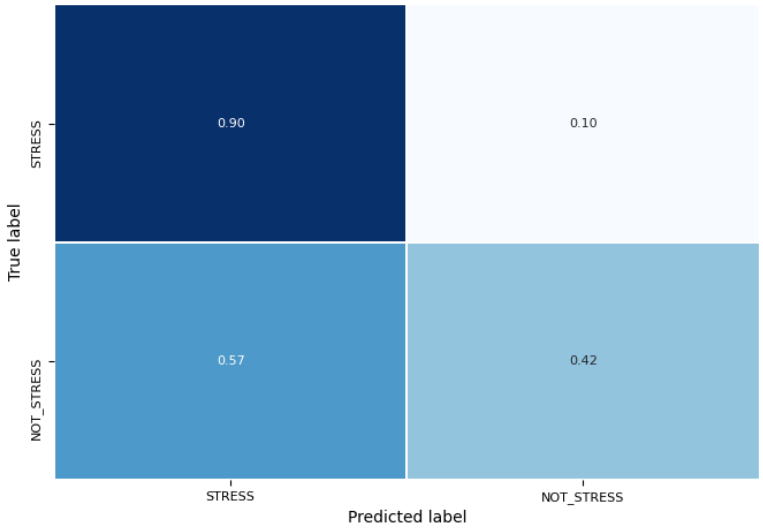

Figure 67 – Binary confusion matrix of the Inception-ResNet V2 GAP model for the Net Images dataset.

Table 48 – Binary metrics of the Inception-ResNet V2 GAP model for the Net Images dataset.

	Precision	**Recall**	**F1 Score**	**Support**
Stress	0.5400	0.9000	0.6750	60
Not stress	0.8500	0.4250	0.5667	80
Weighted Average	0.7171	0.6286	0.6131	140

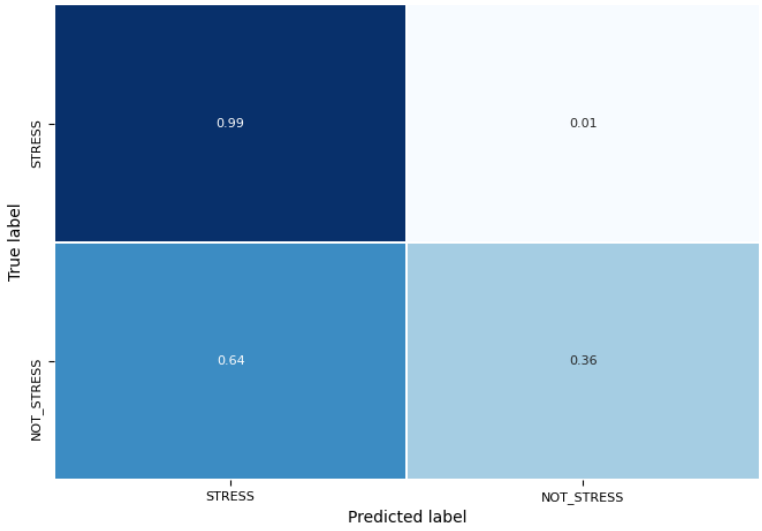

Figure 68 – Binary confusion matrix of the Inception-ResNet V2 GAP model for the CK+ dataset.

Table 49 – Binary metrics of the Inception-ResNet V2 GAP model for the CK+ dataset.

	Precision	**Recall**	**F1 Score**	**Support**
Stress	0.5233	0.9871	0.6840	387
Not stress	0.9747	0.3567	0.5223	540
Weighted Average	0.7865	0.6196	0.5897	927