UNIVERSITY OF TARTU

Institute of Technology

Robotics and Computer Engineering Curriculum

Andro Lominadze

# Real-Time Expression Analysis of Students in a Classroom Using Facial Emotion Recognition

Master's Thesis (30 ECTS)

Supervisor: Prof. Gholamreza Anbarjafari

Tartu 2020

**Real-Time Expression Analysis of Students in a Classroom Using Facial Emotion Recognition**

# Abstract

Life is getting more relied on computers. People create new machines and programs to make their lives easier. Devices are involved in a daily routine, so it might be useful if they were capable of understanding human's verbal or even emotional expressions. Nowadays, computers can learn almost anything and can help to analyze the surrounding world sometimes better than the human sense.

The following study can be used while doing a presentation or giving a speech in front of a big audience. It allows the user to be aware of the emotional condition of attending society. During the speech, it is almost impossible to observe every single face of the audience and guess how do they feel; computer vision techniques can do this job for humans. This framework consists of three main parts. In the first part, a pre-trained face detector model collects all the faces seen in the camera and assigns unique IDs. Each face is tracked during the whole video stream using and developing a Simple object tracking algorithm called Centroid Tracker. This tracker relies on a Euclidean distance measurement between the location of object centroid within the current and previous frame of video.

The second part of this thesis is Facial Expression Recognition (FER). For this part, Convolutional Neural Network (CNN) is trained over the FER2013 data set. The model is fed a set of face images taken from the previous step, successfully classifies seven different emotional states.

The third part stores the data of emotions for each person in such a way that it could be easily understandable for humans. The provided information contains the number of attending people, their facial expressions and overall mood in the audience. By this information, the user gets feedback about his/her speech. This feedback might help people improve presentation skills for the future or even change the presenting style immediately to increase the interest in the audience.

**Keywords:**

Facial Expression Recognition, Convolutional Neural Networks, FER2013

**CERCS: T111 Imaging, image processing**

## Klassiruumis õppivate õpilaste reaalajas ekspressioonianalüüs näo emotsioonide äratundmise abil

Meie aja elu sõltub arvutitest üha enam. Inimesed loovad oma elu lihtsustamiseks uusi masinaid ja programme. Seadmed osalevad meie igapäevases rutiinis, nii et kuigi me oleme nende masinate loojad, vajame neid, et mõista meie suulisi või isegi emotsionaalseid väljendeid. Tänapäeval saavad arvutid õppida peaaegu kõike ja aitavad meil ümbritsevat maailma mõnikord isegi paremini analüüsida, kui teeme seda inimlike meelte järgi. Järgnevat uuringut saab kasutada ettekande tegemisel või suure publiku ees kõne pidamisel. See võimaldab kasutajal olla teadlik ühiskonnas käimise emotsionaalsest olukorrast. Kõne ajal on peaaegu võimatu jälgida iga nägu publikus ja arvata, kuidas nad end tunnevad; arvuti nägemise tehnikad teevad selle töö meie eest. See raamistik koosneb kolmest põhiosast. Esimeses osas kogub eelkoolituse saanud näotuvastuse mudel kõik kaameras nähtud näod ja määrab ainulaadsed ID-d. Iga nägu jälgitakse kogu videovoo jooksul, kasutades ja arendades lihtsat objektide jälgimise algoritmi nimega Centroid Tracker. See jälgija tugineb Eukleidese vahekauguse mõõtmisele objekti keskpunkti asukoha vahel video praeguses ja eelmises kaadris.Lõputöö teine osa on näoilmetuvastus (FER). Selle jaoks koolitatakse FER2013 andmekogu kaudu konvolutsioonilist närvivõrku (CNN). Mudelile sisestatakse eelmisest etapist võetud näopiltide komplekt, see klassifitseerib edukalt seitse erinevat emotsionaalset olekut. Kolmas osa salvestab emotsioonide andmed iga inimese kohta viisil, mis oleks inimestele kergesti arusaadav. Esitatud teave sisaldab osalevate inimeste arvu, nende näoilmeid ja üldist meeleolu publikus. Selle teabe abil saab kasutaja tagasisidet oma kõne kohta. See tagasiside võib aidata edaspidiseks esinemisoskust parendada või isegi esitusstiili kohe muuta, et suurendada publiku huvi.

**Märksõnad:** Näoilmetu-vastus, konvolutsioonilist närvivõrku, FER2013

**CERCS: T111 Pilditehnika**

# Ackowledgement

I want to thank the iCV Research Lab for having me on their team, allowing me to work on such an exciting topic, and for enabling me to use all the necessary resources that were demanded by the project. A special thanks to my supervisor Prof. G. Anbarjafari (Shahab) for suggesting the topic, and for helping me throughout this period through his advice and feedback.

# Abbreviations

| | |
|---|---|
| FD | Feature Detecting |
| CNN | Convolutional Neural Networks |
| DL | Deep Learning |
| DNN | Deep Neural Network |
| FER | Facial Expression Recognition |
| LBP | Local Binary Patterns |
| LPQ | Local Phase Quantization |
| HOG | Histogram of Oriented Gradients |
| MHI | Motion History Images |
| HMI | Human Machin Interaction |
| AN, DI, FE, HA, NE, SA, SU | Anger, Disgust, Fear, Happyness, Neutral, Saddness, Surprised |
| CK | Cohn - Kanade |
| ReLU | Rectified Liner Unit |

# Contents

# 1 Introduction

Machines are getting involved in human lives significantly. People create new devices and let them do jobs instead of a human. As far it is going to make computers inseparable part of the daily life, it is more necessary them to have skills to understand a human. What do people want, say, or even feel. People use their biological sensors, such as eyes and ears, to perceive the surrounding world. Machines use cameras to see, and in some cases, those cameras are better than a human eye. Nowadays computers can learn some new things by themselves, understand and predict some events, make some analyses, give objective feedback. People use these opportunities to make life easier.

Depending on Vanessa Van Edwards' research [1], 67 percent of people think that they can spot others' emotions, but when it comes to standing in front of a big audience it becomes harder to get their emotional state. Machine, unlike a human eye, can get and analyze more data at the same time. Machine learning models can be trained so that they can distinguish facial emotions.

There are lots of positions that require good presenting skill, like a lecturer, sales manager or marketer. Not everybody has such skill, and they need to improve it. The main factor during self-development is feedback from the audience. While doing a presentation, it is not very easy to look at anyone's face and understand what do they feel or how do they like the audition.

The success of service robotics highly depends on a smooth Robot-User Interaction. Thus, a robot should be able to extract information just from the face of its user, e.g. must identify the emotional state. Human accuracy for classifying an image of a face in one of 7 different emotions is 65 percents [1]. One can observe the difficulty of this task by trying to classify the FER2013 dataset [2] images manually. Despite these problems, robot platforms give the possibility to cope with these struggles with computer techniques.

It is needed to deliver a modern solution for the problem stated above.

## 1.1 Contribution

In this thesis work, facial expression recognition model is proposed, which gives an opportunity to detect multiple faces and recognize facial emotions. In this study, it is shown that facial expression recognition needs some primary steps to be done. While working with faces, the most important thing is to detect and localize face as accurately as possible. On the next step it is handful to track the face; tracking is faster and gives many opportunities, such as remaining the given ID in case of occlusion. Firstly, a face detection model is created, which detects faces, registers them with unique IDs and tracks during the whole process. Afterwards, Convolutional Neural Network is fed by the images of faces for emotion classification. The algorithm attempts to classify the given face as portraying one of the seven basic emotions (Anger, Disgust, Fear, Happiness, Sadness, Surprise, Neutral).

The product is able to operate in front of a group of people. It can improve the user's presentation skill, give feedback about the general emotional state of the target audience. In that way, the speaker can see the feedback about every presentation, and solve some issues if there are any. The product must be user-friendly and easy to use, so retrieved information should be represented comfortable and easily understandable way.

## 1.2 Thesis Outline

In section 2, different approaches of Face detection, tracking and facial expression recognition are presented. Information about several different datasets created for emotion recognition is given in this section. Background knowledge needed for this task is given in section 3. In section 4, contribution in the three main parts: Face detection, emotion recognition and presentation of the results are described. Section 5 shows different results and discusses them. The conclusion and future prospects can be found in Section 6.

# 2 Related Work

In this section, different approaches of Face detection, tracking and facial expression recognition are given.

## 2.1 Face Detection

While working with faces, the first and primary step is to localize and extract face from the background. The human face is a dynamic object, and its appearance varies highly, that makes face detection difficult. In this subsection, different ways of face detection are represented [3].

In 1999 Mohamed Abdel-Mottaleb and Ahmed Elgammal developed an algorithm [4], that was working on coloured images and had robust efficiency in face detection, even if nearby was an object with similar colour as the skin.

This algorithm is based on four main stages. On the first stage, it detects pixels with skin color, to execute this step Y'UV color space is used. There is shown the data in three planes, that are YV, YU and UV.

However, there may be other objects with skin colour. In the second stage, surrounding pixels of previously detected pixels are checked. The algorithm determines if these pixels are part of the face, depending on the variance around those pixel. In the third stage, the resulting set of components are grouped in a graph with a vertex, representing each component and an edge between adjacent elements with similar colours. The graph is then recursively segmented into two groups of components, and each group is tested for the potential face. Groups that are not face-like are segmented. This grouping and recursive segmentation help the model to detect faces that were missed during the second stage.

Rowley, Henry A *et al.* developed a View-based Approach Using Neural Networks [5]. In this project face detection is based on the four main steps (1).

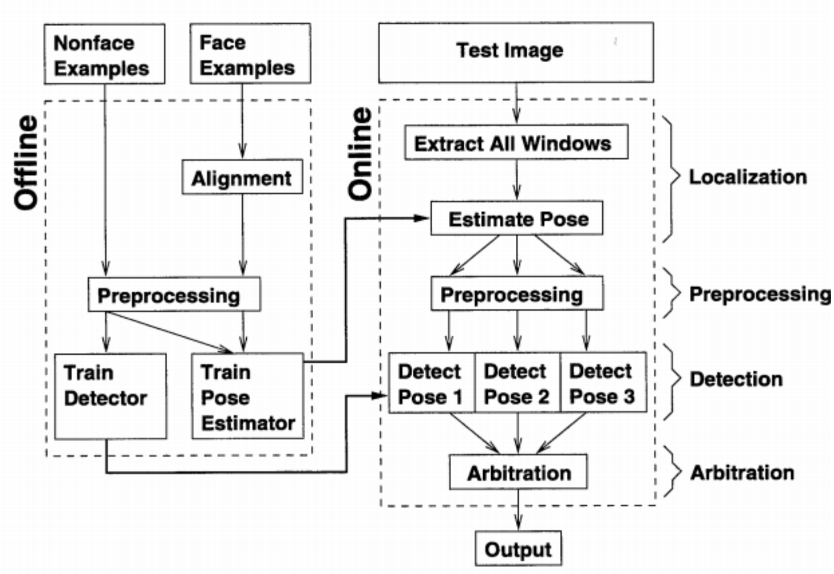


Figure 1. Schematic diagram of the main steps of the face detection algorithms [5]

**1. Localization and Pose Estimation:** Using artificial neural network requires many training examples. They are aligned with one another to reduce variability in the positive training images. The proposed neural network determines the pose of the face.

**2. Preprocessing:** the images are preprocessed with histogram equalization for further reduction of variation caused by lighting.

**3. Detection:** Potential faces that are already normalized in the first two steps are examined if they are faces or not. Separate networks are trained for frontal, partial profile and full profile faces.

**4. Arbitration:** In addition to use three detectors, multiple networks are also used within each pose. Each network learns different things from the data. Their results are then combined.

The paper "FACIAL FEATURE DETECTION USING HAAR CLASSIFIERS" [6] Phillip Ian Wilson and Dr John Fernandez *et al.* uses face detection method produced by Viola and Jones [7]. Haar cascade classifier is based on Haar-like features (2). These features do not look at the intensity of pixels. They detect contrast variances between groups of pixels to locate relatively dark or light regions. These groups of pixels can be adjusted in such a way that they can be able to find objects with different sizes. Integral image is used to calculate the simple rectangular features. These features are then trained with simple AdaBoost model.

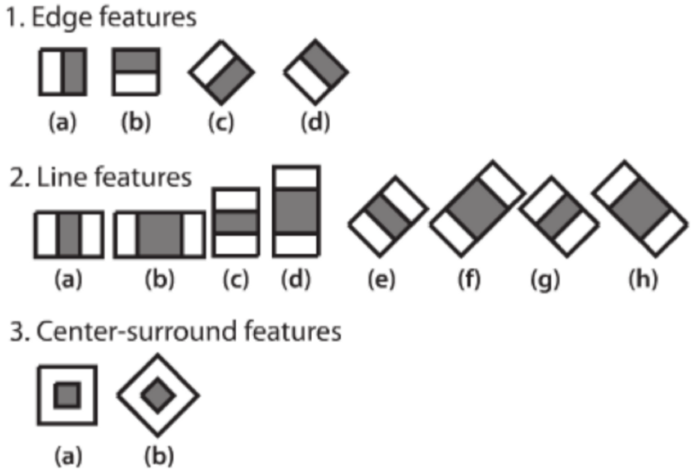


Figure 2. Haar Features [8]

The model has four main steps: Haar Feature Selection, Creating Integral Images, Adaboost Training and Cascading Classifiers.

A Haar-like feature [9] considers adjacent rectangular regions at a specific location in a detection window, sums up the pixel intensities in each region and calculates the difference between these sums. This difference is then used to categorize subsections of an image. among all faces, the region of the eyes is darker than the region of the cheeks. Therefore a common Haar feature for face detection is a set of two adjacent rectangles that lie above the eye and the cheek region. The position of these rectangles is defined relative to a detection window.

Integral Images are used to make the process super fast. The Integral Image or Summed Area Table is used as a quick and effective way of calculating the sum of pixel values in a given image, or a rectangular subset of a grid. When creating an Integral Image, Summed Area Table is created. In this table, for any given point (x,y), the pixel value becomes a sum of all the pixel values above, to the left and including the original pixel value of (x,y) itself (3).

Among all these calculated features, most of them are irrelevant. So selection of the best features out of more than 160000 is accomplished using a concept called Adaboost [10], which both selects the best features and trains the classifiers that use them. This algorithm constructs a “strong” classifier as a linear combination of weighted simple “weak” classifiers. There can be 162,336 Haar features, as used by the Viola-Jones object detection framework, in a $24\times24$ pixel image window, and evaluating every feature can reduce not only the speed of classifier training and execution but in fact reduce predictive power. The AdaBoost training process selects only those features known to improve the predictive power of the model, reducing dimensionality and potentially improving execution time.

Because each Haar feature is only a "weak classifier", a large number of Haar features are necessary to describe an object with sufficient accuracy and are therefore organized into cascade classifiers to form a strong classifier. The cascade classifier [11] consists of a collection of stages, where each stage is an ensemble of weak learners, and they are trained using a technique called boosting. Boosting provides the ability to train a highly accurate classifier by taking a weighted average of the decisions made by the weak learners.

## 2.2 Object Tracking Algorithms

The goal of object tracking is to find an object in the current frame of video stream. Since positions of the object have been detected, the parameters of the motion model is

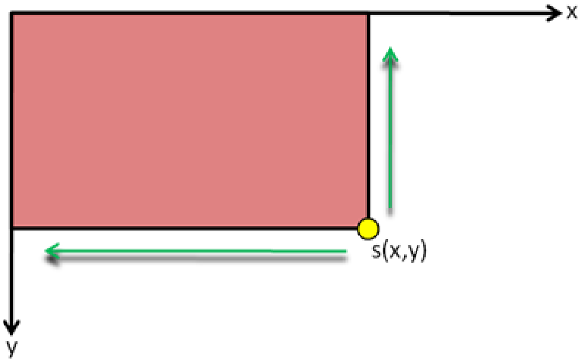


Figure 3. Integral Image Summation Equation: s(x,y) = i(x,y) + s(x-1,y) + s(x,y-1) + s(x-1,y-1) [12]

known. Having the motion model means knowing the location and the velocity ( speed + direction of motion ) of the object in previous frames. If there is no previous information about the object, the new location is still predictable based on the current motion model, and the prediction would be pretty close to where the new location of the object is.
There is more information than motion, such as appearance of the object in every frame. So an appearance model can be built. This appearance model can be used to search in a small neighbourhood of the location predicted by the motion model. Combining these two models will cause more accurate location prediction.
There is presented how different tracking algorithms approach this problem [13].

**BOOSTING Tracker:** This tracker is based on an online version of AdaBoost — the algorithm that the HAAR cascade based face detector uses internally. This classifier needs to be trained at runtime with positive and negative examples of the object. To start tracking the initial bounding box needs to be supplied by the user or by another object detection algorithm. This initial segment is taken as a positive example for the object. All the image patches outside the bounding box are treated as the background. In the new frame, the classifier runs on every pixel in the neighbourhood of the previous location, and the score of the classifier is recorded. The new location of the object is the one where the score is maximum. So now there is one more positive example for the classifier. As more frames come in, the classifier is updated with this additional data. Thus the classifier trains itself.

**MIL (Multiple Instance Learning) Tracker :** By idea, this tracker is similar to the BOOSTING tracker described above. The big difference is that instead of considering only the current location of the object as a positive example, it looks in a small neighbourhood around the current location to generate several potential positive examples. In MIL, positive and negative examples are not specified, but positive and negative "bags". The collection of images in the positive bag are not all positive examples. Instead, only one image in the positive bag needs to be a positive example.

**KCF (Kernelized Correlation Filters) Tracker:** This tracker builds on the ideas presented in the previous two trackers. This tracker utilizes the fact that multiple positive samples used in the MIL tracker have large overlapping regions. This overlapping data leads to some nice mathematical properties that is exploited by this tracker to make tracking faster and more accurate at the same time. Accuracy and speed, in this case, are both better than MIL and it reports tracking failure better than BOOSTING and MIL.

**TLD (Tracking, learning and detection) Tracker:** As the name suggests, this tracker decomposes the long term tracking task into three components — (short term) tracking, learning, and detection. From the author's paper [14], "The tracker follows the object from frame to frame. The detector localizes all appearances that have been observed so far and corrects the tracker if necessary. The learning estimates detector's errors and updates it to avoid these errors in the future." This tracker appears to track an object over a larger scale, motion, and occlusion. If the object is hidden behind another object, this tracker may be a good choice.

**MEDIANFLOW Tracker:** This tracker is Minimizing ForwardBackward error, so it enables to reliably detect tracking failures and select reliable trajectories in video sequences. This tracker works best when the motion is easily predictable and small. Unlike other trackers, that keep going even when the tracking has failed, this tracker knows when the tracking has failed. It has excellent tracking failure reporting. Works very well when there is no occlusion but fails under large motion.

## 2.3 Facial Expression Recognition

Current Human-Machine Interaction (HMI) systems have to meet the high requirements to be accepted in interaction with humans. Nowadays some works can provide HMI systems, which can recognize human's internal emotions.

In 1971 Ekman *et al.* identified 7 facial expressions (AN,DI,FE,HA,NE,SA,SU) as basic emotions. [15]. In 2006 Ali Mollahosseini, David Chan and Mohammad H. Mahoor *et al.* produced a paper "Going Deeper in Facial Expression Recognition using Deep Neural Networks" [16]. In a lot of neural networks, increasing the number of neurons or number of layers is used for the improvement, but people do not take into consideration what problems may cause increasing the depth. Adding an inception layer gives a possibility to remain reasonable depth and also get remarkable results. Another benefit of network-in-network method is that the global pooling performance is increased, which decreases the chance of overfitting. In this paper, there are used seven different datasets; the number of labelled images are given on the Table 1.

In 2016 Daniel Llatas Spiers *et al.* published a paper "Facial emotion detection using deep learning " [17]. This project is divided into two main stages. In the first step, facial emotion labelled data is used to train deep learning network. For this step Extended Cohn-Kanade [18] Database is chosen. Evaluations were made on different networks to determine the best accuracy. Google library "TensorFlow" [19] was used to perform all these training and implementation parts.

AN, DI, FE, HA, Ne, SA, SU stand for Anger, Disgust, Fear, Happiness, Neutral, Sadness, Surprised respectively. [16]

| | AN | DI | FE | HA | NE | SA | SU |
|---|---|---|---|---|---|---|---|
| MultiPie | 0 | 22696 | 0 | 47338 | 114305 | 0 | 19817 |
| MMI | 1959 | 1517 | 1313 | 2785 | 0 | 2169 | 1746 |
| CK+ | 45 | 59 | 25 | 69 | 0 | 28 | 83 |
| DISFA | 436 | 5326 | 4073 | 28404 | 48582 | 1024 | 1365 |
| FERA | 1681 | 0 | 1467 | 1882 | 0 | 2115 | 0 |
| SFEW | 104 | 81 | 90 | 112 | 98 | 92 | 86 |
| FER2013 | 4953 | 547 | 5121 | 8989 | 6198 | 6077 | 4002 |

Table 1

The second step is testing the model with a new data AM-FED [20]. This is done to make a comparison of both data sets. Focus is made on the main parameters: Network Loss, Learning Rate, Dropout and Optimizers

# 3 Background

## 3.1 OpenCV

OpenCV was built to provide a common infrastructure for computer vision applications and to accelerate the use of machine perception in commercial products. The library has more than 2500 optimized algorithms, which includes a comprehensive set of both classic and state-of-the-art computer vision, machine learning and Neural Network algorithms. It has C++, Python, Java and MATLAB interfaces and supports Windows, Linux, Android and Mac OS. OpenCV leans mostly towards real-time vision applications. [21]

## 3.2 Convolutional Neural Network (CNN)

Computers see in a different way than we do. They see every image as arrays of numbers. However, it is still possible to train them to learn new things. A specific type of Artificial Neural Network: a Convolutional Neural Network (CNN) is used to teach the algorithm, how to recognize objects in images. [22]
Convolutional Neural Networks have a different architecture than regular Neural Networks. First of all, the layers are organized in 3 dimensions: width, height and depth. Neurons in one layer do not connect to all the neurons in the next layer but only to a small region of it. Lastly, the final output are reduced to a single vector of probability scores, organized along the depth dimension. CNN has two components (4), which have been described below.

**The Hidden layers/Feature extraction part:** In this part, the network performs a series of convolutions and pooling operations. During this process, the features are detected. Convolution is one of the main building blocks of a CNN. Numerous convolutions are performed on the input, where each operation uses a different filter. All this results in different feature maps, in the end, are taken as the final output of the convolution layer.

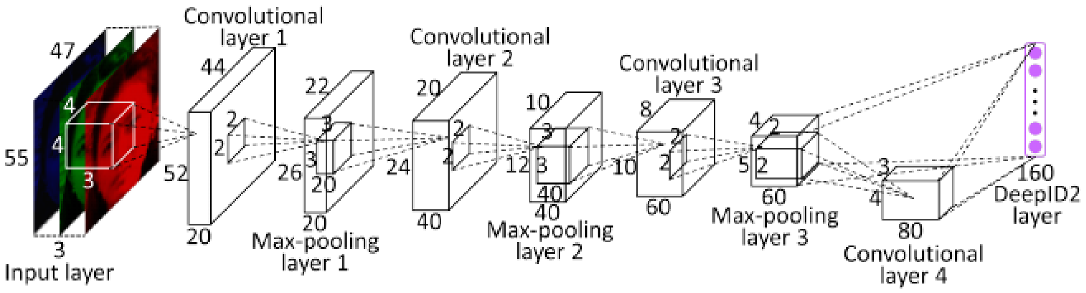


Figure 4. Feature Extraction and Classification in convolutional Neural Networks [22]

**The Classification part:** Here, the fully connected layers act as a classifier. They assign a probability for the object on the image being what the algorithm predicts it is. After the convolution and pooling layers, the classification part consists of a few fully connected layers. Neurons in a fully connected layer have full connections to all the activations in the previous layer.

## 3.3 Euclidean Distance:

In mathematics, the Euclidean distance or Euclidean metric is the "ordinary" straight-line distance between two points in Euclidean space. The Euclidean distance between points p and q is the length of the line segment connecting them ($\overline{\mathbf{pq}}$) [23].

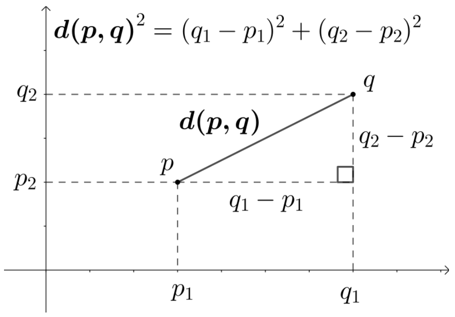


Figure 5. Visualisation of Euclidean Distance in two dimensions [23]

The proposed tracker searches for the minimum euclidean distance between the centroids in two subsequential frames to distinguish new centroids from already tracked centroids.

# 4 Methodology

Framework of the following study consists of three main parts. In the first part, the face detector model collects all the faces seen in the camera. Each face is tracked during the whole video stream using the object tracking algorithm called Centroid Tracker. The second part of this project is Facial Expression Recognition. The images from the previous step go to the model, which is trained on the face images with labels of seven different emotional states. The third part stores the data of emotions for each person and represents in an easily readable way.

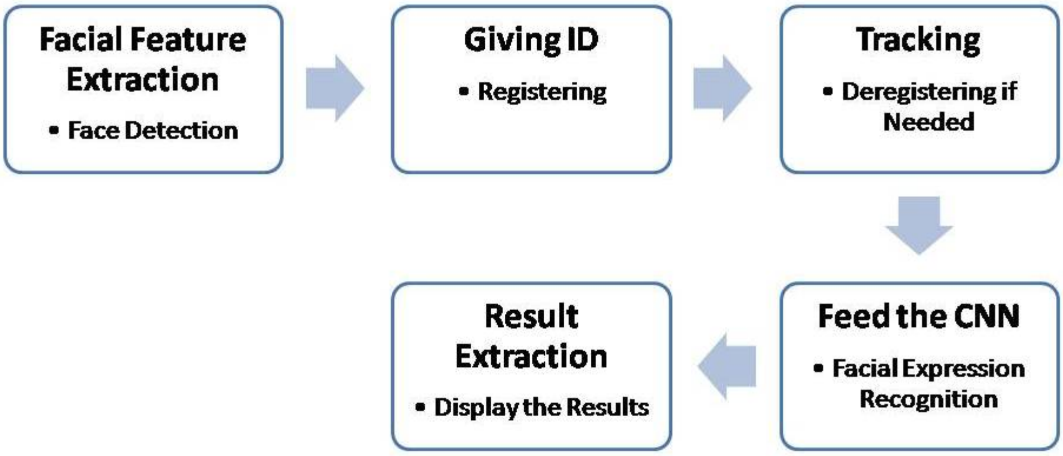


Figure 6. Workflow Visualisation

## 4.1 Face Detection and Tracking

**Preprocessing:** To obtain more accurate predictions from deep neural networks; the data needs to be preprocessed. In the context of deep learning and image classification, these preprocessing tasks typically involve: 1. Mean subtraction (Figure 7b) and 2. Scaling by some factor.

(a) Original Image

(b) Mean Subtracted Image

Figure 7. Mean Subtraction

**Mean Subtraction:** Mean subtraction helps to combat illumination changes in the input images. Typically the resulting value is a 3-tuple, consisting of the mean of the Red, Green, and Blue channels, respectively. In some cases the mean of Red, Green, and Blue values may be computed channel-wise rather than pixel-wise, resulting in the MxN matrix. In this case, the MxN matrix for each channel is then subtracted from the input image during training/testing. Both methods are valid, but the pixel-wise version is used in this study [24].

- R = R - $\mu R$
- G = G - $\mu G$
- B = B - $\mu B$

There also might be a scaling factor, $\sigma$, which adds in normalization, but it is set to $\sigma$ = 1 in this work. This procedure is executed by OpenCV Neural Network library "blobFromImage". Furthermore, it takes five main arguments:

**1. Image:** This is an input image, that is meant to preprocess before inserting into the neural network for classification.

**2. Scale Factor:** After the mean subtraction, images can be scaled; by default, the scale factor is one ($\sigma$ = 1).

**3. Size:** This is the spatial size that the Convolutional Neural Network expects.

**4. Mean :** These are the mean subtraction values. The supplied value is subtracted from every channel of the image.

**5. Swap R and B:** OpenCV assumes images are in BGR channel order; however, the mean value assumes RGB order is used. R and B channels can be swapped in the image to resolve this discrepancy.

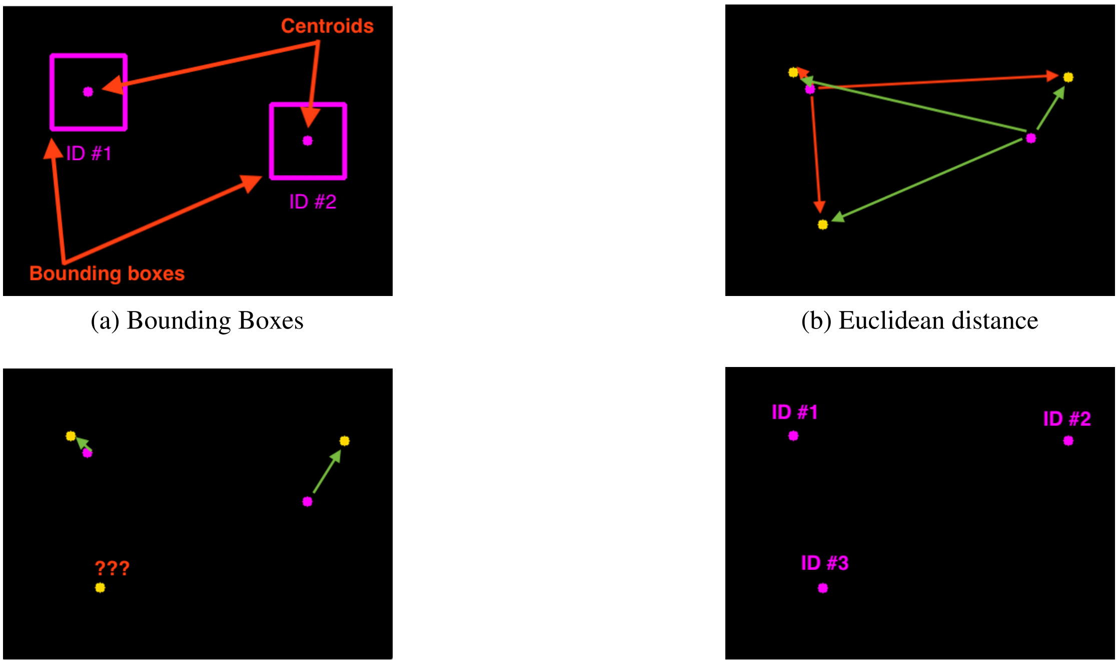
Preprocessed images are fed to the Neural Network, which returns the locations of faces. Finally, these faces are tracked.

**Face tracking:** Face tracking is a process of 1. Taking an initial set of object tracking (Input set of bounding box coordinates) 2. Creating a unique ID for each detection 3. Tracking these objects with maintaining the IDs. Face tracking allows counting the faces in every frame, give them IDs and apply the next, emotion recognition step. This method has the following abilities:

- Requires face detection only once
- Is extremely fast
- Is able to maintain the same ID when face will disappear or reappear
- Is robust to occlusion

Object tracking algorithm used in this thesis is called Centroid Tracking [25]. It relies on Euclidean distance measurement between object centroids, detected in two subsequential frames.

**Centroid Tracking Algorithm:** For building the object tracking algorithm, the first step is to accept bounding box coordinates from an object detector and use them to compute centroids. The example of this part is given in figure (8a).



(a) Bounding Boxes

(b) Euclidean distance

(c) Update (x, y)-coordinates of existing objects

(d) Register new objects

Figure 8. Defining the unique IDs

Ones there are bounding boxes, the next step is to find their centres - centroids. After finding the centroids, Euclidean distance is computed between new centroids and already existing centroids. Three objects are presented in the figure (8b). For simple face tracking with Python and OpenCV, computation of the Euclidean distances between each pair of initial centroids (red) and new centroids (green) are needed.

The second step avoids defining the unique IDs many times in every frame of video.

In the third step, Euclidean distances are used. As it is seen in the figure (8c), there are three objects detected, so proposed object tracking method associates centroids that minimize their respective Euclidean distances. This method relies on the assumption that objects move between frames, but the distance between the centroids for two sequential frames are much less than the distance to other objects. In the fourth step (Figure 8d), there is a new object with a new ID, so in that very frame where detected objects are more than existing IDs, there is a need for the new object to give a new ID, which is called "registering". The algorithm then goes to the second step and does the same sequence over again. Moreover, the final fifth step is deregistering old objects, that do not appear after some number of frames.

The main part of the tracker implementation is "update" method. The update method accepts a list of bounding box rectangles, presumably from a face detector. The format of the "rects" parameter is assumed to be a tuple with this structure: (startX, startY, endX, endY).

Since each object has its unique ID, if there is no detection for some given ID, the algorithm increments "disappeared" count of this specific object. It will also check if the "disappeared" count of any object has reached the maximum number of consecutive frames. If that is the case, there is a need to remove that specific object ID from the tracking systems.

To sum up, the Centroid Tracker works in the following manner:

- It Accepts bounding box coordinates for each object in every frame (presumably by some object detector).
- Computes the Euclidean distance between the centroids of the input bounding boxes and the centroids of existing objects that already have been examined.
- Updates the tracked object centroids to their new centroid locations based on the new centroid with the smallest Euclidean distance.
- And if necessary, marks objects as either "disappeared" or deregisters them completely.

## 4.2 Datasets and Models

**FER 2013:** In this project, FER2013 dataset is used for emotion recognition. This dataset was published in 2013 on the International Conference on Machine Learning (ICML). FER2013 is an opensource dataset; it was first created by Pierre-Luc Carrier and Aaron Courville *et al.* [26]. The dataset contains 35.887 grayscale, 48x48 sized face images (9). These images are labelled with seven different emotions.

- 4593 images- Angry
- 547 images- Disgust
- 5121 images- Fear
- 8989 images- Happy
- 6077 images- Sad
- 4002 images- Surprise
- 6198 images- Neutral

The dataset contains three main columns:

- **Emotion** column contains the numeric label of the facial expression.
- **Pixels** column contains the pixel values of the individual images. It represents the matrix of pixel values of the image.
- **Usage** column contains the purposes of each data sample. There are three different labels in this column: Training, PublicTest, and PrivateTest. For training purposes, there are 28,709 data samples. The public test set consist of 3,589 data samples, and the private test set consists of another 3,589 data samples.

for each emotion there is a numeric label that expresses the category of the emotion: 0 = Anger, 1 = Disgust, 2 = Fear, 3 = Happiness, 4 = Sadness, 5 = Surprise, 6 = Neutral. This dataset is chosen because as it is seen in the Table 1, it contains images with all the seven facial emotions and the fact that data is balanced, affects positively in creation of a proper CNN architecture.

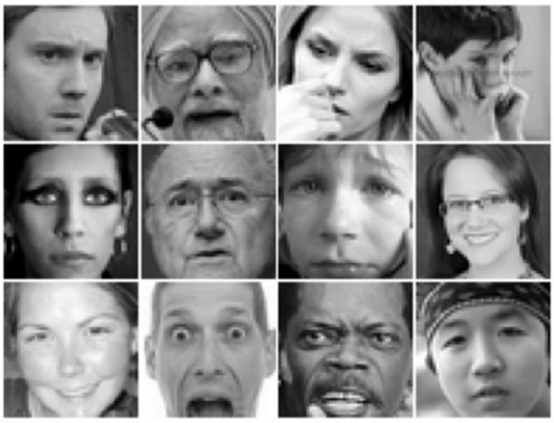


Figure 9. Samples of FER2013 Dataset [27]

**CAFFE (Convolutional Architecture for Fast Feature Embedding):** It supports many different types of deep learning architectures geared towards image classification and image segmentation. It supports CNN, RCNN, LSTM and fully connected neural network designs. Caffe supports GPU- and CPU-based acceleration computational kernel libraries such as NVIDIA cuDNN and Intel MKL [28].
The CAFFE model is trained to detect faces in frames, it can spot faces in almost every position. Reliable face detection is prerequisite of robust facial expression recognition.

## 4.3 Facial Expression Recognition

Facial expression recognition is an image recognition task for computer. Neural Networks process the input image to make emotion classification.

The CNN model proposed in this thesis is designed with the idea of creating the best accuracy over a number of parameters ratio while detectin facial expressions in a group of people.

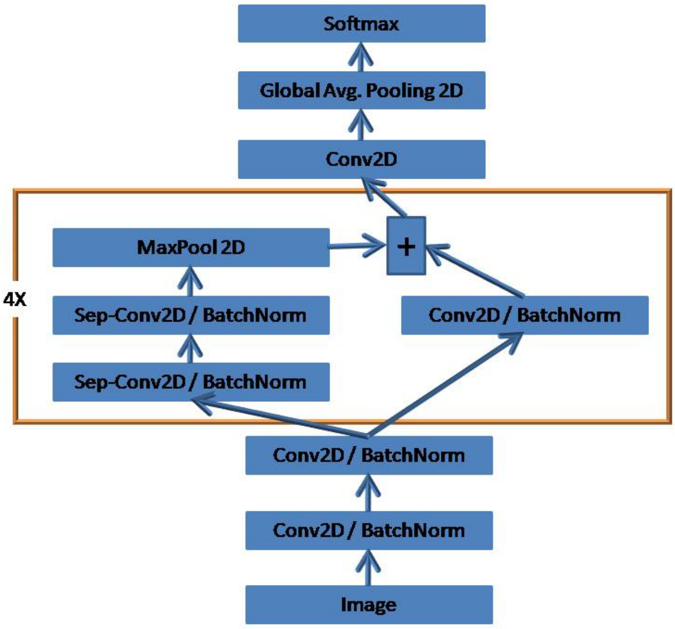


Figure 10. CNN Architecture [29]

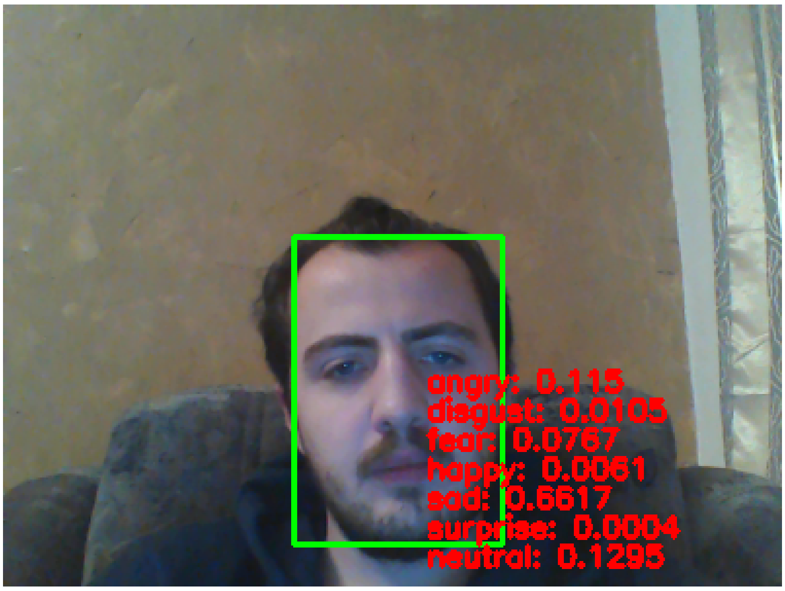
The proposed architecture uses Global Average Pooling to delete any fully connected layers. This is achieved by having the same number of feature maps as a number of classes in the last convolutional layer and applying a softmax activation function to each reduced feature map. The architecture is a standard fully-convolutional neural network composed of 9 convolutional layers; ReLUs, Batch Normalization and Global Average Pooling. This model contains approximately 600,000 parameters. It is trained on the FER2013 dataset, with batch size of 32 and number of epochs 10000.

Input images are faces that are detected in every frame of the video stream; they have different sizes, depending on the shape and position of the face. These images are reshaped to size 64x64 and converted to greyscale. Afterwards, the trained model predicts the probabilities of each emotion class for every face in the specific frame.

On the final step, all the data is processed and provided to the user. The final information contains the number of attending people, their facial expressions and overall mood in the audience. The used model is shown in figure (10).

## 4.4 Data Visualisation and Storing

The emotion recognition model returns probabilities for each of the seven emotional states: Anger, Disgust, Fear, Happiness, Sadness, Surprise and Neutral. All these probabilities are displayed on the screen for the user. This way, the user is always aware of the audience's emotional state (11).



```
angry: 0.115
disgust: 0.0105
fear: 0.0767
happy: 0.0061
sad: 0.6617
surprise: 0.0004
neutral: 0.1295
```

Figure 11. Display The Probabilities of Each Emotional State

Besides, all the data is stored to CSV file that includes nine columns. The first column consists the ID number. ID defines to whom belongs the following emotion state. Following seven columns are the probabilities for each emotion. The final, ninth column is a title of the emotion that has the highest probability. Information stored in the last column says which emotion is more likely to be shown on the specific face.

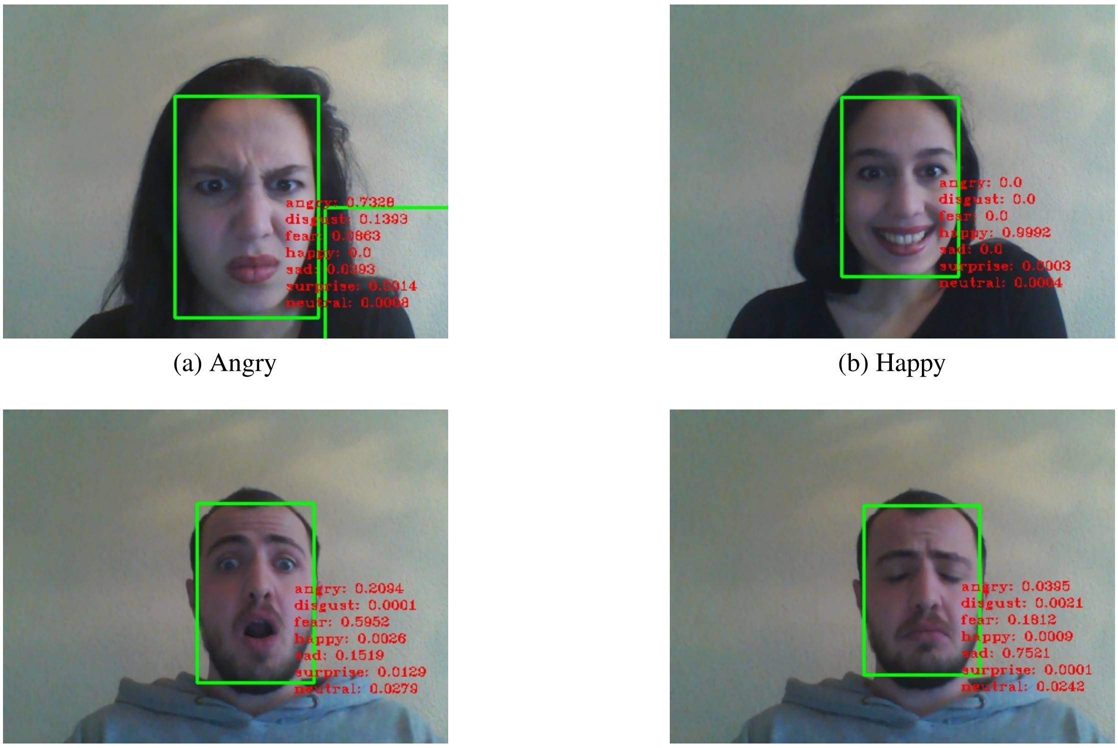
The data is recorded in every frame for every detected face. Since the raw data is very untidy, on the next step, the data is sorted by the IDs. All the data is sorted and saved in the new CSV file that still have the same amount of columns with the same order as in the previous case, but this time all the data is sorted in sequential order; user firstly will see all the emotional states of the attendance with ID=0, then with ID=1 and so on. The final step is to provide the user with short information in a convenient way. Sorted data is taken from the CSV file, and the columns that consist of emotional state probabilities are averaged column-wise. The final result shows the user average emotional state in the audience. The provided information contains the number of attending people, their facial expressions and overall mood in the audience. By this information, the user gets feedback about his/her speech.

# 5 Results

The following section explains the experimental design, performance measures used to evaluate classification methods, the efficiency of the used model with different datasets.

## 5.1 Performance

Results of the real-time emotion classification for different genders can be observed in figure (12). The example shows that the model predicted emotional state "angry" with 73 percents probability (12a), "happy" with 99.9 percents precision (12b), "fear" with 59 percents (12c) and "sad with 75 (12d).



(a) Angry

(b) Happy

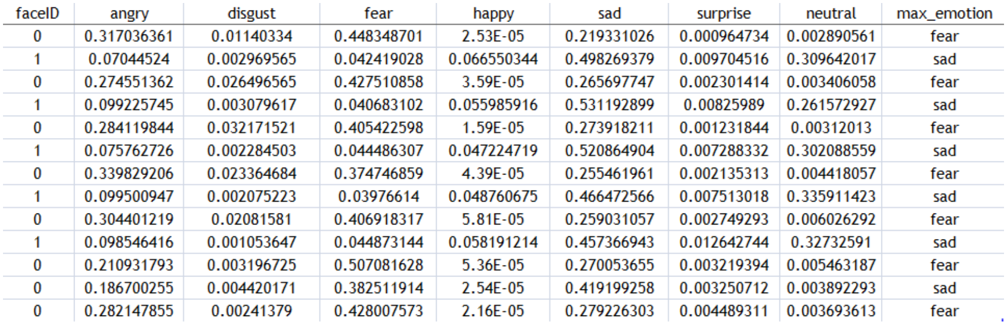
(c) Fear

(d) Sad

Figure 12. Different Emotional States for Different Genders

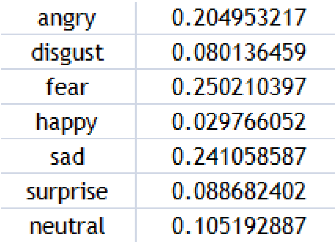
An example of the final state of the information given to the user can be seen in figure

(13). The final result gives the user the average emotional state of the audience (13b), and also the emotional state of each person from attending people (13a). during the testing it is possible to observe several common misclassifications such as predicting "sad" instead of "fear" and predicting "angry" instead "disgust". The CNN learned to get activated by considering features such as the frown, the teeth, the eyebrows and the widening of one’s eyes, and that each feature remains constant within the same class. These results reassure that the CNN learned to interpret understandable human-like features, that provide generalizable elements.



| faceID | angry | disgust | fear | happy | sad | surprise | neutral | max_emotion |
|---|---|---|---|---|---|---|---|---|
| 0 | 0.317036361 | 0.01140334 | 0.448348701 | 2.53E-05 | 0.219331026 | 0.000964734 | 0.002890561 | fear |
| 1 | 0.07044524 | 0.002969565 | 0.042419028 | 0.066550344 | 0.498269379 | 0.009704516 | 0.309642017 | sad |
| 0 | 0.274551362 | 0.026496565 | 0.427510858 | 3.59E-05 | 0.265697747 | 0.002301414 | 0.003406058 | fear |
| 1 | 0.099225745 | 0.003079617 | 0.040683102 | 0.055985916 | 0.531192899 | 0.00825989 | 0.261572927 | sad |
| 0 | 0.284119844 | 0.032171521 | 0.405422598 | 1.59E-05 | 0.273918211 | 0.001231844 | 0.00312013 | fear |
| 1 | 0.075762726 | 0.002284503 | 0.044486307 | 0.047224719 | 0.520864904 | 0.007288332 | 0.302088559 | sad |
| 0 | 0.339829206 | 0.023364684 | 0.374746859 | 4.39E-05 | 0.255461961 | 0.002135313 | 0.004418057 | fear |
| 1 | 0.099500947 | 0.002075223 | 0.03976614 | 0.048760675 | 0.466472566 | 0.007513018 | 0.335911423 | sad |
| 0 | 0.304401219 | 0.02081581 | 0.406918317 | 5.81E-05 | 0.259031057 | 0.002749293 | 0.006026292 | fear |
| 1 | 0.098546416 | 0.001053647 | 0.044873144 | 0.058191214 | 0.457366943 | 0.012642744 | 0.32732591 | sad |
| 0 | 0.210931793 | 0.003196725 | 0.507081628 | 5.36E-05 | 0.270053655 | 0.003219394 | 0.005463187 | fear |
| 0 | 0.186700255 | 0.004420171 | 0.382511914 | 2.54E-05 | 0.419199258 | 0.003250712 | 0.003892293 | sad |
| 0 | 0.282147855 | 0.00241379 | 0.428007573 | 2.16E-05 | 0.279226303 | 0.004489311 | 0.003693613 | fear |

(a) Raw Data



| | |
|---|---|
| angry | 0.204953217 |
| disgust | 0.080136459 |
| fear | 0.250210397 |
| happy | 0.029766052 |
| sad | 0.241058587 |
| surprise | 0.088682402 |
| neutral | 0.105192887 |

(b) Average Data

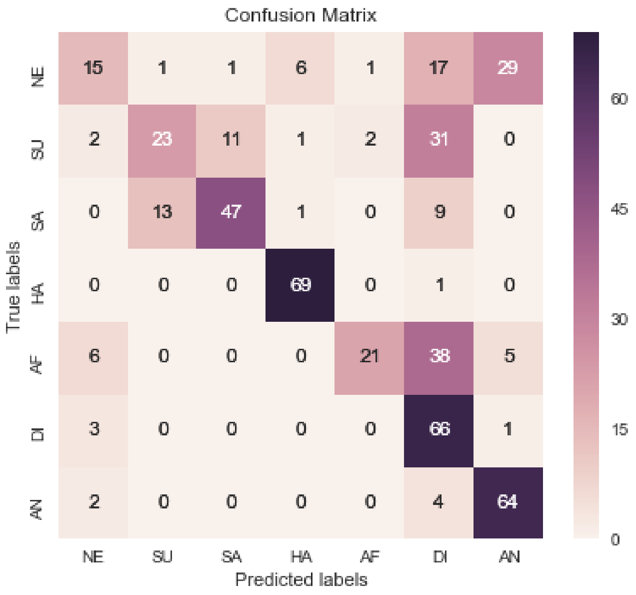
Figure 13. Final Data

## 5.2 Test the Model

**The Karolinska Directed Emotional Faces (KDEF)** [30] dataset is used to test the proposed model. The KDEF data is a set of totally 490 pictures of human facial expressions. The set of pictures contains 70 individuals displaying 7 emotional expressions.

Subjects were 70 amateur actors, 35 females and 35 males. Selection criterias were: age between 20 and 30 years of age, no beards, mustache, earrings or eyeglasses, and preferably no visible make-up during photo-session. Images from the KDEF database are frontal face images. They are coloured and size with 326 pixels by 326 pixels. Before testing the model, these images are converted to greyscale, resized to desired 64X64 shape, and normalised ( Equation 1). For this database the proposed model has 67.25 percents accuracy and the confusion matrix is shown in Figure (14).

$$(x = x/255.0 - 0.5) * 2 \tag{1}$$



| | NE | SU | SA | HA | AF | DI | AN |
|---|---|---|---|---|---|---|---|
| NE | 15 | 1 | 1 | 6 | 1 | 17 | 29 |
| SU | 2 | 23 | 11 | 1 | 2 | 31 | 0 |
| SA | 0 | 13 | 47 | 1 | 0 | 9 | 0 |
| HA | 0 | 0 | 0 | 69 | 0 | 1 | 0 |
| AF | 6 | 0 | 0 | 0 | 21 | 38 | 5 |
| DI | 3 | 0 | 0 | 0 | 0 | 66 | 1 |
| AN | 2 | 0 | 0 | 0 | 0 | 4 | 64 |

Figure 14. Confusion Matrix for KDEF Dataset

As it is seen in the Confusion Matrix on Figure (14), the model predicts mostly all the emotional states correctly. It has the best result with "happy" emotion, since as it is seen on the Table (1), the model had the biggest number of images labelled as "happy" to train. Besides, the model gives a good result in classifying disgusting and angry faces. On Table (2) are given the precision and recall results for the KDEF dataset. Precision

Classification Report for KDEF Database

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| AN | 0.54 | 0.21 | 0.31 | 70 |
| DI | 0.62 | 0.33 | 0.43 | 70 |
| AF | 0.80 | 0.67 | 0.73 | 70 |
| HA | 0.90 | 0.99 | 0.94 | 70 |
| SA | 0.88 | 0.30 | 0.45 | 70 |
| SU | 0.40 | 0.94 | 0.56 | 70 |
| NE | 0.65 | 0.91 | 0.76 | 70 |
| avg/total | 0.68 | 0.62 | 0.60 | 490 |

Table 2

is calculated as the sum of true positives across all classes divided by the sum of true positives and false positives across all classes. Recall is calculated as the sum of true positives across all classes divided by the sum of true positives and false negatives across all classes. The results suggest that the model classified happy, afraid, neutral, surprised faces with the highest F-score value.

**The Japanese Female Facial Expression (JAFFE) Database** [31] contains 213 images of 7 facial expressions posed by 10 Japanese female models. Each image has been rated on 6 emotion adjectives by 60 Japanese subjects. The database was planned and assembled by Michael Lyons, Miyuki Kamachi, and Jiro Gyoba [32].

Images from the JAFFE database are frontal face images. They are greyscale and size with 256 pixels by 256 pixels. Before testing the model, these images are preprocessed; Firstly, there are faces detected and cropped from original images to remove empty areas around the face to avoid misclassification. On the next step, images are resized to desired 64X64 shape and normalised (Equation 1). For this database, confusion matrix is shown in Figure (15). The confusion matrix shows that the proposed model classified

Classification Report for JAFFE Database

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| AN | 0.00 | 0.00 | 0.00 | 30 |
| DI | 0.00 | 0.00 | 0.00 | 29 |
| AF | 0.56 | 0.28 | 0.38 | 32 |
| HA | 0.88 | 0.68 | 0.76 | 31 |
| SA | 0.59 | 0.57 | 0.58 | 30 |
| SU | 0.24 | 0.94 | 0.38 | 31 |
| NE | 0.87 | 0.67 | 0.75 | 30 |
| avg/total | 0.45 | 0.45 | 0.41 | 213 |

Table 3

surprised and neutral faces most correctly. As the model is trained on FER2013 dataset which contains faces of subjects coming from Europe, the model had some difficulties to recognize emotions from people of other origins. On the Table (3) there is given the precision and recall for the JAFFE dataset. The F1-score is calculated by the equation (2) and it conveys the balance between the precision and the recall.

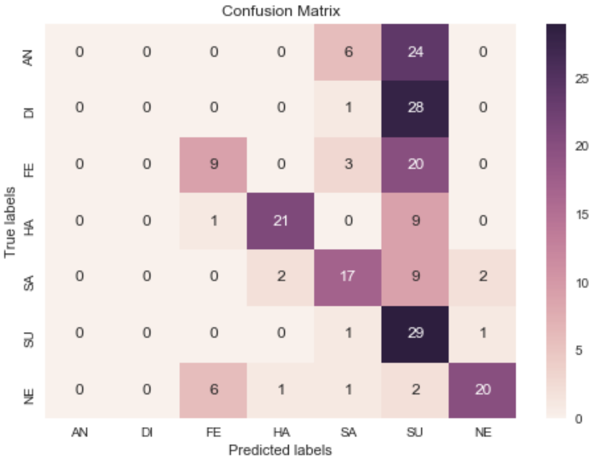


Figure 15. Confusion Matrix for JAFFE Dataset

$$F1 = 2 * Precision * Recall/(Precision + Recall) \tag{2}$$

# 6 Conclusion and Future Work

## 6.1 Conclusion

The product presented in this study is developed to help people improve their presentation skills. It can provide the user with feedback from the audience while he or she is doing an important presentation in front of a group of people. The speaker can focus on the speech. The developed vision system that performs face detection, face tracking and emotion classification, observes the overall emotional state of the attending people.
Multiple faces are detected in every frame of the video stream. Using the tracking model, which is based on the Euclidean Distance measurement, is used to give each face a unique ID and track them during the whole process of the presentation.
The proposed CNN architecture for facial expression classification has been systematically built to reduce the number of parameters. The model is trained on the FER2013 Dataset for 7 different facial expressions.
Testing of the final product shows that the model can be stacked for multi-class classifications while there are many people in the objective of the camera while maintaining real-time inferences. It helps the user to maintain positive and active presentation. The proposed model has achieved human-level performance in the classification tasks.

## 6.2 Future Work

For the future work, the model needs some improvements to predict more accurately. Facial appearance should not have a significant effect on the working efficiency. The emotional state can be converted to the new value, which represents how much does the attending people like the audience. Gender classification can be added to the model to observe how the speech affects different genders.

# References

[1] V. Van Edwards, *Captivate: The science of succeeding with people*. Penguin, 2017.

[2] P.-L. Carrier, A. Courville, I. J. Goodfellow, M. Mirza, and Y. Bengio, "Fer-2013 face database," *Universit de Montral*, 2013.

[3] E. Hjelmås and B. K. Low, "Face detection: A survey," *Computer vision and image understanding*, vol. 83, no. 3, pp. 236–274, 2001.

[4] M. Abdel-Mottaleb and A. Elgammal, "Face detection in complex environments from color images," in *Proceedings 1999 International Conference on Image Processing (Cat. 99CH36348)*, vol. 3. IEEE, 1999, pp. 622–626.

[5] H. A. Rowley, "Neural network-based face detection," CARNEGIE-MELLON UNIV PITTSBURGH PA DEPT OF COMPUTER SCIENCE, Tech. Rep., 1999.

[6] P. I. Wilson and J. Fernandez, "Facial feature detection using haar classifiers," *Journal of Computing Sciences in Colleges*, vol. 21, no. 4, pp. 127–133, 2006.

[7] P. Viola, M. Jones *et al.*, "Rapid object detection using a boosted cascade of simple features," *CVPR (1)*, vol. 1, pp. 511–518, 2001.

[8] "haarfeatures, howpublished = https://www.bogotobogo.com/python/opencv_python/python_opencv3_image_object_detection_face_detection_haar_cascade_classifiers.php, note = Accessed: 2020-13-05."

[9] "haar, howpublished = https://docs.opencv.org/3.3.0/d7/d8b/tutorial_py_face_detection.html, note = Accessed: 2019-22-01."

[10] "adaboost, howpublished = https://en.wikipedia.org/wiki/adaboost, note = Accessed: 2019-22-01."

[11] "cascading, howpublished = https://en.wikipedia.org/wiki/cascading_classifiers, note = Accessed: 2019-22-01."

[12] "integral image, howpublished = hhttps://computersciencesource.wordpress.com/2010/09/03/computer-vision-the-integral-image/, note = Accessed: 2019-22-01."

[13] S. Mallick, "caffe, howpublished = https://www.learnopencv.com/object-tracking-using-opencv-cpp-python/, note = Accessed: 2019-15-12."

[14] J. Shi, X. Wang, and H. Xiao, "Real-time pedestrian tracking and counting with tld," *Journal of Advanced Transportation*, vol. 2018, pp. 1–7, 10 2018.

[15] P. Ekman and W. V. Friesen, "Constants across cultures in the face and emotion." *Journal of personality and social psychology*, vol. 17, no. 2, p. 124, 1971.

[16] A. Mollahosseini, D. Chan, and M. H. Mahoor, "Going deeper in facial expression recognition using deep neural networks," in *2016 IEEE Winter conference on applications of computer vision (WACV)*. IEEE, 2016, pp. 1–10.

[17] D. L. Spiers, "Facial emotion detection using deep learning," 2016.

[18] T. Kanade, J. F. Cohn, and Y. Tian, "Comprehensive database for facial expression analysis," in *Proceedings Fourth IEEE International Conference on Automatic Face and Gesture Recognition (Cat. No. PR00580)*. IEEE, 2000, pp. 46–53.

[19] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin *et al.*, "Tensorflow: Large-scale machine learning on heterogeneous systems, 2015," *Software available from tensorflow. org*, vol. 1, no. 2, 2015.

[20] D. McDuff, R. Kaliouby, T. Senechal, M. Amr, J. Cohn, and R. Picard, "Affectiva-mit facial expression dataset (am-fed): Naturalistic and spontaneous facial expres-

sions collected," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2013, pp. 881–888.

[21] "openCV, howpublished = https://opencv.org/about/, note = Accessed: 2019-17-12."

[22] D. Cornelisse, "cnn, howpublished = https://www.freecodecamp.org/news/an-intuitive-guide-to-convolutional-neural-networks-260c2de0a050/, note = Accessed: 2019-17-12."

[23] "euclidean, howpublished = https://en.wikipedia.org/wiki/euclidean_distance, note = Accessed: 2019-17-12."

[24] A. Rosebrock, "blob, howpublished = https://www.pyimagesearch.com/2017/11/06/deep-learning-opencvs-blobfromimage-works/, note = Accessed: 2019-12-12."

[25] ——, "Object Tracking, howpublished = https://www.pyimagesearch.com/2018/07/23/simple-object-tracking-with-opencv/, note = Accessed: 2019-07-05."

[26] I. J. Goodfellow, D. Erhan, P. L. Carrier, A. Courville, M. Mirza, B. Hamner, W. Cukierski, Y. Tang, D. Thaler, D.-H. Lee *et al.*, "Challenges in representation learning: A report on three machine learning contests," in *International Conference on Neural Information Processing*. Springer, 2013, pp. 117–124.

[27] "Fer2013, howpublished = https://medium.com/@birdortyedi_23820/deep-learning-lab-episode-3-fer2013-c38f2e052280, note = Accessed: 2020-13-05."

[28] Wikipedia, "caffe, howpublished = https://en.wikipedia.org/wiki/caffe_(software), note = Accessed: 2019-12-12."

[29] O. Arriaga, M. Valdenegro-Toro, and P. Plöger, “Real-time convolutional neural networks for emotion and gender classification,” *arXiv preprint arXiv:1710.07557*, 2017.

[30] “KDEF, howpublished = https://www.kdef.se/, note = Accessed: 2020-15-05.”

[31] “JAFFE, howpublished = https://zenodo.org/record/3451524#.xr56suqzbiu, note = Accessed: 2020-15-05.”

[32] M. J. Lyons, S. Akamatsu, M. Kamachi, J. Gyoba, and J. Budynek, “The japanese female facial expression (jaffe) database,” in *Proceedings of third international conference on automatic face and gesture recognition*, 1998, pp. 14–16, accessed: 2020-12-05.

# Appendix

# Licence

## Non-exclusive licence to reproduce thesis and make thesis public

I, **Andro Lominadze**,

1. herewith grant the University of Tartu a free permit (non-exclusive licence) to:

   1.1 reproduce, for the purpose of preservation and making available to the public, including for addition to the DSpace digital archives until expiry of the term of validity of the copyright, and

   1.2 make available to the public via the web environment of the University of Tartu, including via the DSpace digital archives until expiry of the term of validity of the copyright,

   of my thesis

   **Real-Time Expression Analysis of Students in a Classroom Using Facial Emotion Recognition**

   supervised by Prof. Dr. Gholamreza Anbarjafari

2. I am aware of the fact that the author retains these rights.

3. I certify that granting the non-exclusive licence does not infringe the intellectual property rights or rights arising from the Personal Data Protection Act.

Andro Lominadze
Tartu, 13.05.2020