UMONS
Université de Mons

Faculté Polytechnique

Non Verbal expressions prediction

Master 1 - Electrical Engineering

GeorgesTSOLAKIS

POLYTECH
MONS

Under the supervision of M. El haddad KEVIN

2018-2019

pôle
hainuyer

Contents

Introduction

The goal of this study is to predict nonverbal expressions by implementing machine learning algorithms. By detecting the faces inside a video frame and then extracting the facial structure that maps the persons expression, we are gathering data, that later are going to be used to train our networks. The goal of these networks is to be trained upon different sets of points that correspond to different facial expressions.The desired output of will be an appropriate expression to the one used as input.During this study we will develop 3 kind of networks.Firstly a purely CNN network, second a combination of a VGG-16 pre-trained network with the initial CNN network and lastly a LSTM network.Each of these networks will be described in detail on the following pages.

Nonverbal communication represents a large part of communication.It can portray a message both vocally and with the correct body signals or gestures.As of now most of the interactive agents that respond to human input can mostly understand well-defined vocal commands.Taken under account the importance of nonverbal expressions in our everyday life,it is imperative that the development of algorithms that can produce this kind of expressions is being researched and implemented.

Chapter 1

Previous studies

1.1 Nonverbal Conversation Expressions prediction

1.1.1 Definition

Non verbal expressions are behaviour and elements of speech aside from the words themselves that transmit meaning or information, through visual, auditory, tactile, and kinesthetic channels.

A substantial portion of our communication is nonverbal.From they way we cross our arms to the unblinking eye gaze to indicate disapproval, every day we communicate with each other by using non verbal expressions.

The different non verbal means of communication can be grouped as shown below.

- Facial Expressions
- Gestures
- Paralinguistics
- Body language and Posture
- Proxemics
- Eye Gaze
- Haptics
- Appearance

Facial expressions A huge proportion of nonverbal communication is achieved through facial expressions.These can include smiles,laughs and frowns.Facial expression plays a crucial role in nonverbal communication,as it is often the first thing we see,even before we hear the things someone has to say.

Gestures Gestures also play an important role in communicating in a nonverbal manner.Gestures include waving,pointing and other deliberate movements.

Paralinguistics Paralinguistics is a vocal way of communicating by the variation of different factors such as voice,pitch and inflection.A sentence can have different meaning depending on the tone we are using.By using different variation of the factors in play, we can transfer a different message each time, even though we use the same sentence.

Body language and Posture Body language and posture can have a significant role in non verbal communication.Arms-crossing, defensive postures can easily make visible how comfortable a person feels.

Proxemics Proxemics or more commonly known as personal space, is the amount of space and distance between us and another person, we perceive as belonging to us.This factor is influenced by a number of factors including social norms, cultural expectations, situational factors, personality characteristics, and level of familiarity.

Eye Gaze Eye contact and even blinking can play an important role in communication.When people meet things or other people they like, they tend to blink more and their pupils dilate.Also steady eye contact is often take as a sign that a person tells the truth.

Haptics Communication through touch, often called haptics is another important nonverbal behaviour.Through touch feelings such as affection,sympathy and familiarity can be made visible.

Appearance Our choice of colour, clothing, hairstyles, and other factors affecting appearance are also considered a means of nonverbal communication. Research on colour psychology has demonstrated that different colours can evoke different moods. Appearance can also alter physiological reactions, judgements, and interpretations. Just think of all the subtle judgements you quickly make about someone based on his or her appearance. These first impressions are important, which is why experts suggest that job seekers dress appropriately for interviews with potential employers.

1.2 Previous Work

-cite previous recent papers about related works (NCE prediction) -NCE prediction -databases available

Chapter 2

Motivations and Contributions

Motivations Several previous research were found focusing on non-verbal communication.A lot of projects have been made on detecting emotions from images or live feed sources.During this project the emphasis was into predicting expressions in a day to day communication where there is a turn-taking relationship.The role of speaker and listener rapidly change.The focus was also on predicting non verbal expressions such as laughs and smiles.The databases that already existed were not annotated for the most part and those who were, did not meet the annotation criteria for this project.

Contributions For the reasons mentioned before a database was created. This database includes annotations of roles,speaker-listener and also annotations of expressions.This time consuming task was crucial in order to get the data that were later used to train a long short term memory network.

Chapter 3

Deep Learning-Based NCE Prediction System

3.1 Data collection

3.1.1 Databases

During this research different databases were used.For the training phase of the networks the IFA-Spoken-Language-Corpora and a database made by the TCTS lab were used.The databases have the same format.Two persons,each with a different role, speaker or listener are placed in front of a camera and they are being ask different questions.The goal is the capture their face expressions.

3.1.2 Annotations

Some of the databases that were used were already annotated,while other needed to by manually annotated. In order to achieve that, the Elan application was used with a template made for this task.This template includes a section for laugh, smile and also role.For each different section the user must manually specify the role of the person shown to the video i.e speaker or listener, and the time duration for which the speaker or listener keeps that role.After that, depending of the expression laugh or smile, the user must manually indicate the time duration of the expression and also the intensity.The intensity levels for laugh or smile range from high-medium-low.

3.1.3 Time it takes

The skill to manual annotate facial expression from different videos has a steep learning curve.In the begging the average time needed to annotate facial expressions is 60 minutes for 1 minute of video.The time to annotate roles is smaller and equal to amount of 25 minutes for one minute of video.As the time progresses and the user gets familiar with the process the time for expression annotation drops to 35 minutes per minute of video and the time needed for annotating roles to 11 minutes.It is worth mentioning that these times are strongly depended on the complexity of the video.A video where the role of listener and speaker change rapidly is more time consuming to annotate.In comparison the time to annotate expression stays more or less constant because even if we don't consciously conceive it, humans express themselves using facial expressions most of the time.

3.1.4 Quick stats

Average expression annotating time	Average role annotating time
1	4.97 minutes

3.2 Facial regions and data extraction

3.2.1 Data used for training

The data being used by the network are in a X,Y coordinates form.Sometimes when a network requires a 3 channel image, the Z coordinate is also being used.These points in form of coordinates, come from the extraction of 68 points.By implementing the dlib library and using as input videos from different databases we are able to detect if a face is present on the frame.If it's the case, we extract 68 points from each frame of the video.

3.2.2 Facial detection and extraction

The algorithm being used is able to detect positions of interests on faces, called landmarks.These landmarks are salient regions such as eyes, nose, mouth and eyebrows.Facial landmarks work in the same principle as shape prediction problems.Given an object of interest the algorithm tries to localise keys points along the same.For our case this method consists of two important steps.

- Detection of a face
- Localisation of the key points of interest on the face

In order to achieve the first step, a lot of methods can be used such as OpenCV's Haar cascades.Whats important is that at the end of the face detection we will have a bounding box, i.e the x,y coordinates of the face.

Given that first step succeeds, we can then proceed to localise the key facial structures in the face region.There is a variety of facial detection algorithms but most of them are working with the following facial regions.

- Jaw
- Nose
- Left eye
- Right eye
- Left eyebrow
- Right eyebrow
- Mouth

In our case we are using the pre-trained landmark detector included in the dlib library.This method starts by using a training set of labelled facial landmarks on an image. These images are manually labelled, specifying specific (x, y)-coordinates of regions surrounding each facial structure.In a second time the probability on distance between pairs of input pixels is being calculated.Given this training data, an ensemble of regression trees are trained to estimate the facial landmark positions directly from the pixel intensities themselves

The pre-trained detector is used to estimated 68 different points that map the facial structures of the face.

This facial landmark detector is an implementation of the One Millisecond Face Alignment with an Ensemble of Regression Trees paper by Kazemi and Sullivan (2014).

The spatial position of the 68 different points can be shown to the image below

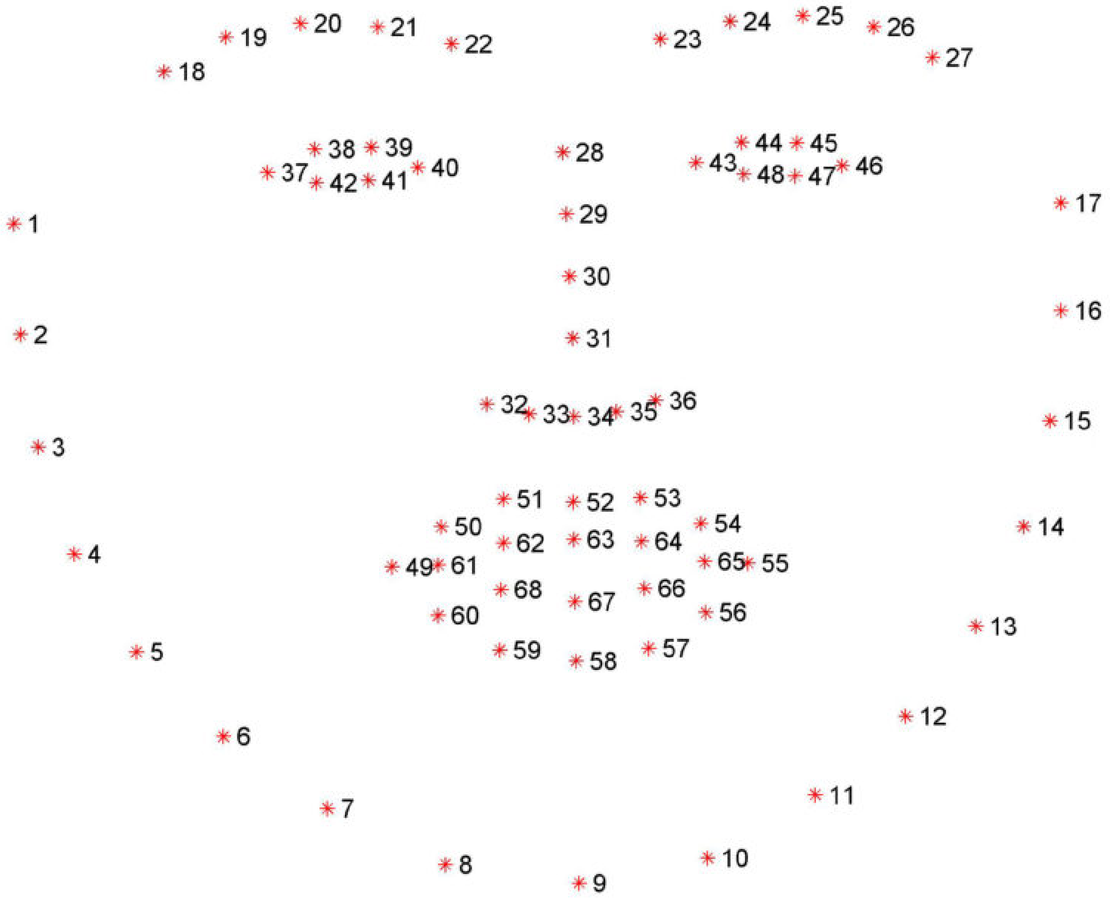


Figure 3.1: Spatial positions of 68 points

Chapter 4

Formation of a face using networks

On the images below we see the progressive formation of a face using different networks.

4.0.1 CNN

With every epoch that passes by, the network learns better and better to produce points that correspond to a face. On the following image, the state of the network after 3 epochs is shown.As it is clear, initially the network places 64 random points and as the epochs progress, the placement of these points becomes more precise.Taken under account that we have a simple CNN network, not many epochs are necessary in order to produce a face.

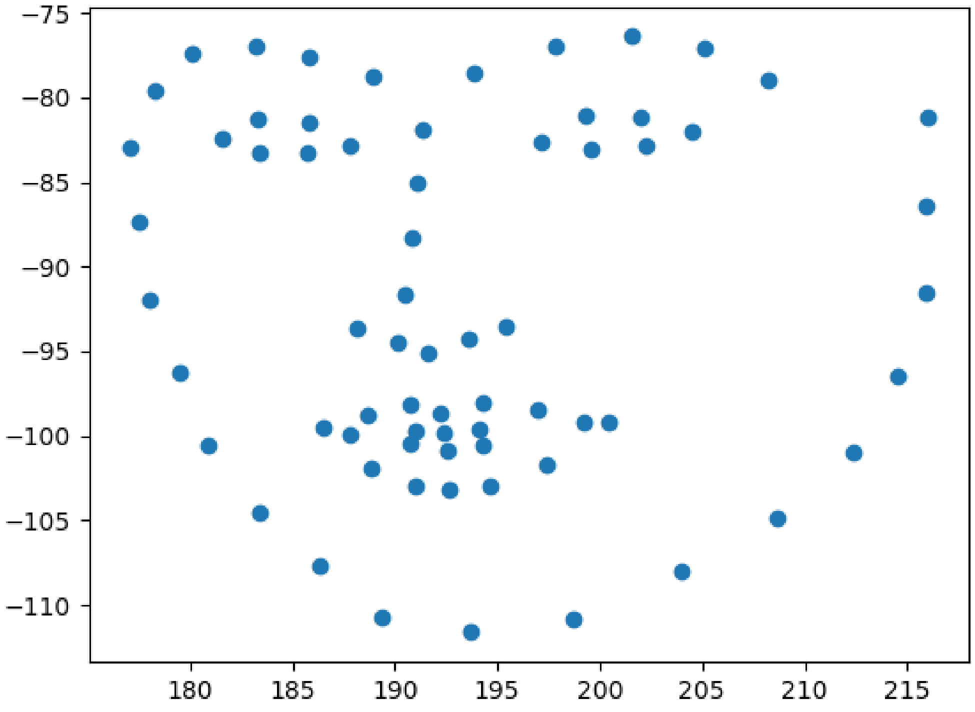


Figure 4.1: CNN after 3 epochs

As is it shown by the image below after 3 epochs the resulting image shows a faces but with some imperfection. These imperfection can be made more visible when we compare this image to an image produce by the CNN network after it was trained with 1000 epochs.

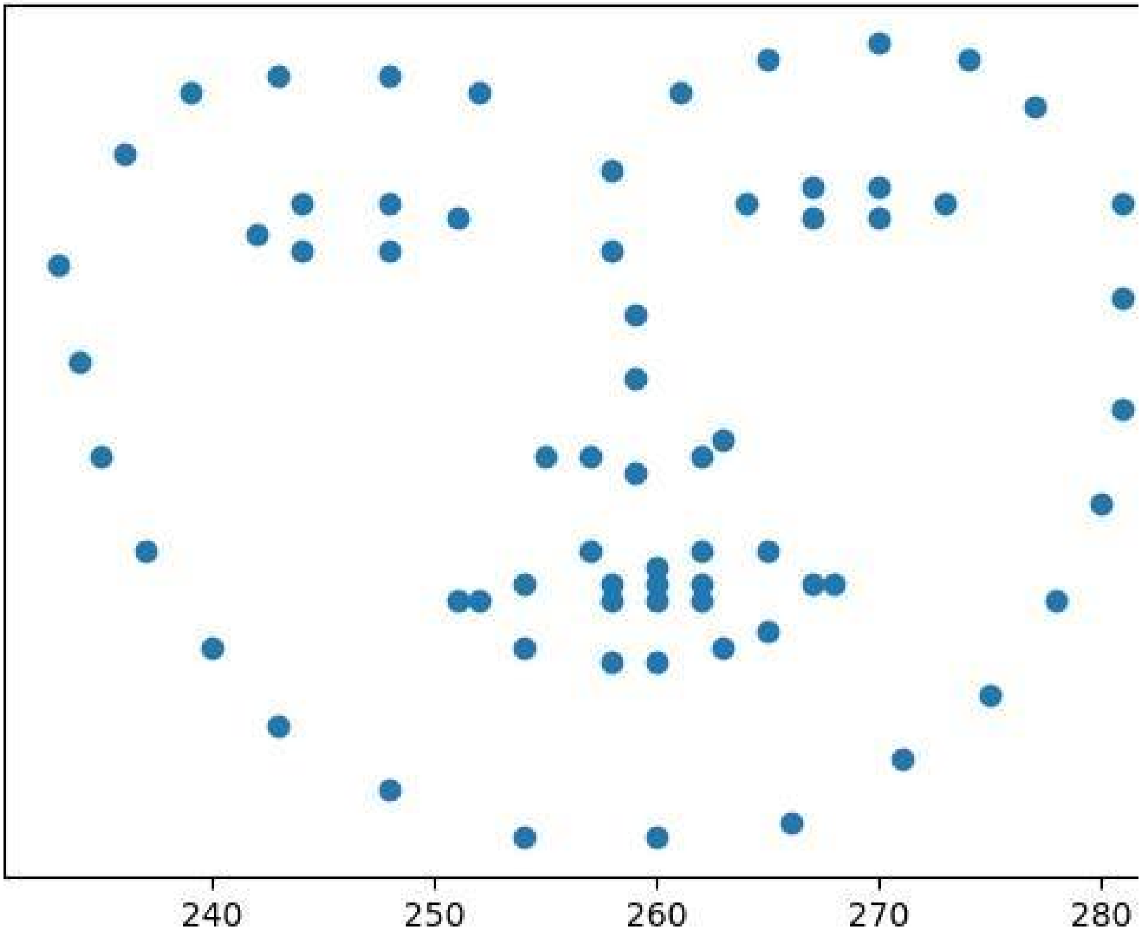


Figure 4.2: CNN after 1000 epochs

It clear that this face in comparison to the previous face,has its points more precisely place and with less imperfections.

4.0.2 VGG and CNN

The VGG and CNN is a more complicated network than the pure CNN network.It is then to be expected that more epochs are required in order to form a face.Below the picture of the output of the network after 3 epochs is shown.

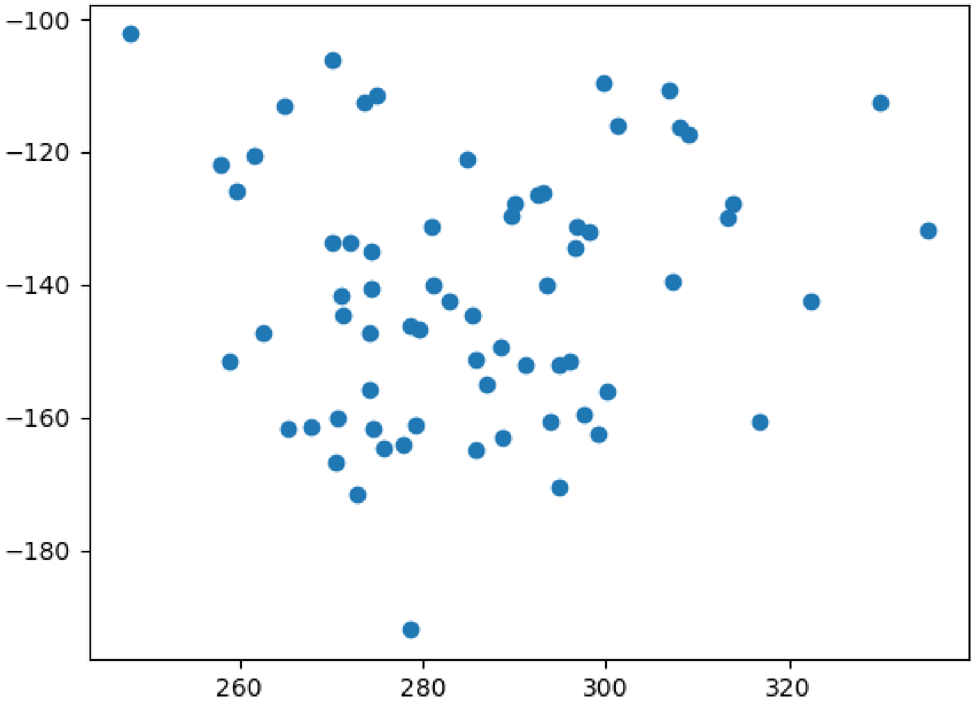


Figure 4.3: CNN and VGG after 3 epochs

As we see on the previous image,the placement of the points are 3 epochs is not precise and surely,it does not represent a face.This result is to be expected,VGG combined with a CNN network is a complicated network with millions of parameters.These parameters need a lot of time and data in order to be optimised for the task.

4.0.3 LSTM

The Long-short term memory network (LSTM) is a highly complicated network that requires a lot of time and data in order to produce reliable results.

4.1 System Overview

4.1.1 CNN

The first approach to the prediction of non verbal expressions was to created a simple convolutional neural network.In order to construct this network the following layers were added.

- A 2 dimensional convolutional layer
- A second 2 dimensional convolutional layer
- A 2 dimension pooling layer
- A dropout layer
- A flatten layer
- Another dense layer
- Another Dropout layer
- A final dense layer

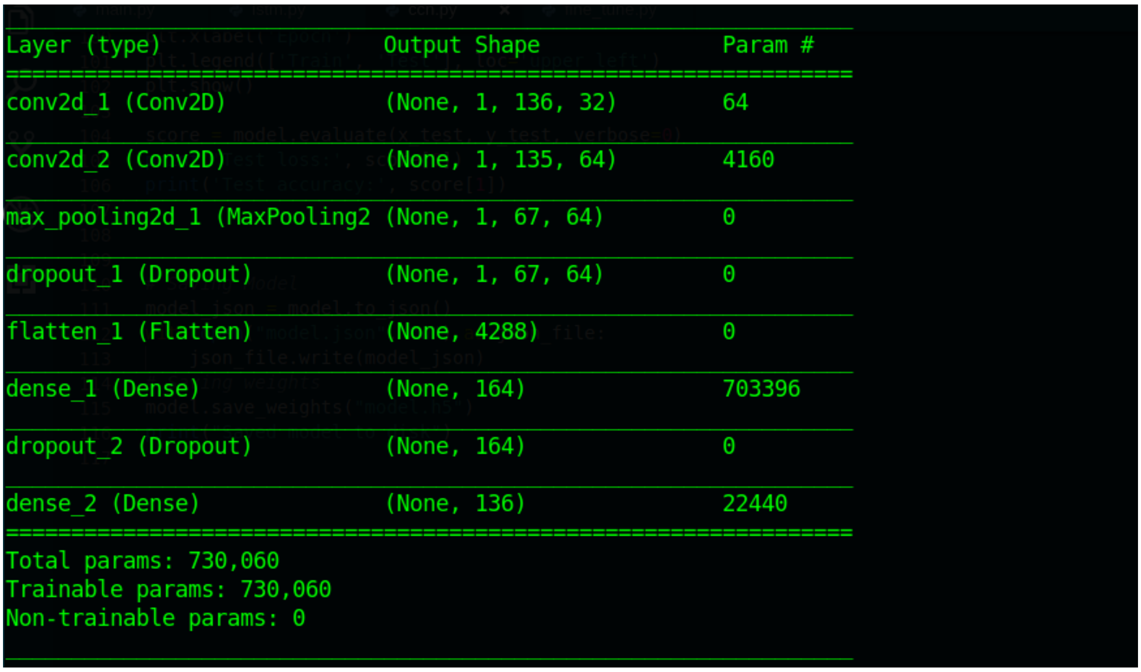
Two dimensional convolutional layers This layer creates a convolution kernel that is convolved with the layer input to produce a tensor of outputs. The first required parameters is the number of filters that the layer will learn.The layer that are closer to the input will learn fewer filters, this is why are second convolutional layer has a higher number of filters as a parameter.The activation function used is the Relu function and the kernel size was defined as (1,1).The kernel size is a 2-tuple specifying the width and height of the 2D convolution window.Taken under account that our network was using points and not images and that the dimensionality of the network was not that high, the choice of a (1,1) kernel was made.Finally for the first 2D convolutional layer the input shape must be specify.This input shape corresponds to the shape of the data we are feeding into the network.In our case it corresponds to (1,136,1).The first number corresponds to the number of lines.One line of data is equal to the x,y coordinates of the 68 points extracted with dlib.The second number corresponds to the number of columns containing a certain facial point , i.e the first column contains the x coordinate of the first facial point, the second column contains the y coordinate the first facial point.The last parameter of the input shape is equal to 1 because taken account that we are working with points, we only have one dimension.In order for this to be more clear two correspondences can be made.Firstly the correspondence with RGB and grayscale images.RGB images are called 3 channel images because they posses 3 vectors, one for each colour (R,G,B).In comparison grayscale images differentiate pixels by different intensities of gray,resulting to only only vector for each image.Another correspondence that can be made in order to understand better the choice of the third parameter of the input shape is the equivalence of (1,136,1) to a x,y,z coordinates system.If we thing the x coordinate as the axis that defines the horizontal plans, the y axis as the axis that defines the vertical plan and the z axis as the axis that defines the depth it is easily understandable that our input shape must be defined as (1,136,1)

Pooling layer This layer is implemented in order to reduce spatial dimensions.The depth dimension is excluded by this reduction.The gains of this layer is that by having less spatial information we gain computation performance, less spatial information also means less parameters, so less chance to over-fit.By defining different parameters we can control the level of reduction.

Dropout layer Dropout is a technique used to improve over-fit on neural networks.Other techniques that have the same goal are techniques like L2 regularisation.Inside the dropout layer we can define the percentage of neurons we wish to deactivate.The main idea of this technique is to generalise the learning.By deactivating certain neurons we reduce the risk of over-fitting, while we improve the generalisation by forcing different neurons to learn the same "concept".

Flatten layer The purpose of the flatten layer is to transform the pool sized feature maps into a column like shape in order to be used further into the network.

Dense layer A dense layer is a fully connected neural network layer, each input node is connected to each output node.The dense layer is the output of the network and this is why in our case is has a dimension of 136.



```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_1 (Conv2D)            (None, 1, 136, 32)        64
_________________________________________________________________
conv2d_2 (Conv2D)            (None, 1, 135, 64)        4160
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 1, 67, 64)         0
_________________________________________________________________
dropout_1 (Dropout)          (None, 1, 67, 64)         0
_________________________________________________________________
flatten_1 (Flatten)          (None, 4288)              0
_________________________________________________________________
dense_1 (Dense)              (None, 164)               703396
_________________________________________________________________
dropout_2 (Dropout)          (None, 164)               0
_________________________________________________________________
dense_2 (Dense)              (None, 136)               22440
=================================================================
Total params: 730,060
Trainable params: 730,060
Non-trainable params: 0
_________________________________________________________________
```

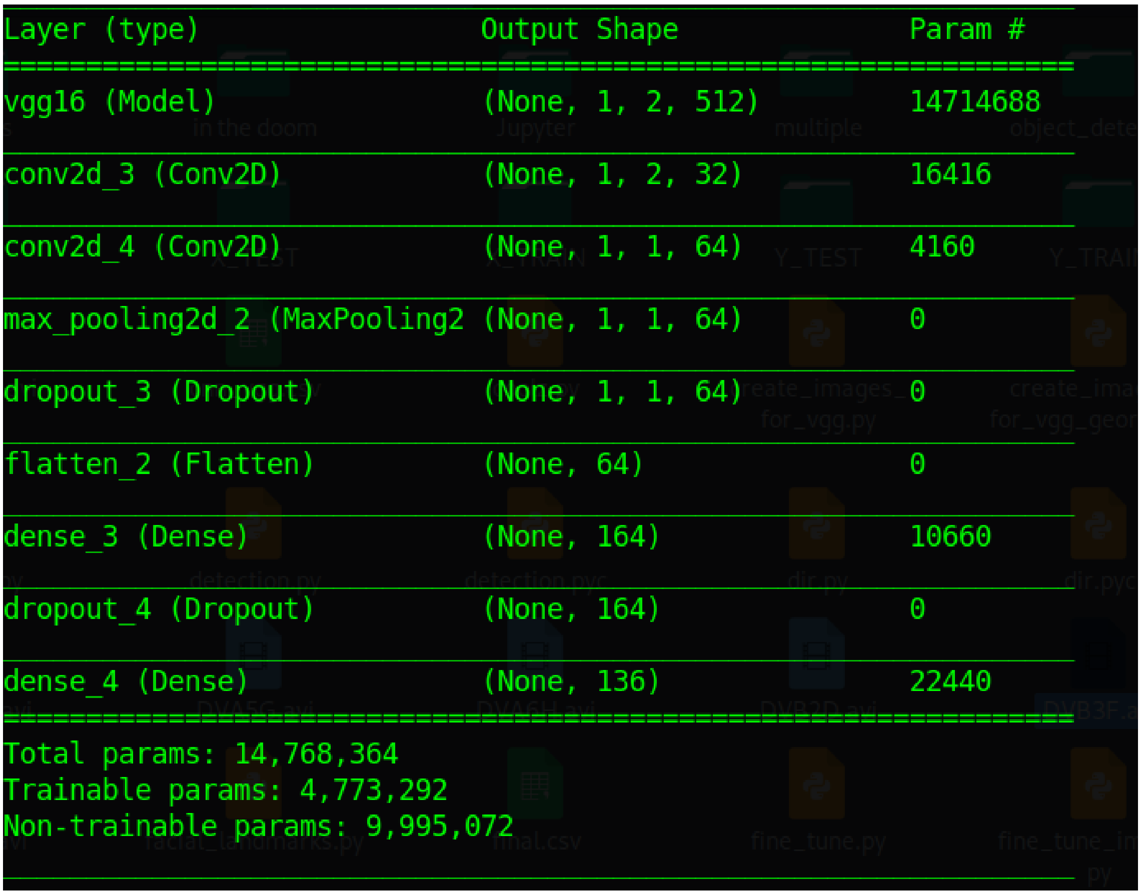
Figure 4.4: Convolutional Neural Network

4.1.2 Pre-trained VGG with CNN

VGG16 is a convolutional neural network model proposed by K. Simonyan and A. Zisserman from the University of Oxford in the paper Very Deep Convolutional Networks for Large-Scale Image Recognition.VGG16 uses as dataset ImageNet.

ImageNet is a dataset of over 15 million labeled high-resolution images belonging to roughly 22,000 categories. The images were collected from the web and labeled by human labelers using Amazons Mechanical Turk crowd-sourcing tool. Starting in 2010, as part of the Pascal Visual Object Challenge, an annual competition called the ImageNet Large-Scale Visual Recognition Challenge (ILSVRC) has been held.

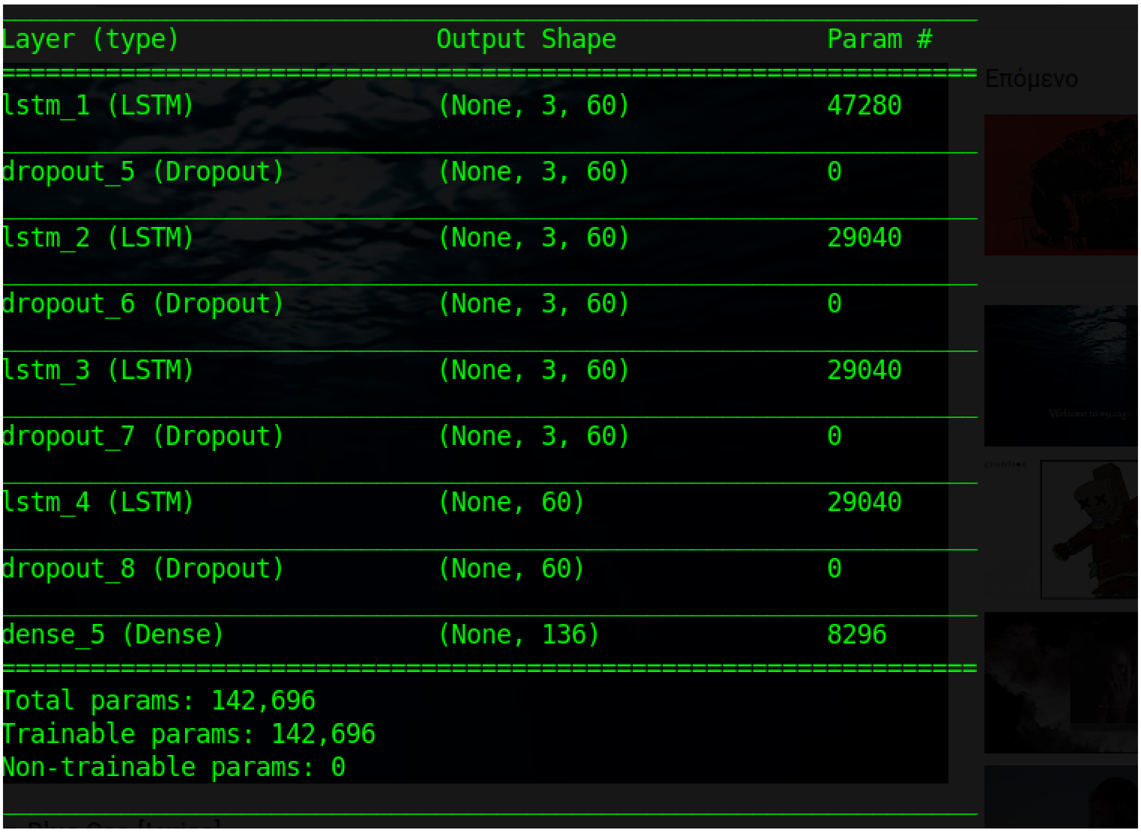
During this study,the layers contained in the VGG16 were used while being trained with the ImageNet dataset.Only the last 3 layer of this network were re-trained using our specific dataset.We then, connected our previous CNN network to the output of the VGG16 network.This is why the number of total parameters are so high, while the number of trainable parameters remains reasonable.



```
_________________________________________________________________
Layer (type)                 Output Shape              Param #   
=================================================================
vgg16 (Model)                (None, 1, 2, 512)         14714688  
_________________________________________________________________
conv2d_3 (Conv2D)            (None, 1, 2, 32)          16416     
_________________________________________________________________
conv2d_4 (Conv2D)            (None, 1, 1, 64)          4160      
_________________________________________________________________
max_pooling2d_2 (MaxPooling2 (None, 1, 1, 64)          0         
_________________________________________________________________
dropout_3 (Dropout)          (None, 1, 1, 64)          0         
_________________________________________________________________
flatten_2 (Flatten)          (None, 64)                0         
_________________________________________________________________
dense_3 (Dense)              (None, 164)               10660     
_________________________________________________________________
dropout_4 (Dropout)          (None, 164)               0         
_________________________________________________________________
dense_4 (Dense)              (None, 136)               22440     
=================================================================
Total params: 14,768,364
Trainable params: 4,773,292
Non-trainable params: 9,995,072
_________________________________________________________________
```

Figure 4.5: Pre-trained VGG with CNN

4.1.3 Long short-term memory



```
Layer (type)                 Output Shape              Param #
=================================================================
lstm_1 (LSTM)                (None, 3, 60)             47280
_________________________________________________________________
dropout_5 (Dropout)          (None, 3, 60)             0
_________________________________________________________________
lstm_2 (LSTM)                (None, 3, 60)             29040
_________________________________________________________________
dropout_6 (Dropout)          (None, 3, 60)             0
_________________________________________________________________
lstm_3 (LSTM)                (None, 3, 60)             29040
_________________________________________________________________
dropout_7 (Dropout)          (None, 3, 60)             0
_________________________________________________________________
lstm_4 (LSTM)                (None, 60)                29040
_________________________________________________________________
dropout_8 (Dropout)          (None, 60)                0
_________________________________________________________________
dense_5 (Dense)              (None, 136)               8296
=================================================================
Total params: 142,696
Trainable params: 142,696
Non-trainable params: 0
_________________________________________________________________
```

Figure 4.6: Long short-term memory

4.1.4 LSTM network trained with annotated data

Instead of training the network using extracted points from dlib, this time the Long short-term memory was trained using annotated data.This data were created by manually annotating videos with a template specific made for this project.The template includes role annotation,speaker-listener and two expressions laugh and smile,with their respective levels.Using the Elan application its was made possible that a list was extracted for every annotated video.This list includes the start time,end time and level of every expression annotated in the video.On top of that it includes the role for each person in the video and the time where that person held that role.

4.2 Training and Evaluation

-Explain how we train our network , parameters being used , inputs , outputs. -cite the results

Chapter 5

Discussion and Interpretations

-Discuss the results from the neural networks -Possible improvements that can be made

5.0.1 VGG-Pre-trained model

As cited before VGG-16 is a pre-trained model used for image qualification.Taken under account that a pre-trained model was used, a lot of restrictions had to be taken under account.Firstly, VGG-16 accepts as input images,or matrix, of a minimal dimension of 48x48x3.Our training data and also the data used for predictions had to be shaped to fill that requirement.Second,even though VGG is a highly skilled network,it's main use is prediction of images.During this project the input data were always sets of points extracted by the face-detection/extraction library,dlib.Taken under account the requirement of VGG-16 to work with images, two possible alternations of the coded network could be made.Firstly, from the extracted points from dlib, the points could be ploted using the script made for this purpose.This approach was not taken under consideration cause of the accumulation of error mainly from dlib's detection and also cause of the plot of points to create an image.Taken under account the important errors,these created images,if used by the network as input will give a highly erroneous prediction.A second approach would be to train the network based on the frames of the videos.Instead of extracting points,the whole frame can be extracted and used as input.This approach has the downside that it requires enormous amount of data.In our approach the output of each network was 136 points,68 pixels to create an image.If the output for the network is set to be images,the amount of data needed increases exponentially because the task becomes more complicated.This conclusion can also be made by the fact that the parameters to be trained and thus optimised for VGG are multiple times higher than the parameters for the purely CNN network.

5.0.2 Training data used

As said before, the training data used were face points extracted using the dlib library.One of the problems during this research was that humans don't change their expressions radically in a short period of time.In a minute minute video the expression changes were usually few and most of the times subtle.This led to the fact that our networks were trained on data that most of the time had a neutral expression.As a result the predictions included many neutral expressions.

5.0.3 Training based on specific expression

Challenge Taken under account the challenges during the training,mainly the static expressions on the video that led to neutral predictive expressions, the idea of training the network based on specific expressions was born.For this to work, a classification of the different databases was crucial to be made.

Applications considered For the extraction of facial landmarks,for the initial training of our networks the dlib library was used.Although dlib is quite successful on extracting points on 2-d axes (x - y coordinates),it fails to be as precise as other libraries, when the focus on a specific expression is needed.For this reason the implementation of other libraries such as open pose and open face were implemented.These libraries are far more precise when it comes to expressions that dlib and have the capability to extract points from 3 axes (x,y,z).The goal is the classify the databases in such a way that if we want to train our networks based on a certain expression i.e laugh we have the option to look our databases and we will know which videos and which time interval from each video we must give as input to the networks in order to be trained upon a certain expression.In that way the predictions made by these networks will not give static (neutral) responses.

Conclusion

-we did cool stuff (summarise them) -people should do these stuff to make it cooler:....(some suggestions) During this project a database manually annotated by following a certain template.The annotations include roles (speaker,listener) and facial expressions such as smiles, laughs and their intensity.By using this annotated data we trained a LSTM network based on facial expressions.Apart from that three other networks were created, a pure CNN , a VGG pre-trained network followed by a CNN and then a LSTM network. Lastly in order to train a network based on a specific expression we classify different databases using the open face and open pose library.