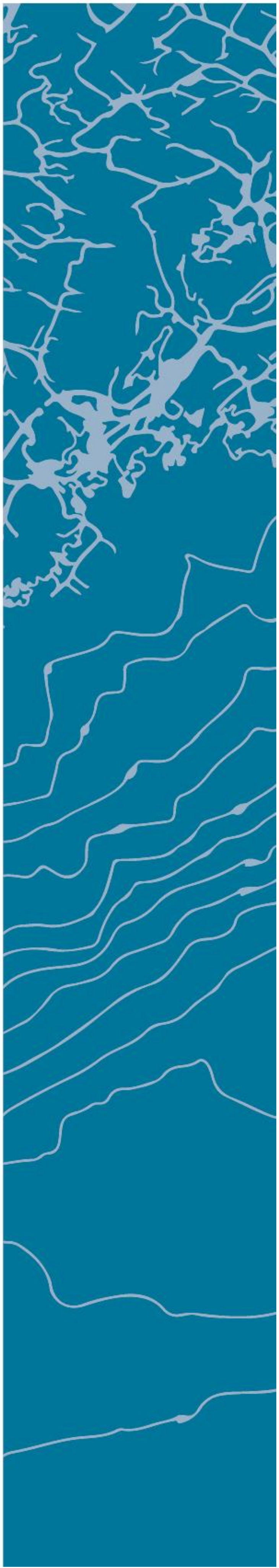
UNIVERSITETET I AGDER

Differentiating Between Spontaneous and Posed Facial Expression using Inception V4

KRISTOFFER MOBERG CHRISTENSEN

SUPERVISOR
Morten Goodwin

University of Agder, 2018
Faculty of Engineering and Science
Department of ICT



Abstract

This thesis proposes a way to simplify and make solutions for spontaneous and posed facial expression analysis more efficient. Traditional approaches have been using hand-crafted features and two image frames to be able to differentiate between spontaneous and posed facial expressions. The solution aims to be as flexible as possible and introduces two models to differentiate between posed and spontaneous facial expression.

We introduce Inception V4 as an algorithm to solve this task. The results indicate that Inception V4 may be too deep and unable to differentiate between spontaneous and posed facial expression accurately. A shallow CNN model is also introduced. The shallow CNN model performs better than the Inception V4 model. None of the two come close to the state-of-the-art results. This may indicate that to differentiate between spontaneous and posed facial expressions the difference between the onset and apex frame of an expression is needed as input. This thesis, also suggests an alternative algorithm based on our findings. For further work, an algorithm which is not as deep as Inception V4 is needed. However, by using parts of the Inception V4 architecture, we may be able to capture facial features better.

The task of differentiating between spontaneous emotion and posed emotion has also been investigated; however, the results do not show great promise. The task does not have any state-of-the-art results to compare our approach with. Our models, although lacking in performance, does seem able to capture relevant facial features from the dataset.

Preface

This thesis is made as a completion of the master eduction in Information Technology and Communication (ICT), at the University of Agder, Norway.

With this preface I would like to thank my supervisor, associate professor Morten Goodwin for invaluable help to complete this thesis. He has time after time provided valuable insights into problems.

I would also like to thank my fellow master student, smund Kamphaug, for valuable ideas and discussions regarding my implementation of Inception V4. He has provided valuable insights into why my model does not work and provided me with several ideas for how to fix it.

Arendal, 4 June 2018.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Emotion classification through facial expressions has received much attention. However, differentiating between spontaneous and posed facial expression has not been investigated a lot in the literature. Research in this area has shown great promise, but most of them only differentiate between posed and facial expressions. It would be more interesting to compare these in more detail, for instance, one could have output as spontaneous angry, posed angry, spontaneous sad, etc.

Differentiating between posed and spontaneous emotions is difficult because the differences between these two may be subtle and hard to capture. Humans often use posed facial expressions to try to disguise their real emotions. Also, humans convey emotion differently, and some people can disguise their 'true emotion' through a posed expression better than others. Hence if a person is skilled at disguising their emotion as spontaneous (genuine), it is difficult to detect whether it is spontaneous or posed. However, even though a person may be skilled at disguising their emotion as spontaneous, there exist micro expressions of a facial expression which is much harder to disguise.[15]

This thesis proposes a method for differentiating between posed and spontaneous facial expression of emotion by using a pre-trained Inception V4 network.

Most of the existing research propose ways to detect spatial and temporal patterns through hand-crafted features and comparison from the start of an expression until full expression. These methods do perform well but do not scale well for real-world applications due to the overhead of preprocessing. This thesis proposes a method to feed images directly into a model without doing any feature extraction. The idea behind feeding images directly into the model is that a deep network, such as Inception V4, should be able to capture these features by itself.

1.1 Motivation

The area of computer vision can hugely impact the society in many ways. Computer vision is a field within machine learning that comprises of processing and understanding visual information. Differentiation between posed and spontaneous facial expressions is a particular kind of computer vision. To be able to differentiate between these two, a machine learning algorithm needs to be able to capture how a person, face, expression and eventually spontaneous and posed expression looks like. A model capable of differentiating between these has many applications, one such is human-computer interaction. It could substantially increase the experience of human-computer interaction by that the computer could be able to understand better how we feel and therefore be able to show content that either tries to negate that feeling or reinforce it. Another possible application is the detection of deception which could be useful in various scenarios. Police investigators could use a model to determine whether a person is hiding something.

The area of differentiating between spontaneous and posed emotion is vital to research more, due to its applications. One application is to use a model to evaluate the trustfulness of a person. It also gives an estimate whether a person is trying to deceive other people by using a posed expression. Being able to tell if a person is deceiving could be beneficial in police interrogation, witness testimony and possibly more. Another application could be using this in medicine, where a doctor can evaluate whether a person's pain is posed(fake) or spontaneous(real). It could give doctors an indication whether the patient is lying about the pain or not and would save doctors a lot of time by not having to use time on a patient who is lying about their condition.

Prior research has for the most part only considered spatial patterns to differentiate between spontaneous and posed emotion. Spatial pattern means the movement of facial feature points and the pattern which can be deduced from this. The prior research has utilized two frames from a video sequence to measure the displacement between facial feature points in onset and apex frame. Apex means when the emotion is at its fullest, while onset is the start of the emotion sequence. Former research approaches would need additional methods to capture both the onset and apex frames from a camera or video. This would not scale well for real-world applications. A better approach is only to use one image, namely the apex frame to differentiate between spontaneous and posed facial expressions. That approach would be more scalable to real-world applications by reducing the overhead of preprocessing.

1.2 Goal

1. Explore the possibility that a pre-trained model of InceptionV4 performs better than a shallower CNN network for the task of classifying spontaneous and posed facial expressions.
2. Explore the possibility that an Inception V4 model can capture the difference between 12 different emotions(Spontaneous angry, posed angry, Spontaneous happy, etc.....)
3. Determine if it is possible to skip the step of feature extraction and feed the facial expression image directly into a neural network model.

1.3 Statement of problem

Differentiating between spontaneous and posed facial expressions is a classification problem. To be able to tackle this problem, a method that can capture both spatial and temporal features is required. Spatial features refer to the movement of facial muscles, where they usually are labeled with facial action units(AUs). Temporal features refer to duration, amplitude, speed, acceleration, symmetry, and trajectory.

Most of the previous work done has focused on hand-crafted features. Gan et al. [6], Wang et al. [21], and Xu et al.[2] used deep learning to tackle this problem and assumed that their models were able to capture the spatial and temporal features by itself. All of them, however, used both apex and onset images to differentiate between spontaneous and posed facial expressions. Our thesis is going to investigate whether it is possible to omit the onset image and only use the apex facial expression image. It would require far less preprocessing to classify spontaneous and posed facial expressions if it is possible for a classifier to perform well when omitting the onset image. The approach of Gan et al. [6], Wang et al. [21], and Xu et al. [2] all require preprocessing to identify the onset and apex facial expression images, which makes them less computationally efficient than our proposed method.

Moreover, our method would have the benefit of being able to take any image and determine whether it is a posed or spontaneous facial expression without having to do any detection of apex and onset images.

1.3.1 Research questions

1. Does a pre-trained and finetuned Inception V4 model perform better than a shallow CNN at the task of classifying posed and spontaneous facial expressions?

2. Will an Inception V4 model that is pre-trained and finetuned be able to capture if a person exhibits a posed emotion or a spontaneous emotion, for instance, posed angry or spontaneous angry?

3. How does the performance differ when one feeds a facial expression image directly into a model, compared to using a set of hand-crafted features that are fed into a model?

Chapter 2

Theoretical Background

This section describes the theoretical knowledge required to understand the contents of this thesis. Subsection 2.1 elaborate on the theory of CNN and the benefits of using CNN over a conventional neural network. Subsection 2.2 describes the state of the art computer vision model used to classify spontaneous and posed facial expressions. Subsection 2.3 explains the machine learning algorithm SVM that is used to create a baseline for the problem.

2.1 Convolutional neural network

Convolutional neural networks(CNN) have in the past years become vastly popular for image classification problems. Up until 2012, the field of deep learning received less attention, as it struggled to be able to solve complex tasks accurately, such as the ImageNet challenge. In 2012 a team entered the ImageNet competition, which is a large-scale database for object detection. The entry, named AlexNet, surpassed the predecessors by halving the existing error rate from 28 % and 26% to 16%. AlexNet spiked the interest of many researchers and showed people that CNN could solve complex tasks. AlexNet utilized the performance of GPUs to train their model. The use of GPUs proved to increase performance drastically and enabled them to train larger models.

CNNs are very similar to conventional feedforward neural networks. The differences are that CNNs are far superior when dealing with images than a conventional neural network. It assumes that every input has the shape of an image, where it assumes the input has a height, width, and depth. The input, however, does not have to be an image.

In figure 2.1, shows a CNN architecture. The red block is the input image, which has a height, width, and depth. The depth equals to the number of channels for the image, where in this case it is 3(Red, blue, green). From one block to another 3 operations are done, namely convolution, activation and then pooling.[8]

The convolution operation takes filters and slide them across the input. There is commonly more than one filter, and each filter is trying to capture distinct features. An activation function is a function applied to the convolution output and results in an activation map. The activation function is a non-linear function with a threshold that decides whether the neuron should fire or not. The resulting output of the activation function is often called a feature map and corresponds to the features captured by the convolution operation. After this, the pooling layer downsamples the output of the activation, meaning it reduces the height and width dimension of the image.

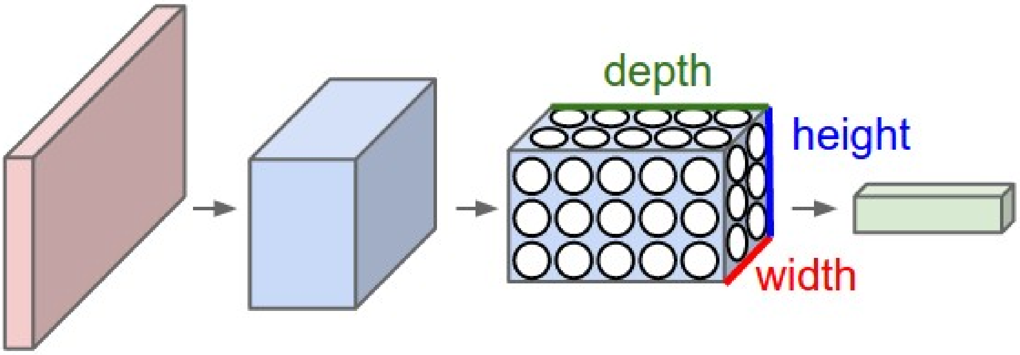


Figure 2.1: Overview of a Convolutional neural Network. Shows that a ConvNet arranges its neurons in three dimensions (width, height, depth) [8]

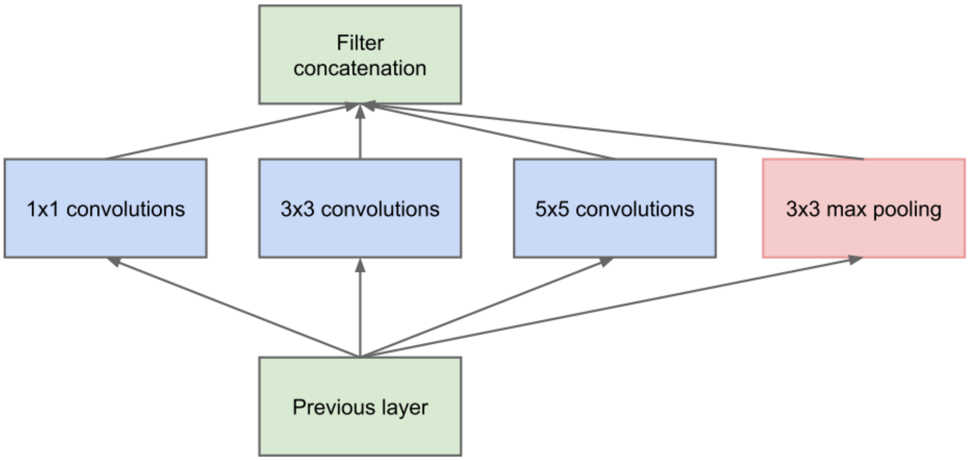


Figure 2.2: Inception module, that shows how one utilizes wider networks

2.2 Inception-V4

Inception V4 is a CNN architecture that has proven to be one of the best methods for solving the ImageNet challenge. It is deep with in total 137 convolutional layers. Inception V4 is the fourth version of the Inception architecture. Inception V1 main idea was to go wider and then concatenate the filters. Figure 2.2 shows an inception module from Inception V1 that consists of several smaller convolutional layers and a pooling layer. The outputs are concatenated to form an output with a larger depth dimension.[17]

2.3 Support Vector Machine

Support Vector Machine (SVM) is a machine learning algorithm based on statistical learning theory. SVM classifies data by dividing it into groups using a decision boundary. This boundary often occurs as a line, separating a linearly separable dataset. A higher order of dimensionality requires other forms of division, for instance, a 3-dimensional dataset requires a plane to separate the dataset. It is possible to separate a dataset such that the error rate is be zero, however, the classifier would not be able to generalize to new data well. Thus to achieve good results on new data one has to generalize the separation of the dataset, which is done by optimizing the hyperparameters C, gamma, and which kernel to use.[10]

SVM is considered an overall good classifier and therefore makes a good baseline classifier for our thesis. CNN generally performs better than SVM when classifying images, which is why we set the SVM classifier as our baseline. To evaluate whether our approach is valid; its performance is evaluated against the SVM classifier performance.

Chapter 3

State of the art

This section presents the state-of-the-art methods used for solving the problem of differentiating between posed and spontaneous facial expressions of emotion.

3.1 Bayesian networks

Bayesian networks(BN) are probabilistic graphical models, which are used to represent knowledge of an uncertain domain. The nodes in the graph represent variables, where the edges between the nodes are probabilistic relationships. The graph can be used to determine the conditional probability of a variable, given another. Dynamic Bayesian networks(DBN) are BN with the addition of the concept of time. This addition of time makes it possible to model time series or sequences, which is especially useful in video.[13]

There was a paper released in 2011 by Melinda Seckington, named ”Using Dynamic Bayesian Networks for Posed versus Spontaneous Facial Expression Recognition.” The paper used dynamic Bayesian networks to differentiate between posed and spontaneous facial expression of emotion. The paper did not use the same dataset as our thesis does; nevertheless, it proves that by using dynamic Bayesian networks one can classify between posed and spontaneous facial expressions. The method relies on a set of rules taken from psychologist researchers. The paper states that the purpose of the study is not to evaluate the performance of a fully automated system, but instead determine whether the set of rules are essential to the task of distinguishing between posed and spontaneous facial expressions.

There are five rules that the paper used as variables for the DBN, namely Morphology, Apex overlap, Asymmetry, Total Duration, and Speed. Morphology refers to the presence of the face muscle that raises the cheek and tightens the upper and lower eyelid. The absence of this muscle is a reliable indicator of a posed smile, while the presence does not necessarily mean a spontaneous smile. Some features are not vital to our thesis and therefore omitted. These features are as follows: apex overlap, total duration, and speed.

Asymmetry refers to asymmetry in the expression. The paper states that asymmetries are more frequent in posed smiles than in spontaneous smiles. Asymmetry occurred in both spontaneous and posed facial expressions, but in the case of posed expressions, asymmetry was stronger on the left side. However, in the case of spontaneous expressions, asymmetry was equally prominent on both sides of the face. These features are important to our thesis, as they are hopefully going to be captured by the Inception V4 model. [14]

The dataset Melinda Seckington used is called MMI, which is a dataset of spontaneous and posed facial expressions with videos. She achieved 97% accuracy, which proves that the features used are essential for classification of posed versus spontaneous facial expressions. Their research proved that some features are more important than others, for instance, morphology, apex overlap, total duration and speed of onset were found to be important. Asymmetry and speed of offset did not contribute to the DBN classification, but are good indicators for spontaneous and posed facial expressions. [14]

Latent Regression Bayesian networks(LRBN) is a particular kind of BN that introduces a latent layer. The latent layer directly connects to the visible layer, where the visible layer is the same as a regular BN. There has been work done in the field of differentiating between spontaneous and posed facial expressions using LRBN with promising results.

Gan et al. released a paper in 2017 using LRBN to differentiate between posed and spontaneous facial expressions, named ”Differentiating Between Posed and Spontaneous Expressions with Latent Regression Bayesian Network.” The work bases itself on the premise that the method can capture both the dependencies among latent variables given the observation and the dependencies among visible variables. They use two LRBN to capture the spatial patterns, where each of them is respectively for posed and spontaneous facial expressions. The input takes the displacement of facial points between apex and onset. They do not take into account the temporal features of facial expressions, which might be a shortcoming of the method. They utilize the NVIE database and SPOS, where they currently to the best of our knowledge has the best accuracy. They achieve an astounding 98,74 % accuracy of the NVIE database, compared to the previous best work listed in the paper at 92,61 %. [5] However, this paper may have been released at the same time as Chang Xu et al. Therefore the previous state of the art for this dataset achieved a performance of 97.96%.[2]

3.2 Restricted Boltzman machine

Restricted Boltzmann Machine (RBM) is a machine learning algorithm that is a generative stochastic artificial neural network.

Wu et al. released a paper in 2016 using RBM to differentiate between spontaneous and posed facial expressions, named ”Posed and Spontaneous Expression Recognition Through Restricted Boltzmann Machine.” The paper introduced a novel approach using RBM to model global spatial patterns of posed and spontaneous facial expressions. Compared to previous work that has used hand-crafted features, their solution does not extract features by hand. Their solution assumes that the RBM can capture the features required to classify between posed and spontaneous facial expression on its own.

Their solution proposes to use two RBM models to classify between posed and spontaneous facial expressions. Figure 3.1 shows the overview of the solution Wu et al. proposes. Their solution uses the displacement of facial feature points to extract facial regions of size 100 x 100 pixels. They use the displacement of the facial feature points between the onset and apex frame as features. For each facial feature points, the ”displacements are discretized with unequal interval”[18]. These intervals represent a specific movement of facial feature points, called facial event in the figure 3.1. Their method requires a considerable amount of preprocessing to acquire all the facial events of an image, whereas our proposed solution does not need the same amount of preprocessing. Their solution does perform well on the USTC-NVIE database with an accuracy of 81.23%.[18]

Wang et al. released a paper in 2016 using RBM, named ”capturing global spatial patterns for distinguishing posed and spontaneous expressions.” Their method uses RBM with similar structure as Wu et al. [18] and appears to be an extension of that paper. The solution introduces gender and expression categories as privileged information into posed and spontaneous expression distinction. This privileged information is only available during training. Their solution uses mul-

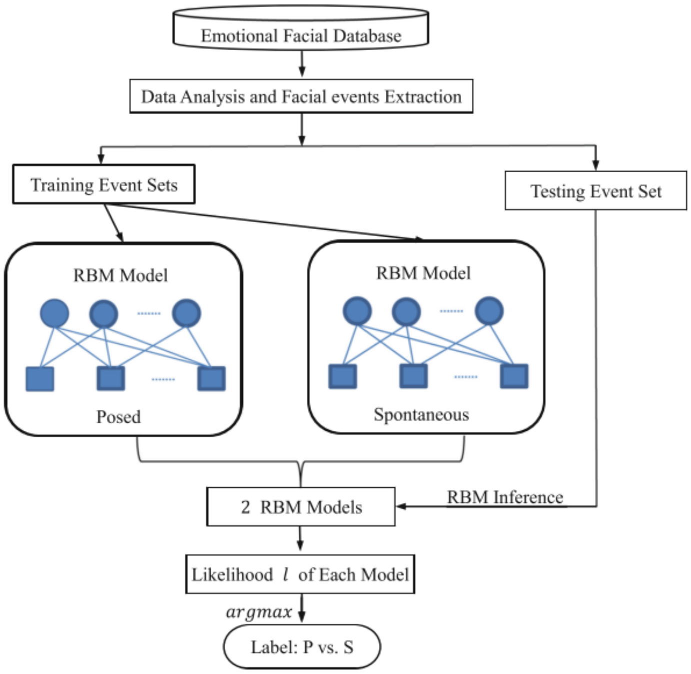


Figure 3.1: Overview of solution of Tian et al.

tiple RBM to distinguish between posed and spontaneous facial expression. Their solution achieves 91.71% accuracy without using privileged information, while when using gender as privileged information the accuracy increases to 92.24%. [21]

3.3 Support Vector Machine

The paper from Wang et al.[21] has implemented SVM as a baseline algorithm for their method. Their approach used the same features as discussed in subsection 3.2. SVM is relatively easy to implement and generally does well on most classification tasks, which is why it often is implemented as a baseline. Their approach achieved an accuracy of 81.52%.

Osuna et al. release a paper in 1997 using SVM for face detection, named ”Training Support Vector Machines: an Application to Face Detection.” They use a dataset containing about 50,000 images, labeled with face and non-face. They show that an SVM can accurately detect the location of faces. Also, it shows that an SVM can capture the features necessary to classify faces. Their approach achieved 97,1% detection rate and outperformed state of the art in 1997.[12]

3.4 Other approaches

There have been several other attempts at differentiating between posed and spontaneous facial expressions. Dibeklioglu et al. released a paper in 2010 that used 3 different classifiers to detect whether a smile was posed or spontaneous. Their approach used eyelid movement as a feature. They proposed a method for distance-based and angular features for eyelid movements. They tested the reliability of these features by using continuous hidden Markov models(HMM), k-nearest neighbor (k-NN) and naive Bayes(NB) classifiers. They tested the classifiers on BBC-smile and Cohn-Kanade dataset. On BBC-smile dataset HMM, k-NN and NB achieved the same classification rate of 85%. While on Cohn-Kanade HMM, k-NN and NB achieved 82.6%, 87.0%, and 91.3% respectively.[3] It has to be taken into account that these datasets are relatively small compared to NVIE and SPOS. Nevertheless, it shows that eyelid features are essential to classification between posed and spontaneous facial expressions.

The problem of face detection is a similar problem to ours, as both involve capturing facial features from images. Sun et al. release a paper in 2015, named ”DeepID3: Face Recognition with Very Deep Neural Networks”. Their approach bases itself upon the architectures of VGGNet and Inception. They propose two different architectures, where both are altered versions of VGGNet and Inception. Sun et al. use the dataset Labeled Faces in the Wild(LFW) for testing, which is a dataset containing images of faces gathered from the internet and labeled. They perform two classification tasks, namely face verification and identification. Face verification is a task where the goal is to compare two faces and tell whether they are the same person. Face identification, on the other hand, is a task where the goal is to identify a person by looking at his or her face. Their approach achieves state-of-the-art performance on both face verification and identification, respectively 99.53 % and 96.0%. Their approach shows that it is possible to capture facial features using only one image per face.[16]

Farfade et al. released a paper in 2015 for multi-view face detection, named ”Multi-view Face Detection Using Deep Convolutional Neural Networks.” They propose a novel method that does not require facial landmarks or annotation of face poses, as other state-of-the-art approaches need. They have named their method Deep Dense Face Detector (DDFD). They have designed DDFD to be as simple as possible; hence it does not require additional components such as segmentation, bounding-box regression or SVM classifiers. They state that DDFD can detect faces from different angles and can handle occlusions to some extent.

DDFD is a CNN that uses an altered version of the AlexNet architecture. They fine-tuned a regular AlexNet and changed its fully connected layers into convolutional layers. DDFD can efficiently run the method on images of any size and obtain a heat-map of the face classifier because of the converted fully connected layers. Their approach reaches state-of-the-art accuracy; however, their approach appears to be the better choice because it reduces the computational complexity when compared to other state-of-the-art methods. [4]

3.5 Convolutional neural network

CNN (Convolutional Neural Network) is a machine learning technique that excels at classification of images. Using CNN for classification of posed vs spontaneous facial expressions of emotion has been done by chang Xu et al. They propose to use a custom comparison layer that compares the onset and apex frame of the facial expression sequence. Onset is considered as the start of the facial expression sequence, while apex is the peak of the facial expression where the emotion is displayed at its fullest.

They argue that the common practice of comparing the onset and apex frames by taking the pixel difference of the raw images loses vital information and also introduces new noise. This method introduces noise because the comparison is done on low-level pixels which makes for a noisy result. The paper proposes that by comparing two images after the first convolution layer, one decreases the noise that is introduced by comparing two images. This is explained by that the comparison is done after preprocessing and abstraction and is, therefore, able to keep vital information. One can theorize that the convolutional layer can reduce noise by extracting the core features corresponding to the task.

The method proposed by this thesis assumes that one has two images, namely onset, and apex, which in a real-world scenario would add additional overhead to acquire and process.

The work does not cover the best state of the art results where the current best holds 98,74 % on the nvie database, while they achieve 97, 96%. However, this work shows that by utilizing convolutional neural network one can achieve high performance and therefore one can argue that by using a pre-trained Inception V4 network one can outperform state of the art.

3.6 Inception V4

Inception V4 is a state of the art model that has outperformed the previous state of the art ILSVRC (ImageNet Large Scale Visual Recognition Competition) models. It improved upon the previous inception models by increasing the model size, while keeping the overall number of parameters and computational cost roughly the same. It achieves an astounding 3.08 % top-5 error. This model has not been used[To the best of our knowledge] in the task of differentiating between spontaneous and posed facial expressions. Due to its high performance in classifying objects from ImageNet, it is safe to hypothesize that the model will perform well for the task at hand.

Chapter 4

Approach

We have implemented a solution to the problem discussed in chapter 1. At the start of development, we had to make certain assumptions, so that we were able to achieve our goals. This is discussed in section 4.1. Our solution uses the database NVIE, which we discuss in section 4.2. This database requires preprocessing to be ready to be fed into a model, which we discuss in section 4.3. We have developed several algorithms to solve the problem at hand, which we discuss in section 4.4. Our final solution has some restrictions regarding design, which we discuss in section **??**. To test the performance of our solution we had to have specific evaluation metrics, which we discuss in section **??**.

4.1 Assumptions

The model is greatly simplified when considering data that is not part of the dataset. The model is not able to classify images that do not represent any facial expressions. The model is also not able to classify an image accurately if there is no face involved in the image. Both of these assumptions were necessary to achieve the goals set for this thesis.

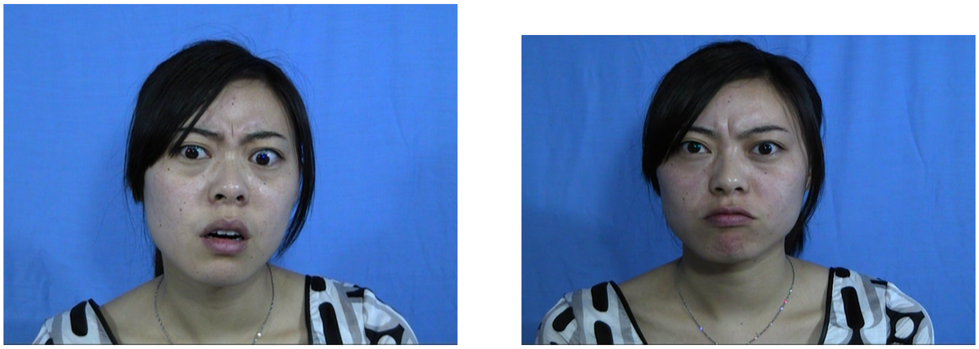
If one were to try to introduce classes for no face, one would need more data to define what is not a face. Also, it would prove a much more complicated task to solve.

4.2 Dataset

Most of the databases available for emotion recognition only focuses on posed facial expressions. These databases do not scale well for real-world applications as most of the expressions in a real world are spontaneous. Wang et al. made the database Natural Visible and Infrared Facial Expression Database(NVIE), to fill the void of spontaneous and posed databases. The database includes both posed and spontaneous facial expressions and has also labeled each with the corresponding emotion. Wang et al. have collected six emotions of both posed and spontaneous, where they are expressions of happiness, disgust, fear, surprise, sadness, and anger.

Spontaneous expressions were collected by showing videos that were supposed to induce a particular emotional reaction. 5 students evaluated the subjects' expressions through evaluation of the intensity of the six emotion categories. Wang et al. selected the emotion category with the highest average intensity as the label. To ensure that the expressions were spontaneous, the experimenters did not require the subjects to keep their head in a fixed position, as requiring so would feel unnatural for the subjects and therefore the expression would not be spontaneous. To acquire posed facial expressions, the experimenter asked the subject to perform a series of expressions in front of the camera without any stimuli. The subjects were required to pose for images with and without glasses for the posed database. In figure 4.1, two images are displayed of spontaneous and posed anger. Us as humans can see a difference between the two when they are beside each other. However, if a person sees any of the images without its counterpart, it might prove harder to estimate if it is a posed or spontaneous anger facial expression.

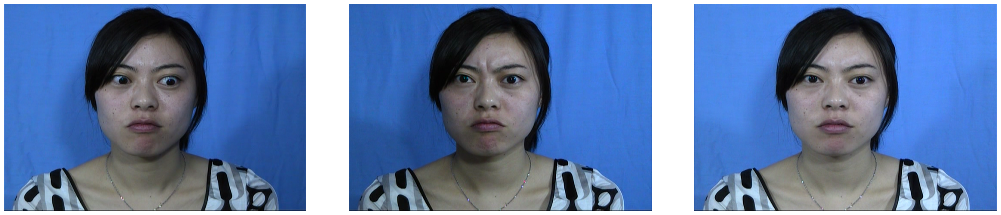
Classification of images is generally vastly affected by lighting conditions and for a model to be able to generalize it is necessary to provide the model with images showing different lighting conditions. Wang et al.[20] acquired all of the



(a) Spontaneous Anger

(b) Posed Anger

Figure 4.1: Example images of subject showing spontaneous and posed anger



(a) Posed Anger left illumination

(b) Posed Anger front illumination

(c) Posed Anger right illumination

Figure 4.2: Example images of posed facial expression with different illumination directions

images with different illumination directions, i.e., light from the front, left, and right. Figure 4.2 shows facial images with different illumination directions and how they affect the image. By utilizing all of the lighting conditions, the model should be able to generalize better to unseen images.[20]

4.3 Preprocessing

To be able to use the data, one needs to gather, filter out and augment it. Loading the data has required the use of two separate algorithms for posed and spontaneous data, as their folder and file structure was different from each other.

4.3.1 Loading data

A data loader was implemented to load the data. Posed and spontaneous images were as we previously mentioned stored in different folders with different file structures for both of them. Figure 4.3 show us the different file structure for posed and spontaneous. The figure shows the file structure after we had restructured the spontaneous database. It was necessary to restructure the spontaneous database because the file containing the emotion of the spontaneous database was not labeled properly. The old structure contained a folder of apex images and 6 folders of sequences. The sequence folders contained roughly 60 images, where they represent a series of images of a subject's face from onset to the apex. However, each folder did not contain a sequence. The missing sequences meant that for a subject there could be 3 or 4 apex images, which did not have a name corresponding to which sequence it belongs to. The missing synchronization between apex and sequences made it difficult to identify the apex images that belong to each label.

The identification problem was solved by traversing through each sequence folder and checking whether the apex image corresponded to that sequence. If we find a match, the image is moved to the new file structure as shown in figure 4.3.

To increase the performance of the model, we omitted spontaneous facial expression images that did not correspond to any emotion. We omitted these images because including these in training is outside the scope of this thesis.

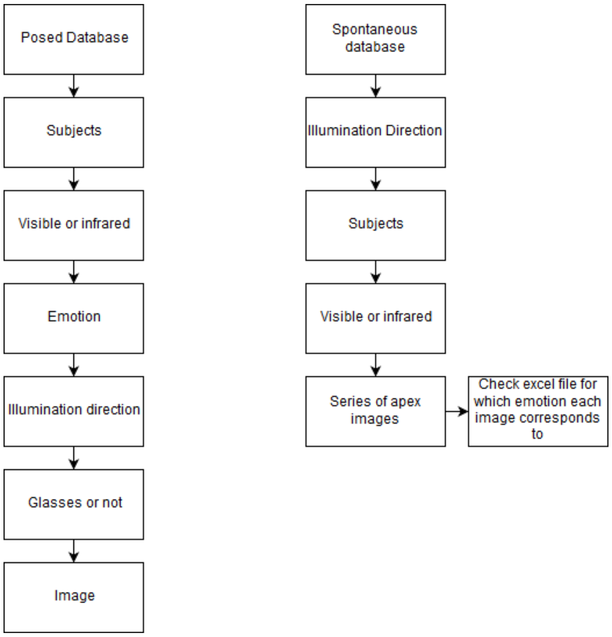


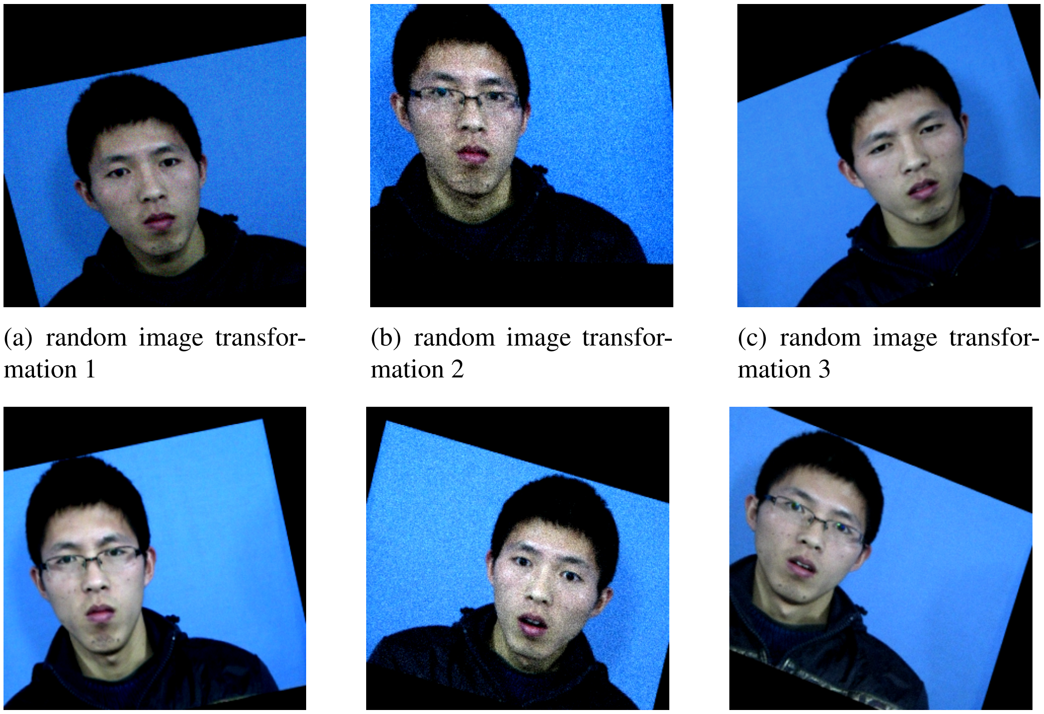
Figure 4.3: File structure and operations

4.3.2 Data Augmentation

Machine learning algorithms perform better when it has more data because it can generalize better to new data. However, it is laborious and time-consuming to gather sufficient data and often impossible for researchers. It is impossible for some due to the time required to gather and label data. However, there exists a way to increase the data at hand, namely data augmentation. [19]

Data Augmentation enables us to take a small dataset and alter the images several times to increase the size of the dataset. Also, data augmentation helps with the model's ability to generalize to new data. Images are often augmented by translating x and y, scaling, rotating, changing contrast, distorted or shaded with a hue. The NVIE dataset consists of images of facial expressions where the subjects are seated in front of a blue background. Considering a model that does not utilize the power of data augmentation. That model has not seen data in different conditions, such as different lighting conditions, object in different positions, and data that is warped. The model would therefore not be able to generalize well to new data. This is one of the reasons data augmentation is so beneficial. [19]

Figure 4.4 shows 6 different images augmented by the library imgaug[7]. The transformations applied are respectively rotation, scaling, translation, gaussian blur, flipped, cropped, contrast change, Gaussian noise and changing color. We transformed each image randomly. Also, some transformations were activated randomly for each image. The image augmentation library imgaug makes it easy to transform images as it has built-in functions. It is important that the augmented data is still intact when using them to train a neural network. In addition, the augmented images need to have the object of importance inside of the image. By looking at the images, one can see that the face of the subject is still intact and inside of the image.



(a) random image transformation 1

(b) random image transformation 2

(c) random image transformation 3

(d) random image transformation 4

(e) random image transformation 5

(f) random image transformation 6

Figure 4.4: Example images showing transformations done to images

4.4 Algorithms

This section firstly describes how and which hyperparameters were selected. Afterward, we discuss the primary algorithm used, namely Inception V4 and the CNN model used. Last, we discuss the implementation of an SVM classifier used as a baseline for performance.

4.4.1 Hyperparameters

Common to all deep learning algorithms, is the need for hyperparameter search to get acceptable results. Hyperparameters, in the sense of deep learning, are a set of parameters which has to be set appropriately by the developer to maximize the usefulness of the algorithm[1]. Most commonly there are only a handful of hyperparameters that affects the performance, however, finding these parameters is often difficult and time consuming[1].

Our solution uses a grid search to search for hyperparameters, albeit a limited one. The grid search is limited in the way that we assume that some variables do not directly affect each other concerning performance and therefore does not need to be optimized simultaneously. Michaus has shown through empirical work that the learning rate and batch size is dependent upon each other. Figure 4.5 shows the results of the work. The information one can draw from this figure is that the higher the batch size, the higher the learning rate has to be.

Table 4.1 shows the hyperparameters used to find the optimal model. By looking at the table, one can infer from knowledge about CNN's that some parameters are not directly dependent upon each other concerning performance. An example of this is Augmentation and batch size, by varying the batch size and turning on and off augmentation one experiences the same amount of increase in performance.

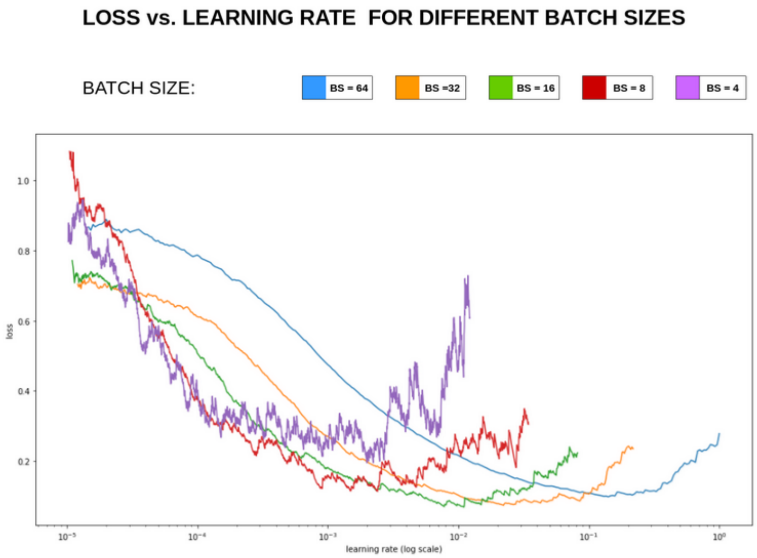
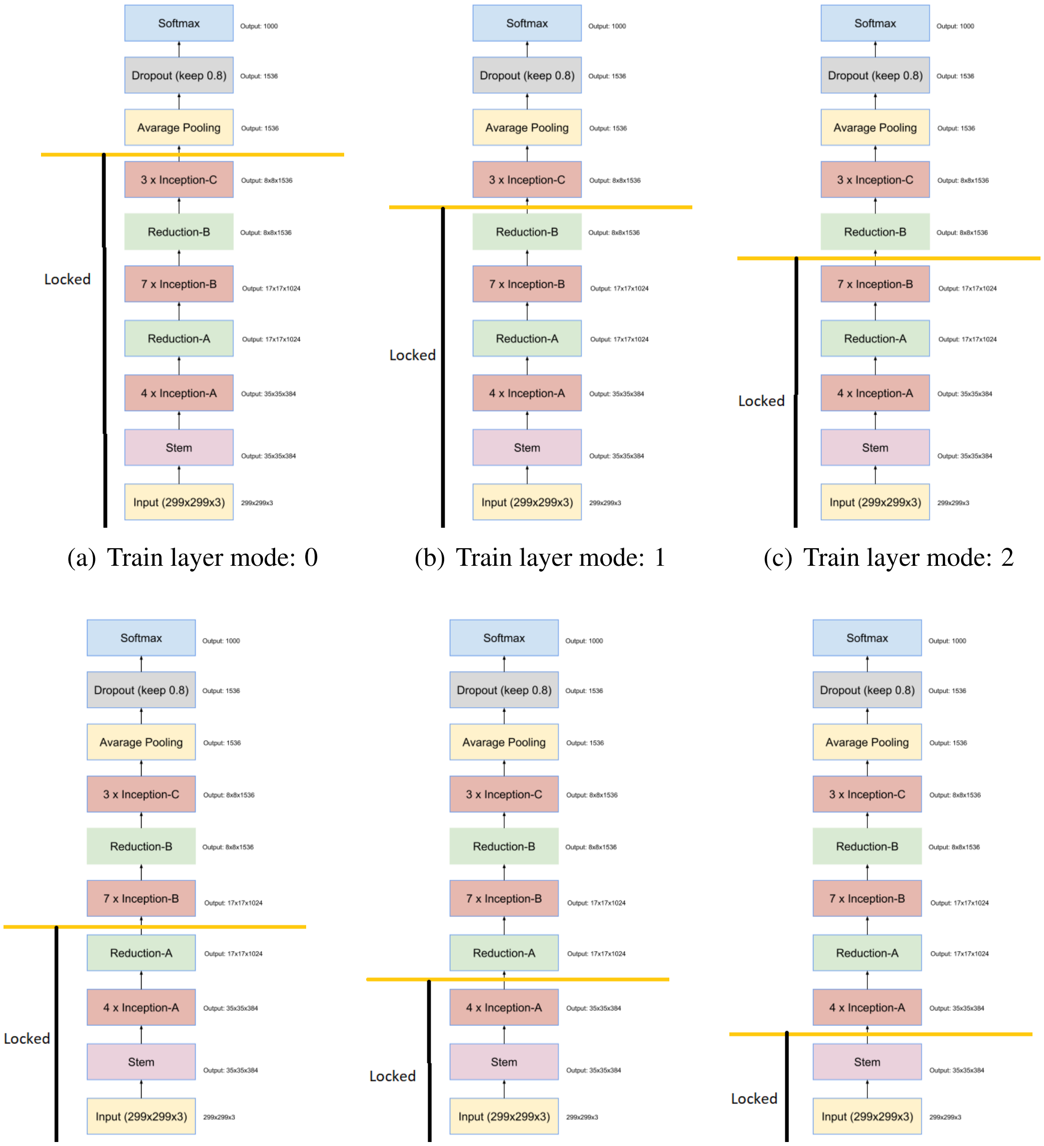


Figure 4.5: Learning rate versus Batch size [11]

parameter	values
Dropout	[0.0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0]
Learning rate	[1e-3, 1e-4, 1e-5, 1e-6]
Train layer mode	[0, 1, 2, 3, 4, 5]
Softmax or Sigmoid	[Softmax=True, Sigmoid=False]
Batch Size	[8, 16, 32, 64]
Learning rate decay rate	[0, 2, 4, 8, 10]
L2 regularization	[True, False]
Augment train set	[True, False]

Table 4.1: Table of hyperparameter used

The different algorithms used require different hyperparameters. If one looks at the table 4.1 and the row Train layer mode. Figure 5.4 shows the hyperparameter train layer mode and how it affects which layers that we train. This parameter, we explain further in subsection 4.4.2.



(a) Train layer mode: 0

(b) Train layer mode: 1

(c) Train layer mode: 2

(d) Train layer mode: 3

(e) Train layer mode: 4

(f) Train layer mode: 5

Figure 4.6: Images showing which layers are locked with respect to parameter train layer mode. Ever layer below the yellow line are locked for gradient updates.

4.4.2 Inception V4

The Convolutional Neural Network Inception V4 was implemented in python using the library Tensorflow. We implemented Inception V4 as described in the paper by Szegedy et al.[17]. We implemented the model by using Tensorflow's low-level API methods, which was done to get a deep understanding of the Inception V4 model. By doing this, we are saving time in the future; for instance, if we run into problems with the model, we are better equipped to determine what is wrong.

Inception V4 is a 100+ layer deep network, which is considered to be a profoundly deep network. By having such a deep network, one becomes more prone to overfit on the data. To battle this, Inception V4 has used dropout in the layer in front of the output layer. Additionally, Szegedy et al. designed the Inception V4 model toward the classification of the ImageNet dataset, which contains over a million images. Szegedy et al. may have optimized the model to such an extent that it is only suited for datasets similar to ImageNet.

The Inception V4 model is trained on a wide variety of images and has therefore captured essential parts of what an image is. Most datasets do not have an enormous amount of samples, which is why transfer learning is often used to train deep models on small datasets. Transfer learning is defined as training a model on a large dataset and using the weights learned on a new classification task. The process of adapting the learned weights to a new classification task is called fine-tuning.

It is essential to look at how different the two datasets are when fine-tuning the Inception V4 model. Considering a new dataset, that is small but similar to the original one. In this case, it is often most beneficial to only train the last layer containing the classes as training more than the last layer will most certainly end in an overfitted model. However, a dataset that is similar to the original but substantial in size is often better to train on the whole model and use the weights as initialization. Training the whole model is possible because the new dataset is large and therefore overfitting is not of concern.[9]

4.4.3 CNN

We implemented a shallow CNN model that we train from scratch. The idea behind implementing a simple CNN model is to get an indication whether the problem of differentiating between posed and spontaneous facial expressions is a simple or complex task. If it is a complex task, which it most likely is, the better choice of a model is Inception V4. However, if it turns out to be a simple task, then the better choice would be a shallow CNN model.

The CNN model was implemented using Tensorflow's low-level API, which allowed us complete control of all its parameters. We made the solution as flexible as possible by allowing us to change the number of convolutional and fully connected layers. Being able to change the number of layers allows us to add the number of layers as a hyperparameter for the grid search.

We optimized the hyperparameters for the CNN model with a grid search on a subset of the variables mentioned in table 4.1. We omitted both of the hyperparameters train mode, and sigmoid/softmax because they are specific hyperparameters for the Inception V4 model.

Figure 4.2 shows the layers used for the shallow CNN model. This architecture is designed by us to be as simple as possible and also be able to capture features in a face. The architecture contains 8 layers, including convolution, pooling, and fully connected layers.

Type of layer	Input size	Output size
Convolutional	[batch size, 96, 96, 3]	[batch size, 96, 96, 8]
Convolutional	[batch size, 96, 96, 8]	[batch size, 96, 96, 16]
Max pool 3x3 pool and 3 stride	[batch size, 96, 96, 16]	[batch size, 32, 32, 16]
Convolutional	[batch size, 32, 32, 16]	[batch size, 32, 32, 24]
Convolutional	[batch size, 32, 32, 24]	[batch size, 32, 32, 32]
Max pool 3x3 pool and 3 stride	[batch size, 32, 32, 32]	[batch size, 11, 11, 32]
Fully connected	[batch size, 3872]	[batch size, 2048]
Fully connected	[batch size, 2048]	[batch size, 2 or 12]

Table 4.2: CNN model architecture

4.4.4 SVM

Support Vector Machine(SVM) was implemented, as previously mentioned, as a baseline for the other algorithms. As previously mentioned, an SVM classifier performs relatively well on most classification problems, which makes it perfect to set as a baseline performance. The idea behind setting a baseline performance is to see if our model outperforms that method. We consider our approach good if it outperforms the baseline set by the SVM. However, if it does not, it could indicate that our approach is inadequate for the problem.

In order, to spend less time on implementation and more time on the main algorithms, the machine learning library SkLearn was used [10]. The library contains everything one needs to implement an SVM classifier in relatively short time. However, like most other machine learning algorithms, the SVM classifier performs best if we optimize its hyperparameters.

To find the best possible parameters we first tested GridSearch to find the best combination of the parameters listed in table 4.3. However, this turned out to be very time consuming and therefore a new approach was needed. We decreased the amount of data by omitting images with illumination direction from left and right. By doing this, we assume that the best parameters for images with frontal illumination direction, also applies to images with illumination direction left and right. In addition to this, we applied Random search for hyperparameters instead of grid search. The difference is that the random search does not go through every possible combination of the parameter grid; instead, it samples randomly from the grid until it has done n iterations. We have set the iterations to 70, in that way it does not take several weeks to find the best parameters.

Table 4.4 shows the confusion matrix of an SVM classifier that we trained by using standard parameters found on sklearn's website [10]. The table 4.4 shows that the SVM is not able to differentiate between spontaneous and posed facial expressions. We set the c value to 1.0, and the gamma to 0.001. Usage of default

	Parameters
Gamma	[0.00001, 0.0001, 0.001, 0.01, 0.1, 1.0, 10.0]
C	[0.001, 0.01, 0.1, 1.0, 10.0, 100.0]
Kernels	[Rbf, Poly, linear]

Table 4.3: SVM possible parameters

parameters was done to see if the SVM was able to classify the images without searching for hyperparameters. It is safe to say that the decision boundary the SVM made with these default values classifies every image as posed facial expressions.

We ran a hyperparameter search using Scikit's library for a random search. The search returned a c value of 0.01, gamma of 0.001, and polynomial kernel. We ran an additional run with these optimal values such that we could verify the accuracy and generate a confusion matrix. Table 4.5 shows the confusion matrix for this new run. Compared to the SVM classifier ran the first time, the new run for SVM does indicate that it is better to differentiate between posed and spontaneous facial expressions. Spontaneous appear to be accurately classified as spontaneous facial expressions, however, posed most of the time appears to be classified as spontaneous. Comparing table 4.4 with 4.5 shows that the SVM classifier used in 4.5 is a lot better to differentiate between posed and spontaneous facial expressions. The SVM classifier from table 4.4 classified all spontaneous and posed facial expression as posed. This bias towards posed indicates a decision boundary not capable of dividing the dataset. Figure 4.6 shows us the accuracy of both the runs . Both have fairly similar accuracies; however, after looking at the confusion matrix for each, it is evident that the SVM classifier with optimal values is better to divide the dataset.

		Predicted Values	
		Spontaneous	Posed
True Values	Spontaneous	0	656
	Posed	0	711

Table 4.4: Confusion matrix of SVM first model without hyperparameter optimization

		Predicted Values	
		Spontaneous	Posed
True Values	Spontaneous	408	95
	Posed	651	464

Table 4.5: Confusion matrix of SVM model with hyperparameter optimization

	Accuracy (%)
SVM before optimization	52.01%
SVM after optmization	53.8%

Table 4.6: Result of SVM before and after hyperparameter optimization

Chapter 5

Results and discussion

In this chapter, we discuss and show the experiments conducted by us. The experiments are designed to test the performance of our approach against state of the art and give insight into the results.

Loss and accuracy are used to evaluate the performance of our CNN models. Both the CNN models use softmax with cross-entropy as the loss function. This is a loss function that takes the softmax of the model's output, which results in probabilities for each class. The cross-entropy function takes softmax as input and calculates the loss by calculating negative ln of the softmax probabilities. This means that we can calculate the threshold of when the model is outputting random classes. Differentiating between spontaneous and posed facial expression require two classes, spontaneous and posed. A CNN model that is choosing randomly between the two has a softmax of 0.5 for each. By calculating cross entropy, we get a value of 0.69. This implies that a CNN model with a loss of 0.69 or above is randomly guessing between the two classes. While if the loss is below, the model has been able to capture features relevant to the task. We use this same evaluation method for the task differentiating between spontaneous emotion and posed emotion, however, the threshold for random guessing is in this case 2,48.

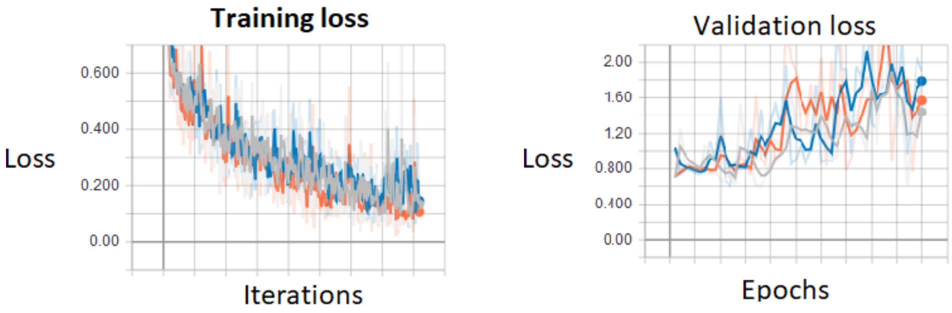
5.1 Experiment 1

Experiment 1 is designed to compare the performance of the Inception V4 model against our baseline algorithms and state of the art. The goal of this experiment is to classify images as either posed or spontaneous.

Our results on Inception V4 has proven to be not as good as one would expect and hope. The results have altered this experiment to be a proof that Inception V4 is not well suited for classifying spontaneous and posed facial expressions. This experiment is going to show the different hyperparameters that have been tested to see if the performance increases.

Our Inception V4 approach did struggle with several problems at the start of development; however, they were all sorted out. Our initial approach to the problem was to search for a good learning rate, as it most often is the culprit when the performance is lacking. To find suitable parameters quickly, we first searched shallowly after parameters. The search was shallow concerning the number of epochs trained. The initial search had many parameters to search through, which is another reason the number of epochs was set low. We set the number of epochs to 30, and then we evaluated the best results by training the model for 50 epochs.

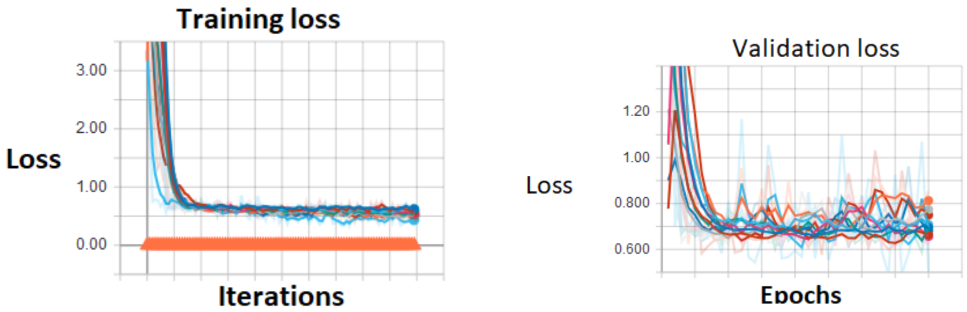
Figure 5.1 shows 3 different runs with different learning rates. We have chosen these runs from previous shallow learning rate search where they turned out to be the best, and therefore a deeper search was needed. The figure shows that the training loss is decreasing, which means that the model is learning the training set. However, looking at the validation loss, we can see that it does not decrease, but increases. The increase in validation loss indicates that the model is overfitting. When the model is overfitting, the first thing to check is whether the dropout is too high and search for a better value. Also, to test that dropout is working properly, a dropout of 0.0 is used. Figure 5.2 shows the different dropout values that are used. The orange graph shows us that the dropout is working, as the value for training loss is undefined. The other results are similar in performance,



(a) Training loss

(b) Training loss

Figure 5.1: Figure showing learning rate of 0.001, 0.0001, and 0.00001 with dropout of 0.8



(a) Training loss

(b) Training loss

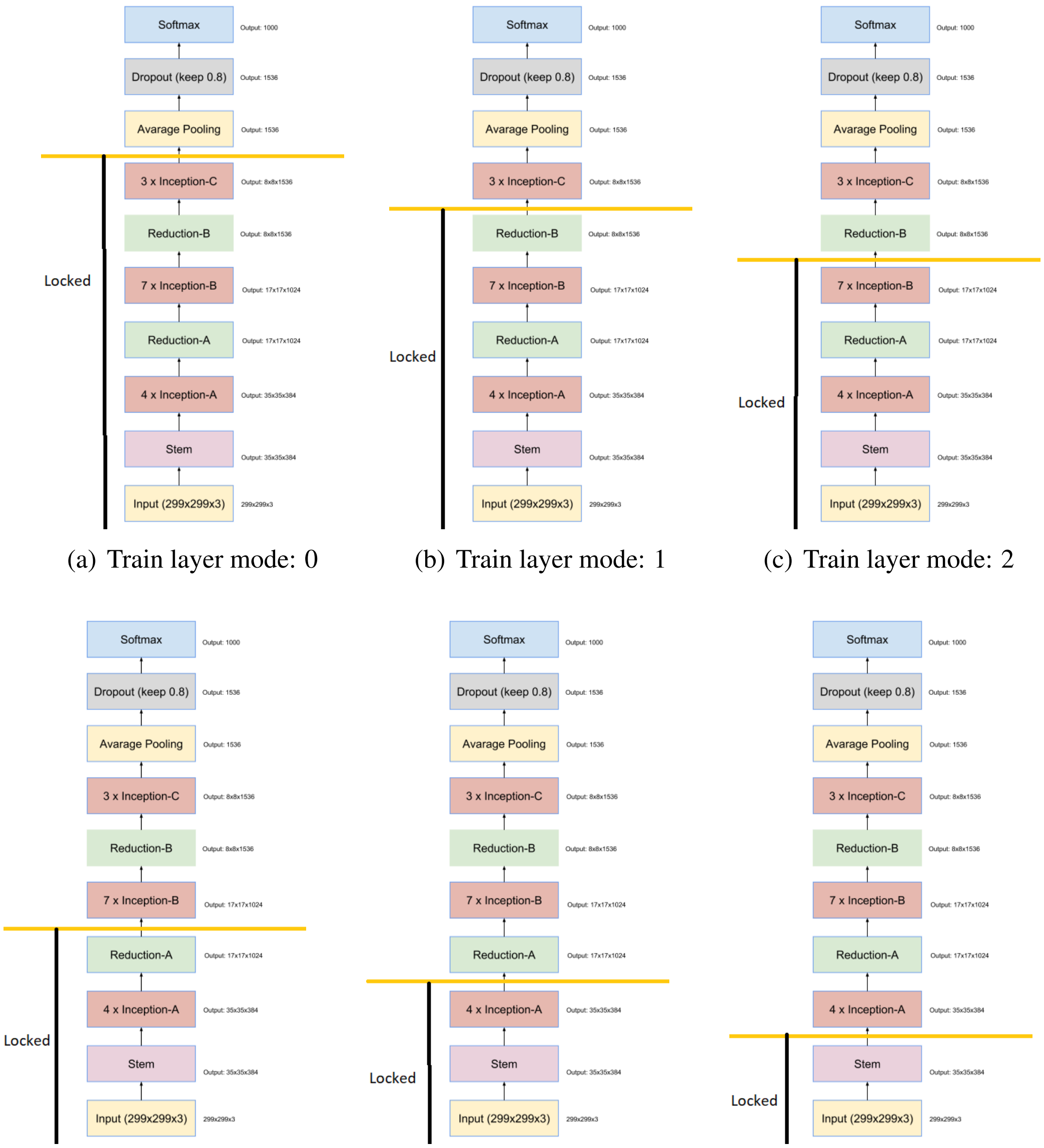
Figure 5.2: Figure showing dropout search ranging from 0.0 to 1.0 with learning rate of 0.001

which indicates that the dropout is not the culprit for the lacking performance. These results indicate that the hyperparameter train layer mode may have to be optimized.

Figure 5.4 shows 5 different runs with the self-made hyperparameter train layer mode. The figure shows the validation loss of the different runs. We can see from the results that all of the runs increase from the start. This increase in loss indicates severe overfitting. All of the runs are above the loss threshold of 0.69, which indicates that the model is randomly guessing output. From the results, we can see that there is a need for a deeper search of the train layer mode hyperparameter. We perform a deeper search by locking individual parts of the layer Inception-C, seen in figure 5.3. The figure shows Inception-C as being stacked three times. Each of the Inception-C modules is usually locked in train layer mode 1. We propose to try to lock each Inception-C module and see if this increases the validation performance.

Figure 5.5 shows the results of our investigation of locking separate iterations of Inception -C module. The results show more promise than the previous attempt shown in figure 5.4. The results indicate that with a batch size of 64 and learning rate of 0.00001 are the optimal batch size and learning rate for the task. This has been further verified by the results shown in figure 5.6. These results have been trained using early stopping, meaning that the training of the model does not stop until the validations loss stops to decrease. Both the validation and training loss are below the random guessing threshold of 0.69, which indicates that the model has been able to capture relevant facial features. The model achieved an accuracy of 67.4%. The model has been greatly optimized towards the task of differentiating between posed and spontaneous facial expressions, but not all have been shown in this thesis. For instance, sigmoid cross entropy loss function has been tried without improvement. Also, other implementations of Inception V4 has been tested without improvement.

Our shallow CNN model results do show more promise than the Inception V4 model. Figure 5.7 shows multiple runs of different batch sizes and learning rates. The results of the validation loss are not shown at this stage in the hyperparameter search, because it is essential that the model can learn by training. When we have optimized the model for training; we can focus on validation performance. We can see that the model can learn by looking at the loss. With a batch size of 16, we can see that the result with a learning rate of 0.001 does far better than the runs with learning rate lower than 0.001. We can see the same trend in both runs with a batch size of 32 and 64. The results indicate that the learning rate of 0.001 with batch size 16 is the best learning rate. The results with batch size 32 and



(a) Train layer mode: 0

(b) Train layer mode: 1

(c) Train layer mode: 2

(d) Train layer mode: 3

(e) Train layer mode: 4

(f) Train layer mode: 5

Figure 5.3: Train layer mode hyperparameter. Weights are locked for updates, by the optimizer, below the line.

64 indicate that the learning rate is too low. We can see that the learning rate is

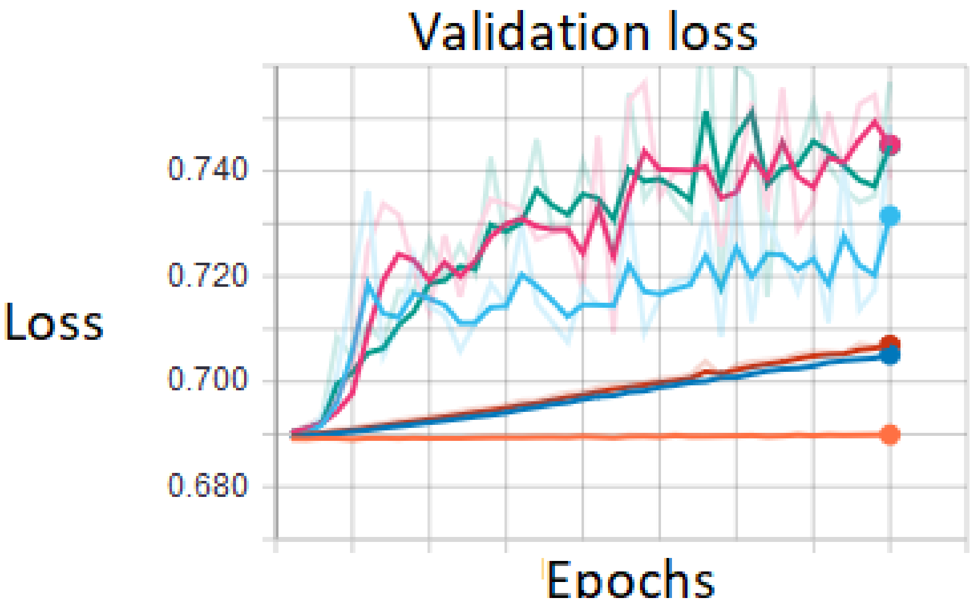
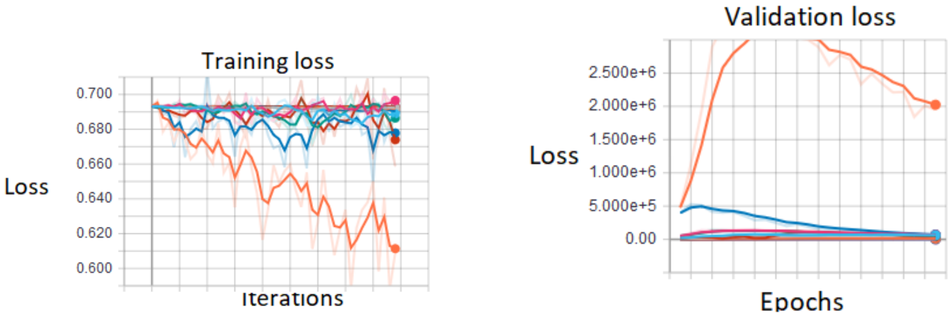


Figure 5.4: Inception V4 train mode layer optimization for

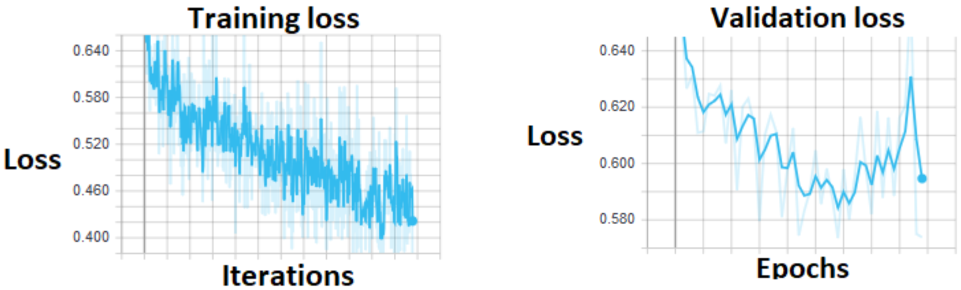


(a) Training loss

(b) Valiation loss

Figure 5.5: Validation and training loss for Inception V4. Learning rate from 0.001 to 0.000001, batch size of 64, dropout of 0.5, and train layer mode ranging from 1 to 3

too low because the graph converges very slowly. The results show us in what range the learning rate should be. The next logical step is to train with a higher learning rate with batch sizes 32 and 64. Also, it is essential to start to improve



(a) Training loss

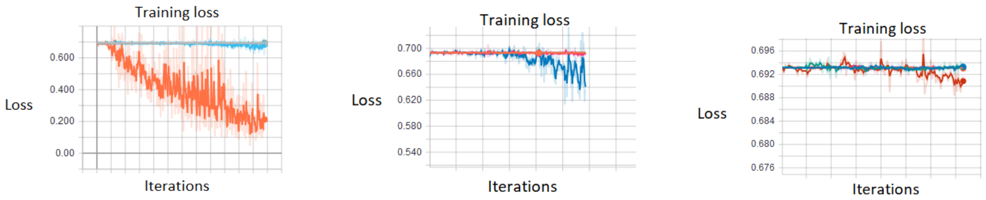
(b) Validation loss

Figure 5.6: Inception V4 model final results. Using learning rate of 0.0005, batch size of 64, dropout of 0.4, l2 regularization, and data augmentation. Run for 44 epochs

the validation performance by searching for suitable dropout values.

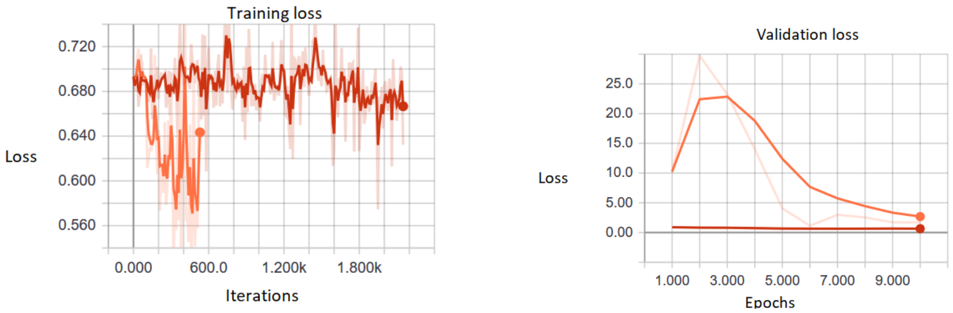
Figure 5.8 shows the most promising results for a hyperparameter search for dropout values for batch sizes of 16, 32 and 64. The figure shows 2 runs with a batch size of 16 and 64 with learning rates of 0.001 and 0.01, respectively. The run with a batch size of 16 has the best validation loss of 0.63, which is below the random threshold of 0.69. This indicates that these hyperparameters may be best suited for the problem. However, the run with a batch size of 64 does have a lower training loss. The run does have a higher validation loss, but the graph indicates that the loss is still decreasing. Both of these results are worth investigating further.

We introduce early stopping to our model, such that we can stop the model before it overfits. Also, we introduce data augmentation and l2 regularization to get the best possible result. Figure 5.9 shows the final model trained. The graph shows that the validation loss converges and we get the best possible result for that model. The model achieved an accuracy of 71.6%, which is a considerable increase compared to results gained before hyperparameter optimization.



(a) Batch size 16 (b) Batch size 32 (c) Batch size 64

Figure 5.7: Shows CNN training loss for batch sizes of 16, 32, and 64. Learning rate from 0.001 to 0.000005. Trained 30 epochs.Dropout of 0.8

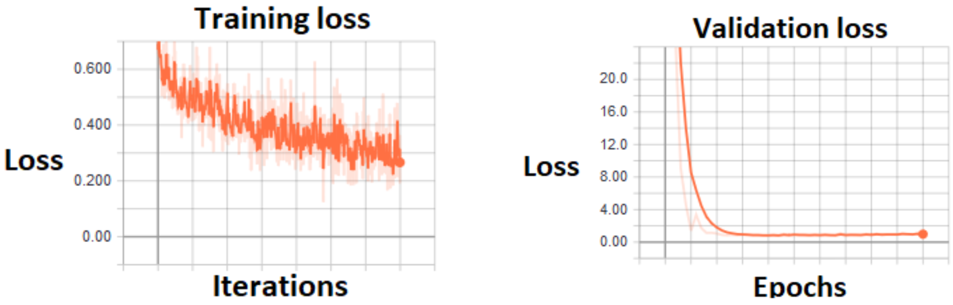


(a) Training loss (b) Validation Loss

Figure 5.8: Most promising results from runs with batch sizes of 16, 32 and 64. Batch size 16 has learning rate of 0.001. Batch size 32 has learning rate of 0.01. Batch size 64 has learning rate of 0.1 and 0.01. 3 runs for each configuration with dropout of 0.4, 0.5 and 0.6. Run for 10 epochs. The figure shows batch size 64 and 16 with learning rate 0.01 and 0.001 respectively.

Algorithm	Accuracy (%)
SVM	53.8%
CNN	71.6%
Inception V4	67.4%
Gan et al. [5]	98.74%
Chang xu et al.	97.96%

Table 5.1: Final results for differentiation between spontaneous and posed



(a) Training loss

(b) Validation loss

Figure 5.9: CNN model final results. Using learning rate of 0.00005, batch size of 64, dropout of 0.4, l2 regularization, and data augmentation

5.2 Experiment 2

This experiment is designed to show the performance of the Inception V4 model when faced with 12 output classes instead of previously 2. These 12 classes correspond to the posed and spontaneous emotion categories mentioned in the introduction. We performed this experiment despite the lacking performance of the Inception V4 model. Since the Inception V4 is an enormously deep CNN model, it can capture complex problems. We think that the problem of differentiating between spontaneous and posed facial expression may be too simple for the model. Therefore, it is necessary to show empirically that the Inception V4 does or does not work for differentiating between spontaneous emotion and posed emotion.

Figure 5.10 shows a learning rate search for classifying between spontaneous emotion and posed emotion. The results show that the blue graph is preferable. The blue graph has a learning rate of 0.0005. The orange graph, which has a learning rate of 0.001, appears to be converging too early compared to the blue graph. That result is a strong indicator of a learning rate that is too high. The blue line may be converging too early as well, but more testing is required to validate this.

Figure 5.11 shows the final model for Inception V4 of this experiment. The results show that the validations loss is below 2.48, which means that the model has learned relevant facial features on its own. The result was gathered by using early stopping to stop the model before it overfits. The model achieved an accuracy of 17.2%. When comparing the result to the SVM classifier, it appears to be a poor result.

Figure 5.12 shows the result for the shallow CNN model when classifying 12 labels. The validation loss is lower than the random guessing threshold of 2.48. This means that the model has captured relevant facial features. We have optimized this model by using the same parameters from experiment 1 and fine-tuned these to fit the task of differentiating between posed emotion and spontaneous

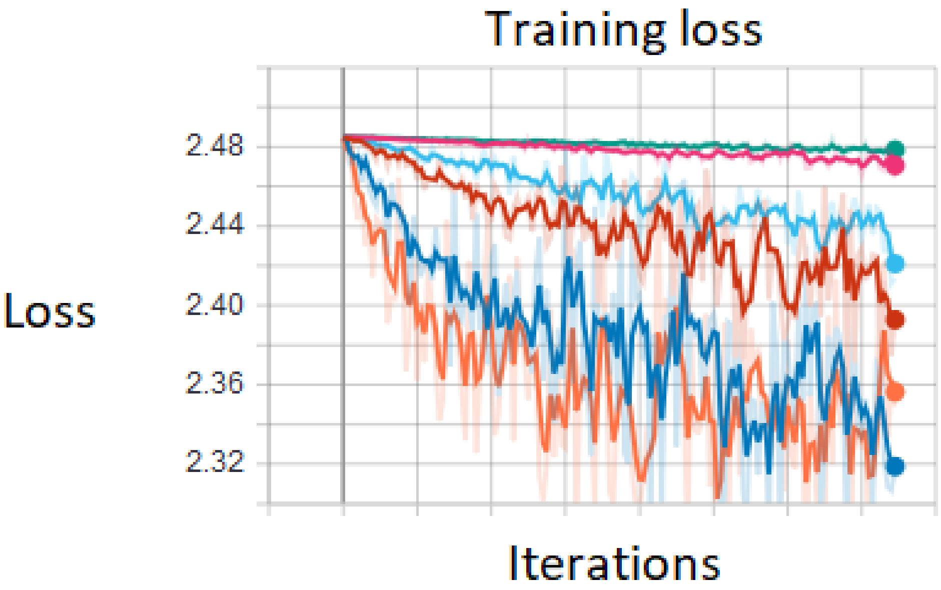
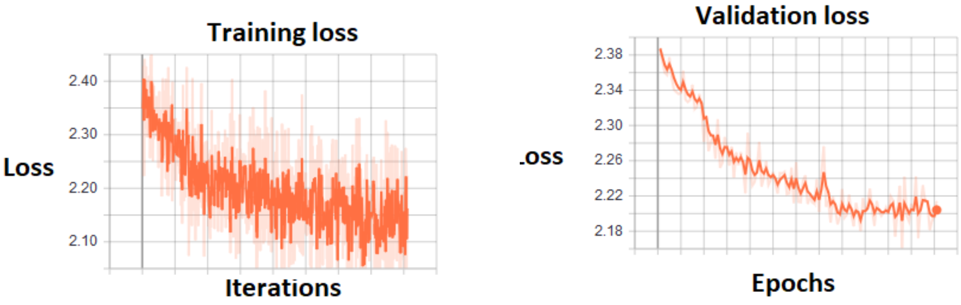


Figure 5.10: Training loss of Inception V4 model trying to classify spontaneouse and posed emotion. The different runs are for learning rates from 0.001 to 0.000005 and 30 epochs



(a) Training loss

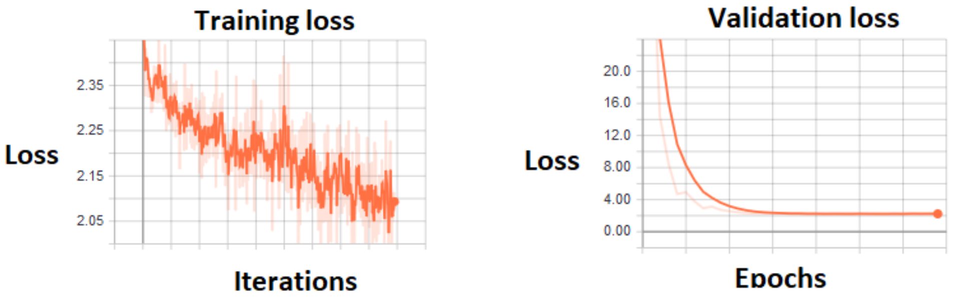
(b) Validation loss

Figure 5.11: Performance of Inception V4 model for differentiating between posed emotion and spontaneous emotion. Learning rate of 0.0005, dropout of 0.7, augmentation, and l2 regularization

emotion. However, all of these hyperparameter searches are not shown because it is quite similar to experiment 1 hyperparameter search. The final model achieved

		Predicted Values											
		Spontaneous Happy	Spontaneous Disgust	Spontaneous Fear	Spontaneous Surprise	Spontaneous Anger	Spontaneous Sad	Posed Happy	Posed Disgust	Posed Fear	Posed Surprise	Posed Anger	Posed Sad
True Values	Spontaneous Happy	58	0	9	17	15	81	29	7	31	15	0	8
	Spontaneous Disgust	3	0	0	2	1	6	0	0	0	0	0	0
	Spontaneous Fear	9	0	1	4	1	9	0	1	1	0	0	0
	Spontaneous Surprise	32	0	2	15	8	54	5	2	2	6	0	3
	Spontaneous Anger	20	2	2	7	5	29	0	2	0	2	0	3
	Spontaneous Sad	39	0	3	25	16	80	2	5	1	6	0	5
	Posed Happy	33	0	5	14	12	32	41	8	14	14	0	4
	Posed Disgust	31	0	4	12	12	41	18	11	28	14	0	5
	Posed Fear	29	0	8	9	6	40	14	7	37	24	0	3
	Posed Anger	0	0	0	0	0	0	0	0	0	0	0	0
	Posed Sadness	30	0	4	9	12	42	23	3	32	16	0	5

Table 5.2: Confusion matrix of SVM model with hyperparameter optimization 12 labels



(a) Training loss

(b) Validation loss

Figure 5.12: Performance of CNN model for differentiating between posed emotion and spontaneous emotion. Learning rate of 0.00005, dropout of 0.4, augmentation, and l2 regularization

an accuracy of 22.7% and is the best result that we got for this task.

		Predicted Values											
		Spontaneous Happy	Spontaneous Disgust	Spontaneous Fear	Spontaneous Surprise	Spontaneous Anger	Spontaneous Sad	Posed Happy	Posed Disgust	Posed Fear	Posed Surprise	Posed Anger	Posed Sad
True Values	Spontaneous Happy	0	0	0	3	0	18	0	0	0	0	0	3
	Spontaneous Disgust	0	0	0	0	0	3	0	0	0	0	0	0
	Spontaneous Fear	0	0	0	0	0	6	1	0	1	0	0	0
	Spontaneous Surprise	0	0	0	6	0	32	1	3	3	0	0	1
	Spontaneous Anger	0	0	0	1	0	11	0	0	1	0	0	0
	Spontaneous Sad	0	0	0	3	0	60	3	4	3	0	1	0
	Posed Happy	0	0	0	0	0	14	14	0	10	0	2	6
	Posed Disgust	0	0	0	0	0	18	6	1	11	0	3	8
	Posed Fear	0	0	0	0	0	15	9	0	13	0	3	6
	Posed Surprise	0	0	0	0	0	13	8	0	16	0	3	6
	Posed Anger	0	0	0	0	0	16	10	0	15	0	2	5
	Posed Sadness	0	0	0	0	0	16	7	0	13	0	4	6

Table 5.3: Confusion matrix of CNN model classification of posed emotion and spontaneous emotion

Algorithm	Accuracy (%)
SVM	17.2%
CNN	22.7%
Inception V4	17.3%

Table 5.4: Final results for differentiation between spontaneous emotion and posed emotion

5.3 Overall discussion

Figure 5.1 shows the state-of-the-art results compared to our own results. Our results do not come close to the results achieved in state-of-the-art, but they show that the models have been able to capture relevant facial features. Both the Inception V4 and the shallow CNN model has beaten the baseline SVM classifier. This indicates that our models are valid solutions. Figure 5.4 shows the same trend in the results for Inception V4 and the shallow CNN model. The shallow CNN model outperforms the Inception V4 model because the Inception V4 model is too deep and its sheer complexity when fine-tuning it.

Our results for both the tasks indicate that the models have been able to capture relevant facial features. The shallow CNN model does perform better on both of the tasks, which indicates that the tasks are too simple for an Inception V4 model. It also has to be taken into account that the Inception V4 model used has been heavily optimized for the ImageNet challenge and means that the Inception V4 model architecture needs to be altered for it to capture facial features better.

Chapter 6

Conclusion and further work

This chapter is going to conclude the findings of this thesis and suggest possible future work for the method proposed.

6.1 Conclusion

In this thesis, we introduce a novel method for differentiating between spontaneous and posed facial expressions. The method utilized the power of the Inception V4 model and compared to related work does have less preprocessing required. This thesis also introduces a shallower approach by using a shallow CNN model. It does have the benefit of being computationally more efficient, which means it is faster to train than the Inception V4 model.

Our results indicate that the Inception V4 can capture facial features when differentiating between spontaneous and posed facial expressions. However, the results are far away from the state-of-the-art performance which indicates that Inception V4 is too deep for the task and data available. The Inception V4 model achieved an accuracy of 67.4%, compared to the best state-of-the-art accuracy of 98.74%. The shallow CNN model implemented shows more promise than Inception V4 but is not close to the state-of-the-art performance. The shallow CNN model achieves an accuracy of 71.6%, which is an improvement over Inception V4. On the other hand, the shallow CNN model results show that a simpler Inception V4 model should be investigated further.

We conducted another experiment where we wanted to see if it was possible to differentiate between posed emotion and spontaneous emotion. Inception V4 achieved an accuracy of 17.3%; however, we do not have any other state-of-the-art results to compare this to, but it does indicate a poor result. The shallow CNN model performed better than the Inception V4. The shallow CNN model was able to reach an accuracy of 22.7%. All of the results outperformed the SVM classifier set as a baseline.

Our results indicate that the models can capture relevant facial features by itself. However, the results also show that using both the apex and onset facial images may be vital for differentiating between posed and spontaneous facial expressions.

6.2 Further Work

There are still other approaches that can be tested to find out if Inception V4 can differentiate between spontaneous and posed facial expressions. As this thesis has empirically proven, inception V4 is not able to classify spontaneous and posed facial expressions, but the solution feeds the image directly into the model.

As discussed in state of the art, related work has all used both the onset and apex frame to find the difference between onset and apex. Another approach to this problem would be to do like the paper using CNN to differentiate between posed and spontaneous and use the same method for comparing between onset and apex images. By using Inception V4 and this method, one would most likely get results very close or above state of the art.

DeepID3 is an approach discussed in state-of-the-art that uses a simplified version of Inception V4. Sun et al. achieved excellent results when detecting faces and therefore this may be an alternative approach to the problem of differentiating between posed and spontaneous facial expression. This can be backed up by our findings that the shallow CNN model is better suited at the task and therefore it makes sense that a shallow Inception V4 would outperform the shallow CNN model.

Bibliography

[1] M. Claesen and B. De Moor, “Hyperparameter Search in Machine Learning,” pp. 10–14, 2015. [Online]. Available: http://arxiv.org/abs/1502.02127

[2] I. I. Conference, “CONVOLUTIONAL NEURAL NETWORKS FOR POSED AND SPONTANEOUS EXPRESSION RECOGNITION,” no. July, 2017.

[3] H. Dibeklioglu, R. Valenti, A. A. Salah, and T. Gevers, “Eyes do not lie: spontaneous versus posed smiles,” *Proceedings of the international conference on Multimedia*, pp. 703–706, 2010. [Online]. Available: http://doi.acm.org/10.1145/1873951.1874056

[4] S. S. Farfade, M. Saberian, and L. J. Li, “Multi-view Face Detection Using Deep Convolutional Neural Networks,” *International Conference on Multimedia Retrieval 2015 (ICMR)*, p. 19, 2015. [Online]. Available: http://arxiv.org/abs/1502.02766

[5] Q. Gan, S. Nie, S. Wang, and Q. Ji, “Differentiating between posed and spontaneous expressions with Latent Regression Bayesian Network,” *31st AAAI Conference on Artificial Intelligence, AAAI 2017*, pp. 4039–4045, 2017. [Online]. Available: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85030460520{&}partnerID=40{&}md5=9bac267b297345b148ab1e7a93a764b8

[6] Q. Gan, C. Wu, S. Wang, and Q. Ji, “Posed and spontaneous facial expression differentiation using deep Boltzmann machines,” *Affective Computing and . . .*, pp. 643–648, 2015. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs{_}all.jsp?arnumber=7344637

[7] A. Jung, “imgaug.” [Online]. Available: https://github.com/aleju/imgaug

[8] A. Karpathy, “Convolutional Neural Networks (CNNs / ConvNets).” [Online]. Available: http://cs231n.github.io/convolutional-networks/

[9] ——, “Transfer Learning.” [Online]. Available: http://cs231n.github.io/transfer-learning/

[10] S. Learn, “1.4. Support Vector Machines¶.” [Online]. Available: http://scikit-learn.org/stable/modules/svm.html

[11] M. P. Michaus, “Visualizing Learning rate vs Batch size.” [Online]. Available: https://miguel-data-sc.github.io/2017-11-05-first/

[12] E. Osuna, R. Freund, and F. Girosit, “Training support vector machines: an application to face detection,” *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 130–136, 1997.

[13] S. J. Russel and P. Norvig, *Artificial Intelligence A Modern Approach*. Prentice Hall.

[14] M. Seckington, “Using Dynamic Bayesian Networks for Posed versus Spontaneous Facial Expression Recognition,” no. September, 2011.

[15] M. Shreve, S. Godavarthy, V. Manohar, D. Goldgof, and S. Sarkar, “Towards macro- and micro-expression spotting in video using strain patterns,” *2009 Workshop on Applications of Computer Vision, WACV 2009*, 2009.

[16] Y. Sun, D. Liang, X. Wang, and X. Tang, “DeepID3: Face Recognition with Very Deep Neural Networks,” pp. 2–6, 2015. [Online]. Available: http://arxiv.org/abs/1502.00873

[17] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. Alemi, “Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning,” 2016. [Online]. Available: http://arxiv.org/abs/1602.07261

[18] Q. Tian, N. Sebe, G. J. Qi, B. Huet, R. Hong, and X. Liu, “Posed and Spontaneous Expression Recognition Through Restricted Boltzmann Machine,” *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 9516, pp. 127–137, 2016.

[19] J. Wang and L. Perez, “The Effectiveness of Data Augmentation in Image Classification using Deep Learning,” *Unpublished*, 2017. [Online]. Available: http://cs231n.stanford.edu/reports/2017/pdfs/300.pdf{%}0Ahttp://arxiv.org/abs/1712.04621

[20] S. Wang, Z. Liu, S. Lv, Y. Lv, G. Wu, P. Peng, F. Chen, and X. Wang, “A natural visible and infrared facial expression database for expression recognition and emotion inference,” *IEEE Transactions on Multimedia*, vol. 12, no. 7, pp. 682–691, 2010.

[21] S. Wang, C. Wu, and Q. Ji, “Capturing global spatial patterns for distinguishing posed and spontaneous expressions,” *Computer Vision and Image Understanding*, vol. 147, pp. 69–76, 2016. [Online]. Available: http://dx.doi.org/10.1016/j.cviu.2015.08.007

Chapter 7

Appendix

7.1 Source code

Available at the following URL: https://github.com/kris456/IKT590