Efficient Facial Expression Recognition in Everyday Photos

Master's thesis in Artificial Intelligence, Radboud University

Wouter van der Weel - s4243773

Supervisor: Tom Heskes
Second reader: Johan Kwisthout
External supervisor: Jasper van Dalen

March 2019

Contents

Abstract

Automatic facial expression recognition has been a much-researched topic over the past decade and even before. However, most studies have used simple, lab-controlled data. In practice, facial expressions can vary significantly from image to image, due to differences in lighting, strength of the expression and pose, in addition to interpersonal differences. Most studies that use more practical data also use very large neural networks that lead to inefficient image classification or allow the networks to get familiarize with the test data. This project aims to classify facial expressions as accurately as possible for use in an online application with user-uploaded data. As such, a much harder dataset than is usual needs to be used to emulate the data and there are strong constraints on image processing time, inference time, and network size. Four network structures (VGG16, MobileNet, Xception and a small, simple CNN) were used with the AffectNet dataset. AffectNet contains a large amount of human-labeled images of facial expressions found on the internet. To make classification easier, image pre-processing that aimed to make the data more uniform - keeping only expression-related information - has been applied. Each network was trained twice to compare the results of this pre-processing. Additionally, it is studied what the networks have learned about the data through optimal input generation. It is shown that facial expressions can be recognized quite robustly by most networks and there are only relatively small differences in network accuracies. This is because the ambiguity in emotions creates much inconsistency in the labeled images, limiting any classifier's performance. The fact that the differences in network speed and size are large means that a very small network is most efficient on this dataset. Pre-processing does not unambiguously increase network performance, while it does reduce processing speed.

1 Introduction

Emotions are very important in our lives and in the interaction with others, because they carry a lot of meaning. People have a liking for smiling people in photographs, and seeing smiling people can give them a 'nice, warm feeling' [24]. Emotions also play a role in the formation of memories [25] and influence our affection for others [30]. Many people like to preserve fond memories of emotional events in the form of photos, but sometimes sorting these images into a meaningful collection is hard. The vast amount of photos in today's digital media can incur a very long processing time, whether by hand or with a computer. Identifying which photos carry the most meaning could be achieved by finding out which emotions are present in each image. Facial expressions are one of the most important measures of emotion in automatic emotion detection and as such have been studied extensively [17]. Facial expression recognition could thus be used to help in creating a subset of photos which is meaningful.

This project is done at Resnap[1], a Nijmegen-based company. Resnap creates photo books automatically. A user can upload a batch of any images they like, usually personal photos, from a vacation, their wedding or all photos taken in the last year. From these images, the best (according to several metrics) are taken and arranged in a custom lay-out. This project consists of developing the means to automatically and robustly recognize facial expressions in an efficient way, to add to the multitude of AI (artificial intelligence) features Resnap has to select the right photos for users. As stated, a user can upload any set of images they want, and all of these images will be processed by the system, as each may be important to the user. This means that the quality of the content of these images can vary greatly. Some users may upload professional photos, while others may want a photo book with photos taken by their smartphone, or a mix of the two sources. The images could have bad lighting conditions, obstructions or faces not looking at the camera. These conditions can make it hard to recognize faces, let alone facial expressions. Furthermore, in a company, expenses on a user-by-user basis have to be kept to a minimum. A user first

[1] https://www.resnap.com/

generates a book online and then decides whether to buy it. It should not take too much computational power to generate a book, in order to keep in check both waiting time for the client and server costs for the company. Therefore, there are restrictions on the network size and other methods that can be used to create robust facial expression recognition software.

The goal of this project is to create facial expression recognition for an application that creates a selection of photos that the user will like. Photos are selected based on multiple criteria, and one of these criteria will be the facial expressions in the photographs. Facial expression recognition should be done efficiently, since users will have to wait for its processing time. In short, the research question is:

> How can efficient facial expression recognition be created, to be used in large sets of images with substantial constraints on processing time per image?

In the next chapter (chapter 2), I will discuss related work , including the possible applications of, and contemporary data sets used for facial expression recognition. In chapter 3, I will explain some of the concepts behind neural networks. Afterwards, the research questions are discussed more extensively (chapter 4). In the methods section (chapter 5), I will explain the choices made to improve recognition of facial expressions and limit computation time, in terms of dataset, network architecture and pre-processing. Then I will continue with the experiments done (chapter 6). Afterwards, I will present the results of these experiments (chapter 7). In chapter 8, I will look back at the choices made and how they have influenced the results, in the discussion section. In the second-to-last chapter, I will discuss the possibilities for future research (chapter 9). Finally, I will conclude the project and answer the research question (chapter 10).

2 Related Work

For a long time, there has been a research interest in detecting facial expressions from images or video footage automatically using artificial visual systems [38]. Ever since the 1960s, researchers have tried to construct such a system, using varying methods and with varying degrees of success [8]. These techniques include the use of artificial neural networks (ANNs), rule-based analysis [32], naive Bayes classifiers [10], local binary patterns [41], support vector machines [7], and spatially-localized facial dynamics using geometric features of the face [40]. Of these techniques, ANNs are the most reliable, as most of the other techniques are less robust to changes in lighting condition, inter-personal differences, and variation in the strength of the expression [34, 47].

2.1 Applications

One of the first examples of an experiment using a convolutional neural network (CNN) to automatically detect facial expressions stems from 2003 [29]. Although there were only two categories - whether subjects (N=10) were smiling or talking - the accuracy was very high, at 97.6%. This shows that automatic facial expression recognition using a CNN can work and has been available in some form for a long time. The classes of emotion that are used most often in contemporary facial expression recognition are happiness, anger, sadness, fear, surprise and disgust. These categories are based on the basic, universal emotions as defined by Paul Ekman, which are perceived equally across cultures [15, 14]. Some approaches also use contempt or the 'neutral' expression, which indicates that there is no emotion that is very strong or prevalent. Even with this large increase in the number of categories, some current approaches are still able to achieve accuracies of over 80% in predicting the facial expression from a face [11, 13]. Because Resnap has customers all over the world and the user-uploaded images are irregular, using all universal

emotions is important. The system will have to be applicable across cultures and cover all possible emotions.

An automatic facial expression recognition system has many possible applications. Emotion recognition could be applied any time an automated system can get improved by reacting to a user's emotional state. This includes testing customer satisfaction while using products, or marketing, education, and entertainment purposes [21]. There are numerous emotion detection APIs (Application Programming Interfaces) that allow users to apply automatic facial expression recognition in their software. Examples include Microsoft Azure's emotion recognition API [5] or APIs made by one of the many emotion-detection based tech companies such as Affectiva [1], Face++ [4], or Emotient [2], a company acquired by Apple. However, as they can only be accesed through the API, it is impossible to know how the underlying algorithms work exactly.

The popularity of facial expression recognition is in part demonstrated by the existence of multiple facial expression recognition challenges. Premier examples are the annual Emotion Recognition In The Wild challenge (EmotiW) [3] and the Facial Expression Recognition and Analysis challenge (FERA) [46], which has been organized three times since 2011. Both challenges have focused on a variety of subjects over the years, such as achieving high accuracy with straightforward emotion recognition, intensity of emotions, as well as the combined emotion of a group of people. Clearly, improving the automatic recognition of facial expressions by digital systems is still an important and widely-researched topic in the current scientific literature.

2.2 Datasets

Most of the previous approaches (e.g. [11, 13, 16]) use facial expression datasets that show posed subjects with exaggerated expressions under good lighting conditions, such as the Cohn-Kanade (CK) [27], JAFFE [28], FER+ [6] or RaFD [23] datasets. These datasets are constructed in a laboratory and facial expressions in them are very easily recognizable (see Figure 1). The images are all of the same size and

Figure 1: *Example of the Radboud Faces Database, a relatively clean dataset, with lab-controlled images.* [23]

color scheme, faces within them are very uniform in appearance and make up the majority of the image, and the subjects in the images clearly show which emotion is portrayed, because they are posed and exaggerated. The creators of the data sets set up lighting beforehand and give a cue that the subject should show a certain expression. These datasets also do not contain many different subjects, and the same subjects are used in (each of the) different emotion categories. Also, many approaches that use

these datasets use the same subjects in both the training and in the test set. This makes the problem a lot easier within the test set, as the facial expression recognition software can get used to and perform better on these subjects' faces [26]. It has been shown many times that a good accuracy can be achieved in this way [11, 13, 42]. However, this does not mean that this performance will translate to more realistic data. It has been shown that a network trained on a relatively easy dataset will generalize poorly to images from one of the harder datasets or to real-world examples [42]. A more complex and natural-looking dataset will be needed to get good performance on user-uploaded photos.

While most research to date uses one or more of these clean datasets, there is some research on harder and more varied datasets, such as the SFEW [12] or BP4D [51] datasets. The first one contains 700 images of acted facial expressions from movies. The latter one contains 3D facial expressions that are spontaneous (not posed after a cue), but to correctly capture all of the information in the data, larger networks are necessary. The EmotiW and FERA challenges have also focused more on "in the wild" datasets in recent years. Much of this research is conducted with video data, allowing a system to adjust its rating of the subject's expression over time.

Using a very clean dataset for network training can have its merits. Using substantial pre-processing, any low-quality images that need to be classified could become much easier to classify [26]. Using clearer images such as those in the above datasets to train might be fruitful if it is possible to obtain the necessary features from the low-quality test images using pre-processing. A network trained on such images may be easier to train in the first place and may still perform quite well on pre-processed images.

3 Facial expression recognition using neural networks

In recent years, the primary technique for detection of facial expressions has been to use convolutional neural networks (CNNs). Convolutional neural networks are a type of artificial neural network (ANN). Neural networks can learn an input-output mapping based on examples. For example, a neural network that is given enough images labeled with categories (e.g. 'cat' and 'dog') can 'learn' to which category new images (of cats and dogs) belong.

3.1 Artificial neural network basics

An ANN (see Figure 2) is a weighted graph consisting of a certain number of layers. The layers are ordered, and by definition the first layer is the input layer and, often, only the last layer is the output layer (intermediate input and output layers may exist). In between these layers are a number of hidden layers (zero or more). The more hidden layers a network has, the 'deeper' the network is considered to be. Each layer consists of a number of nodes (also called neurons) that are connected by weighted edges to other nodes in previous and later layers. A neural network can map an input to an output by calculating a function for each node over the inputs it receives from nodes in previous layers. After the network gives an output, the weights in the network are adjusted based on how large of an error the network has made. The process of adjusting the weights based on the error is called backpropagation [35]: the effect of the error in the last layer is passed back through each layer of the network, and the weights in each layer are adjusted.

[2] https://en.wikipedia.org/wiki/Artificial_neural_network

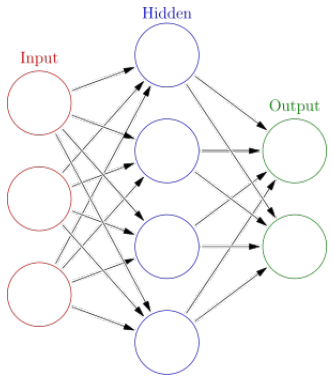


Figure 2: *Example of a (fully-connected) artificial neural network, also called a multi-layer perceptron (MLP). This network has an input layer with 3 nodes, a hidden layer with 4 nodes, and an output layer with 2 nodes. Each node is connected to all other nodes in the previous layer and all nodes in the next layer.*[2]

3.2 Convolutional neural networks

A CNN (see Figure 3 for an example) is an ANN that uses convolutional layers instead of fully-connected layers. Convolutional layers are loosely based on the visual system of humans. In a convolutional layer, each node is called a *filter*, which has its own receptive field, just as cells in the occipital cortex do. Each filter calculates an output based on a combination of subsections of the input and transmits this to the next layer. A filter has a certain size, for example 3x3 (see Figure 4). This filter slides over the input

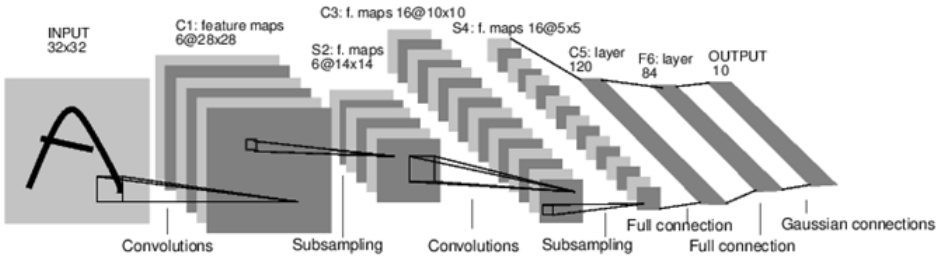


Figure 3: *Example of a convolutional neural network called LeNet. The input is shown (an image of the letter 'A'), followed by convolutional and subsampling layers. The output of the network is preceded by several fully-connected layers.*[3]

image until it has been in every possible position. For every position, the pixel values in the original image are multiplied with the values in the filter. These element-wise computations are summed up, resulting in a number that is a single value of the feature map. The feature map represents a feature, which means that it results in a high activation when a particular pattern exists in the image. For each filter that we use in a network layer, another feature map is created, which means we can represent an additional feature in that layer. The pattern represented by a feature map is more simplistic at earlier convolutional layers (like a diagonal line or a curve) and more advanced at later convolutional layers (like an eye or a mouth) (see Figure 5). This means that deeper convolutional networks can represent

higher-level features. In this way, parts of the input are summarized and the input to the next layer is

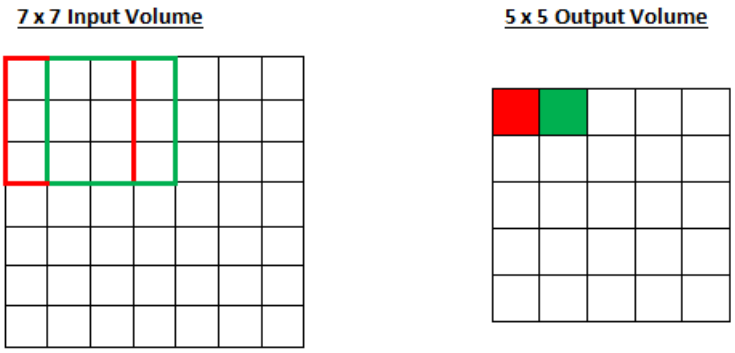
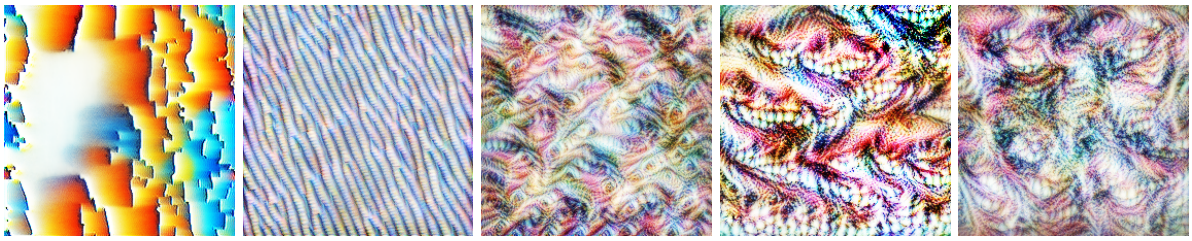


Figure 4: *Example of a 3x3 convolutional filter (left) and a 5x5 feature map (right). The filter summarizes each 3x3 part of the input and calculates 1 value for the output volume.*[4]

much smaller than in a fully-connected layer. For image processing, this is very important, because every pixel is a relevant variable of the input. If we would use a (fully-connected) ANN for the same problem, this would lead to a combinatorial explosion when using many layers, as the amount of parameters (that can influence the output of the network) for each node in the network would be equivalent to the width times the height of the image in pixels. By summarizing areas of the input image, we can significantly reduce the amount of parameters per node. Using a 3x3 filter in a 7x7 input volume will result in a 5x5 output volume, as there are 25 unique positions the filter can occupy. Using small filters will result in



(a) Layer 7 (b) Layer 27 (c) Layer 62 (d) Layer 73 (e) Layer 85

Figure 5: *Examples of patterns that network nodes in specific layers are most focused on. In earlier layers (a), these patterns are very abstract. In later layers (b) they become more intricate. In the later layers (c), (partial) objects are already visible (i.e. eyes). In the last layers, (d) clear objects are showing (mouths with teeth) as well as (e) combinations of different objects in one filter (mouths and eyes).*

a small reduction of volume size, but will let the filters focus on the details in the image; larger filters will result in a lower volume size and less focus on details. Each filter has a set of weights, one for each part of the input. The weights for each part of the input are shared between filters, because a feature detected by the filter could also be relevant in other areas of the input. These types of networks can be used effectively for image processing.

[3] https://adeshpande3.github.io/assets/LeNet.png
[4] https://adeshpande3.github.io/assets/Stride1.png

3.3 Network training

A neural network can learn to map certain inputs to certain outputs (for example, an array of pixel values to a class label representing an emotion) by tweaking the connection strength between nodes, called the weights. This process is called training. Training involves supervised learning with a training function. This means that the network receives an input for which the correct output is known to the training function. If the network gives the wrong output, the training function updates the weights based on the error of the output, and the activations of the nodes in the network are updated to be closer to values that give the correct output. The network needs a large amount of data and many training iterations to converge. Luckily, there are many datasets of images available for many different challenges, including emotion recognition.

Because all network weights need to be updated and trained for several iterations, convolutional neural networks can take a very long time to train depending on their structure. The development of large-scale deep neural networks accelerated when GPUs started to get used for the training of these networks. In 2009, it was first shown how much faster GPUs are for deep learning than CPUs [33]. GPUs are more suited for this purpose, because they are capable of many more parallel computations than CPUs, even multi-core CPUs. In 2011, Krizhevsky et al. [22] showed the power of GPU-accelerated training by training AlexNet, a much deeper network than state-of-the-art approaches at the current time. Training this network would have taken too long in practice if they had not been able to use GPUs. They entered their network, called AlexNet, for the ImageNet Large Scale Visual Recognition Competition (ILSVRC) 2012 [36]. The ILSVRC has been held annualy from 2010 onwards and has focused on object recognition in more than 10.000 categories, with at least 1000 images to describe each category. AlexNet achieved a substantially better result than the second-best and previous years' entries.

The developments in the use of deep CNNs are very important for automatic facial expression recognition. Previously, hand-constructed features were used [38] and inferencing the expression from an image could take a very long time. This can now be done much more efficiently. A network has to be trained first, but once it is trained, it can be used to recognize facial expressions multiple times per second. Until recently though, applications using this recognition did not work in real time (2.5 frames per second) [13]. Larger networks will take more memory to load and, more importantly, will take more time to make an inference. However, accuracy of deeper networks is often better. Thus, there is a trade-off between accuracy of a network and its inference time and memory usage.

(a) Original (b) Bounding box (c) Landmarks

Figure 6: *Example representation of face detection (b) and landmark detection (c).*

Furthermore, facial expression recognition depends on other deep neural networks to be more efficient.

Usually, the first step to facial expression detection is face detection. Face detection is simply the detection of where all the faces are in an image, and putting a bounding box around them (See Figure 6b).One might recognize such bounding boxes their digital cameras with face detection. Detecting faces can be done efficiently with a Viola-Jones detector [47] or a deep convolutional neural network, and the bounding box can restrict the area in which the next steps have to be done. Next, it is common practice to detect facial landmarks within the bounding box, which are a set of (usually 68) important points on the face (See Figure 6c). These points indicate how the subjects' lips, eyes and eyebrows and other facial features are positioned relative to each other. These distances can be used as features for a neural network for both face recognition (recognizing which person is on the picture) and facial expression recognition.

4 Research questions

To reiterate: due to the nature of being applied at a company, the neural networks that are used have to be relatively small in size and processing methods will have to be relatively simple and fast. This is because, in contrast to many scientific approaches, time and cost constraints for each user need to be considered to maximize profit. Whereas some researchers push the boundaries of how large networks can be in order to increase performance, in this project, the networks need to be constrained to a size of at most about 50 MB (For comparison, AlexNet [22] is about 250 MB). Loading in a network of a larger size is simply not feasible for Resnap. It is also necessary to reduce the load time of the network and the inference time when the network is used online. In practice, inference time should not be more than about half a second per image. As stated before, facial expression recognition is often done with easy datasets, but this research does not translate well to harder-to-classify datasets like user-uploaded photos. It is also done on harder datasets, but this most often involves very large networks with long inference times or computationally expensive image processing. I have to use inexpensive processing techniques, because it is not feasible to take more than 500 ms in total for each image. As mentioned above, some companies exist who seem to have robust facial expression recognition techniques, but it is unknown what algorithms they use exactly or how well they perform in general. Furthermore, a dataset has to be used that contains expressions that are only sometimes posed, and most often not exaggerated, as well as photos with relatively bad quality and faces under an angle or bad lighting conditions. Such a dataset would be most similar to Resnap's data, which includes any pictures users may have made and possibly want to add to their photo book.

5 Methods

5.1 Dataset

In this project, a dataset has to be used that has the same conditions as the data the application is used on. I use a recent dataset called AffectNet [31], in which the images are tagged with the categories of facial expressions as well as a continuous scale of valence (positive/negative) and arousal (intensity of the emotion). For example, anger has low valence and high intensity, sadness has low valence and low intensity, and happiness has high valence and a range of intensities. The categorical model includes seven different emotions: 'Anger', 'Contempt', 'Disgust', 'Fear', 'Happy', 'Sad' and 'Surprise'. It also contains a category for the 'Neutral' expression and for 'None' (no expression), 'Uncertain' (not sure about the category of the expression) and 'Non-Face' (images that do not contain a face or are affected with post-processing such as watermarks).

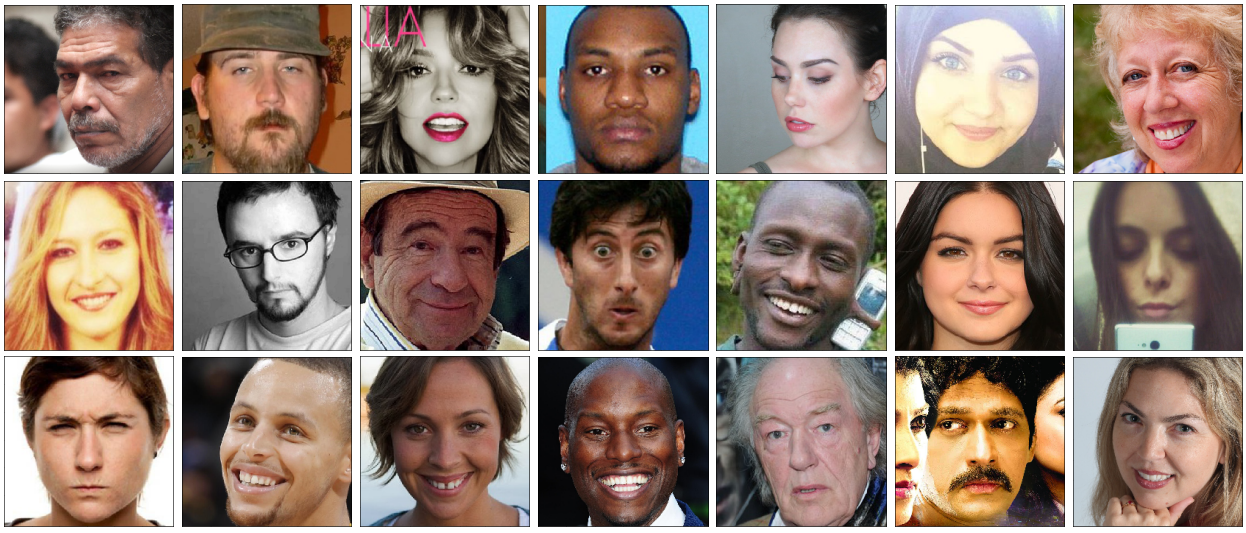


Figure 7: *Excerpt from the AffectNet dataset*

This dataset consists of 420.299 images and is constructed using Google Image Search in different languages, with 1250 keywords, and has been labeled by hand by 12 annotators, with a maximum of 1 person considering the label for each image. The large amount of images is helpful in the training process. The images vary significantly in fidelity, between being posed and non-posed, and between being well- and poorly lit (see Figure 7). Only the photos containing emotions in the facial expressions ('Anger', 'Contempt', 'Disgust', 'Fear', 'Happy', 'Sad', 'Surprise' and 'Neutral') will be used, because these are the only relevant categories.

5.2 Pre-processing

Because good performance in facial expression recognition can be achieved by using images with uniform lighting and posing conditions [11, 13, 42] and the images in the dataset vary in several aspects, some pre-processing techniques are used to make the data more uniform. I will consider the effects of pre-processing on the performance of the trained networks. It has been shown that with these pre-processing techniques, combined with relatively simple (and fast) machine learning, good accuracy can be achieved on face recognition and facial expression recognition [45, 26]. In the meantime, processing time is supposedly not impacted significantly. This allows for both faster and more accurate recognition.

5.2.1 Contrast-limited adaptive histogram equalization (CLAHE)

Histogram equalization normalizes the histogram of an entire image, enhancing the contrast of the image as a whole (see Figure 8b). Adaptive histogram equalization does this per section of the image, increasing contrast in each part of the image. This works better when some areas are much lighter or darker than others, and/or the entire image is bright or dark. This method is again improved by being contrast-limited, stopping enhancement of noise in homogeneous areas, where contrast should not be enhanced significantly. This pre-processing technique enhances the features of faces by enhancing edges and lines,

(a) Original image. (b) Applying CLAHE. (c) Applying rotation correction. (d) Applying spatial normalization.

Figure 8: *Pre-processing steps.(a) Original image. (b) Applying contrast-limited adaptive histogram equalization (CLAHE). (c) Applying rotation correction after CLAHE. (d) Applying spatial normalization after CLAHE and rotation correction.*

allowing for easier recognition of facial features of importance, such as the shape of the eyebrows, eyes, mouth or possibly the appearance of laugh lines [52].

5.2.2 Rotation correction

Facial landmarks can be used to calculate the rotation of the face in the 2D plane and rotate the image so that the line drawn between both eye centers is horizontal (see Figure 8c). This will normalize the rotation of the head in the 2D plane. The first step is detecting facial landmarks. These points on the face are already calculated at Resnap for face recognition. The facial landmarks are used to calculate the eye centers by taking the average of the eye landmarks for each eye. Then, the eye centers are used to calculate the midpoint between them, and the angle that the straight line between them makes with a horizontal line. The image is then rotated with this angle around the midpoint, to have a horizontal line between the eyes in the image.

5.2.3 Spatial normalization

Spatial normalization changes the positioning of a face's features so that for each face, the features such as eyes, nose and mouth are almost always in the same place (see Figure 8d). The differences in facial features between faces will be relatively smaller and the difference in facial expression will be relatively larger. This ensures that expressions can be recognized better [26]. This is done by taking the midpoint calculated during rotation correction and cropping the image to a factor of 1.7 times the inter-eye distance to the sides, 1.3 above the eyes and 3.2 below the eyes. The factors are based on [26], with a larger factor to the sides because of less uniform data. This pre-processing greatly increases the uniformity of the dataset, as can be seen in Figure 9 (Compare to Figure 7).

Figure 9: *Pre-processed excerpt from the AffectNet dataset*

5.3 Network structures

On this dataset various (convolutional) network architectures are trained and their performance is compared. These networks are constructed in Python using Keras[5] (version 2.2.4) with a TensorFlow[6] (version 1.12.0) back-end. Keras is a high-level Python API for neural networks, which can be used on top of TensorFlow to construct, train and do inference with neural networks. TensorFlow itself is an open source software library for numerical computations. The constructed networks are outlined below.

5.3.1 VGG16

VGG16 is a relatively old network trained on the ImageNet dataset and competing for the top spot in the ILSVRC in 2014 [43]. VGG16 uses blocks containing normal 3x3 convolutional layers followed by a pooling layer, to a total of 23 layers. This makes the network architecture straightforward, but also makes the number of parameters (138 million) and file size (528MB) a lot larger than newer networks. This also makes VGG16 take a long time to (re-)train and slow to make an inference. Even if more effort is put into training of VGG16, it is unlikely that it will have a higher accuracy than the newer networks that I use, but it serves as a baseline to compare them against. Despite its age, VGG16 is still widely used for deep-learning tasks involving image classification.

5.3.2 MobileNet

MobileNet was conceived in 2017 [19]. It is a deep and well-performing neural network model, but very light-weight because of the use of convolutional layers called depthwise separable convolutions instead of standard convolutional layers. This network is already used at Resnap for image classification, and while there is a newer version of this network [39], it was found at Resnap that the new version performs worse for

[5] https://keras.io/
[6] https://www.tensorflow.org/

them. The network has many convolutional layers (88) and similar performance on the ImageNet dataset to VGG16, but only has 4,3 million parameters and a size of 16 MB. Instead of doing a convolution over the length, width and number of channels at the same time, these operations are separated into multiple layers. Keep in mind that this inflates the number of layers. First a depthwise convolution is done, applying a single filter to each input channel, keeping the number of input channels (see Figure 10). Afterwards, a pointwise convolution is done, which is a 1x1 convolution that combines the input channels. Both operations are followed by a batch normalization and ReLU layer. A depthwise separable

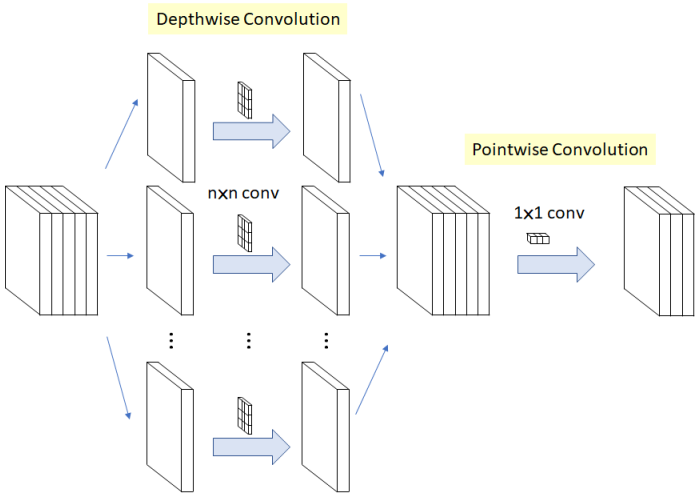


Figure 10: *Example of a depthwise separable convolution in MobileNet, in which a convolution is separated into (1) a depthwise convolution over the channels (one nxn convolution for each channel) and (2) a pointwise 1x1 convolution to change the dimensions of the input.*[7]

convolution decreases accuracy very slightly, but saves tremendously on the number of parameters and multiply-add operations; the authors claim a reduction factor of 8 to 9 for 3x3 convolutions [19]. This reduction stems from the fact that there is no interaction anymore between the number of channels and the number of filters, preventing combinatorial problems. This network is highly applicable to be used at a company due to its high efficiency and good performance, and due to it being fast to train.

5.3.3 Xception

Xception [9] is a newer and improved version of Inception v3 [44], which is a network structure developed by Google over several iterations. It is a large network with 22,9 million parameters, a depth of 126 layers and a size of 88 MB. Like MobileNet, it uses depthwise separable convolutions to achieve this. However, unlike MobileNet, the researchers first perform a pointwise convolution, followed by the depthwise convolution (see Figure 11) and they only use batch normalization and a ReLU layer after the second operation. Xception has a significantly better performance than MobileNet on the ILSVRC, due to the much larger number of parameters and an even more efficient use of its parameters. Due to the file size, this network is too large to be used straightaway by Resnap, but it will likely achieve the best accuracy possible.

[7] https://towardsdatascience.com/review%2Dxception%2Dwith%2Ddepthwise%2Dseparable%2Dconvolution%2Dbetter%2Dthan%2Dinception%2Dv3%2Dimage%2Ddc967dd42568

[8] https://towardsdatascience.com/review-xception%2Dwith%2Ddepthwise%2Dseparable%2Dconvolution%2Dbetter%2Dthan%2Dinception%2Dv3%2Dimage%2Ddc967dd42568

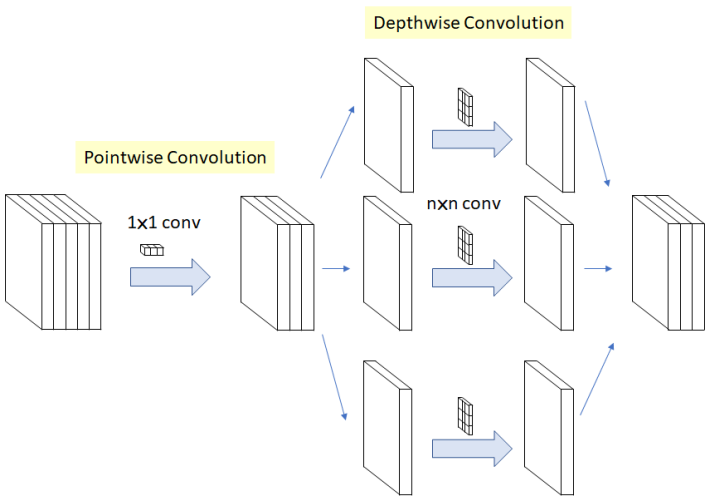


Figure 11: *Example of a new type of depthwise separable convolution in Xception, in which a convolution is separated into (1) a pointwise 1x1 convolution to change the dimensions of the input and (2) a depthwise convolution over the channels (one nxn convolution for each channel).*[8]

5.3.4 Small convolutional network

A very simple network is also trained on the standard and pre-processed data. Following [26], such a network can get a high accuracy (96.76%) on pre-processed, clean data. This network serves to test whether pre-processing the images to be uniform has an equally great performance when the original images are of worse quality. This network has 9 layers, with 367,304 parameters and a size of 4.4 MB. It has two sets of a convolutional layer followed by a pooling layer, followed by flattening, dropout, and dense layers. This simple network is not expected to perform very well on the standard images, due to the quite complex nature of the problem. However, the pre-processed images should be more easily recognizable. This will clearly show the effect pre-processing can have on the accuracy.

5.4 Fine-tuning

The networks outlined above (except for the small network) were originally trained for the ImageNet Large Scale Visual Recognition Competition (ILSVRC) [37] in various years (noted for each network). They were trained for object recognition and can classify up to a thousand categories of objects, meaning their last layer has up to a 1000 nodes. They have been trained extensively to recognize patterns that appear in any of the object categories. The lower levels of these networks are very general patterns, while the higher levels are more specifically geared toward the categories the network was trained on. However, since the number of categories they were trained on is so high, they are also very generally applicable. To allow these networks to specifically recognize facial expressions, they only need to be re-trained for a relatively short period of time [49]. This is called fine-tuning: tweaking the trained network for different classes. Usually, this is done for only the last few layers of the network, and the very last layer of the network is replaced by a layer with a node for each of the new classes. However, the images used for facial expression recognition, containing only faces, contain quite specific patterns. Therefore, the networks will be fine-tuned completely to also update the earlier layers.

5.5 Dealing with class imbalance

Class weights can counterbalance class imbalance in the dataset so that each class is of equal importance to the network. When class weights were not implemented, the networks would only predict the class with the most data points (happy). Class weights for each class i are calculated as $\frac{max(R)}{C \times n_i}$ with R the list of ratios of the data in each class, C the number of classes and n_i the number of samples for class i. As an example, class weights for the different classes in the training set (with 8 different classes) are shown in Table 1.

Expression	**Nr. of images**	**Weight**
Anger	24882	0.6752
Contempt	3750	4.4804
Disgust	3803	4.4180
Fear	6378	2.6343
Happy	134412	0.1250
Neutral	74874	0.2244
Sad	25459	0.6599
Surprise	14090	1.1924

Table 1: *The number of files and the weights for the network for each class.*

The weights for the classes are lower the higher the percentage of samples for a certain class is in the data. For example, the weight of happy is 0.1250 and the weight of sad is 0.6599. This is because there are $0.6599/0.1250 = 5.2795$ times as many samples for happy as there are for sad. The weight balances out the disparity between the number of samples for different classes.

5.6 Visualization of network filters

Using the Python library Lucid[9], it is possible to visualize what the network is looking at to predict classes. Lucid tries to create an optimal input for a neuron. As described earlier (section 3.2), a neuron 'recognizes' a pattern by having a high activation if that pattern occurs in the image (see Figure 5). A neuron that represents a certain class will recognize features that are distinctly present in the images of the class that it was trained on. In the last layer of the network, there is exactly one neuron for each class of facial expression. These neurons will all react to features of the input image that are distinct for their corresponding class. For example, a neuron representing the 'angry' class could represent frowning eyebrows or angry eyes, while a neuron representing 'happy' might show teeth and smiles as they are often not visible in other emotions. Lucid generates a random image and optimizes this image for a neuron over several iterations to maximize the activation of that neuron. I will generate several of these images and study them to use as an indication of what the neurons are focusing on and what makes the classes of facial expressions distinct from one another. This helps determine what to focus on for further research and how pre-processing changes the focus of the networks.

[9] https://github.com/tensorflow/lucid

6 Experiments

6.1 Training parameters

Four different network architectures are trained on the AffectNet dataset: VGG16, MobileNet, Xception and a small and simple convolutional model. To train the networks optimally, several parameters have been determined empirically. For each network, the standard input size was used. For VGG16 and MobileNet, this is 224 by 224 pixels, with 3 channels (RGB). For Xception, this is 299 by 299 pixels with 3 channels. For the small CNN, a very small input size of 32 by 32 pixels by 3 channels was used, based on the paper that inspired this network [26]. Each network is trained with the Adam optimizer [20], with a learning rate of 1×10^{-4}, a factor 10 lower than the standard learning rate, which makes sure the fine-tuning does not remove what the network has previously learned. Loss is calculated using the categorical cross entropy, and the metric to improve is the categorical accuracy. Because of limited GPU memory, a batch size of 32 is used, with each epoch consisting of 256 batches. Each epoch, the images were shuffled and each image had a 50% chance of being horizontally flipped to increase the amount of different data seen by the networks. More image augmentation, such as random cropping or rotation was not possible, because it would lead to more uniformity between standard and pre-processed images, negating the difference. Each network was trained until validation loss no longer decreased.

6.2 Data (pre-)processing

For every network structure, training was done twice. One copy of each network is trained with the standard images from AffectNet, re-sized to the input size. The other copy is trained on the images after they are pre-processed (as described in section 5.2). For each copy of the network, the resulting accuracy and difference between filter visualizations (section 5.6) are compared.

6.3 Optimizing class grouping

As a baseline, the data is grouped into 8 classes: anger, contempt, disgust, fear, happy, neutral, sad and surprise. As can be seen in Table 1, disgust and contempt are very underrepresented in AffectNet. They are also not very common expressions in real life and thus likely not in photo books. Therefore, they may not be very important for Resnap to focus on. Still, in network training they are given the same importance when training with 8 classes. As shown in the results section, they also consistently have the lowest F1 score of all classes. Especially contempt seems hard to classify. Therefore, leaving them out may not be harmful for practical relevance of the network and will likely increase accuracy due to less classes leading to less confusion for the network. Less classes means that the baseline accuracy goes up from $\frac{100}{8} = 12.5\%$ to $\frac{100}{6} = 16.7\%$, so the network has less chance of predicting the wrong class. Apart from the baseline accuracy, the accuracy will likely increase further because of the leftover classes themselves being easier to classify and thus less confusion existing between the classes. MobileNet will be fine-tuned with the 6 remaining classes.

To categorize and select photos, it may be beneficial to group the classes into only 3 larger classes: positive, neutral and negative facial expressions. This can be useful because of the way this information can be used to give photos a score for photo books. A smaller number of classes that have clearer distinctions makes it easier to judge whether a photo should be selected for a photo book or not, based on which class it is in. The proposed grouping of classes is:

- Positive: happy

- Neutral: neutral, surprise
- Negative: anger, contempt, disgust, fear, sad

Very negative surprise can be grouped with fear, while very positive surprise usually indicates extreme happiness. Thus, surprise is grouped with neutral, because, on average, it is a neutral emotion. Again, MobileNet is fine-tuned with these 3 classes, because it is the network that has proven to be applicable in concrete cases.

7 Results

As described in section 6, four networks have been trained on the AffectNet dataset. This section contains the results of network training on standard and pre-processed images, with 8, 6 and 3 classes, as well as pre-processing and inference duration.

7.1 Validation accuracy

The validation accuracy (See Table 2) is in most cases not much higher than the test accuracy (See Tables 3 and 4). This is a good indicator of how much the networks have overfitted on the training set. For VGG16, the validation accuracy is 0.06% lower than the test accuracy without pre-processing and 0.67% higher with pre-processing. For MobileNet, the decrease from validation to test is 1.12% without pre-processing, while it is 8.14% with pre-processing. For Xception, without pre-processing, the difference is 2.30% and with pre-processing it is 2.36%. For the small CNN, the validation accuracy is 0.07% higher than the test accuracy without pre-processing and 0.54% higher with it. This means that the networks are not overfitted on the training set and can generalize well to the test set. It is likely that they will also generalize well to other similar data such as that used by Resnap.

Table 2: *Network accuracy on the validation set for all networks, with standard and pre-processed images.*

Network	Pre-processing	Validation accuracy
VGG16	standard	0.5570
	pre-processing	0.5697
MobileNet	standard	0.6602
	pre-processing	0.7114
Xception	standard	0.6406
	pre-processing	0.7107
Small CNN	standard	0.4790
	pre-processing	0.5620

7.2 Test accuracy

Table 3 shows the top-1 and top-2 accuracies, as well as the precision, recall and F1 score for the networks that are trained with standard images (only re-scaled to their respective input sizes).

Table 3: *Accuracy, precision, recall and F1 score on the test set for all networks, with standard images.*

Network	top-1 accuracy	top-2 accuracy	precision	recall	F1 score
VGG16	0.5776	0.7783	0.74	0.58	0.63
MobileNet	0.6490	0.8568	0.74	0.65	0.68
Xception	0.6176	0.8387	0.73	0.62	0.65
Small CNN	0.4783	0.6409	0.63	0.48	0.53

7.2.1 VGG16

VGG16 has a 57.76% accuracy on the test set without pre-processing. Its top-2 accuracy is 77.83%, an increase of more than 20%. Precision is one of the highest, tied with MobileNet at 0.74 and 0.01 above Xception, but recall is comparatively low at 0.58 resulting in an F1 score of 0.63.

7.2.2 MobileNet

MobileNet has the highest top-1 accuracy of all the networks, with 64.90%. It also achieves the highest top-2 accuracy at 85.68%. Its precision is tied as the highest with VGG16 and its recall is also highest at 0.65. MobileNet's F1 score is the highest at 0.68. Since MobileNet has many fewer parameters than Xception, and a similar architecture, it is surprising that it surpasses Xception in performance. This indicates that the data can already be captured by MobileNet and a larger network with more parameters may not be necessary.

7.2.3 Xception

Remarkably, Xception does not have better top-1 or top-2 accuracy than MobileNet. These metrics are very similar to those of MobileNet, at 61.76% and 83.87%, respectively. Both its precision (0.73) and recall (0.62) are also lower, leading to a lower F1 score of 0.65. Due to the much larger amount of parameters, Xception was expected to have better accuracy because it should be better able to capture features in the data.

7.2.4 Small CNN

The small CNN has the worst performance of all networks, as was expected. Its top-1 accuracy is 47.83% and its top-2 accuracy is 64.09%. This means that its top-2 accuracy is (almost) as good as MobileNet's top-1 accuracy. Its precision comes close to the other networks, at 0.63, but recall is much lower, at 0.48. This results in an F1 score of 0.53. Although this network is small, it still seems quite capable of capturing the specific patterns in the data to classify facial expressions.

7.3 Pre-processing

Training the networks with pre-processed images and doing predictions with images pre-processed in the same way seems to improve the accuracy and F1 score of (most of) the networks. For Xception and the small CNN, the top-1 accuracy and F1 score are higher with pre-processed images than they are without it. For VGG16 and MobileNet, the top-1 accuracy and F1 score are lower with pre-processed images (Compare Tables 3 and 4).

Table 4: *Accuracy, precision, recall and F1 score on the test set for all networks, with pre-processed images.*

Network	top-1 accuracy	top-2 accuracy	precision	recall	F1 score
VGG16	0.5630	0.7840	0.75	0.56	0.62
MobileNet	0.6200	0.8254	0.77	0.62	0.67
Xception	0.6871	0.8751	0.75	0.69	0.71
Small CNN	0.5566	0.7560	0.72	0.56	0.61

7.3.1 VGG16

VGG16 has a worse top-1 accuracy when trained on pre-processed images, although the difference is minimal (57.76% to 56.30%). Its F1 score is also 0.01 lower than it was before. While the precision of the network is slightly higher with pre-processing (0.75), the recall is lower (0.56). However, the top-2 accuracy has increased slightly to 78.40%.

7.3.2 MobileNet

Mobilenet's top-1 accuracy has decreased to 62%, while its top-2 accuracy has decreased to 82.54%. Its precision (0.77) is higher than before, while recall (0.62) and F1 score (0.67) are slightly lower. Its precision is higher than VGG16's and Xception's precision. It no longer has the highest top-1 accuracy, top-2 accuracy or F1 score, because Xception has surpassed it when also trained with pre-processed images.

7.3.3 Xception

With pre-processed images, Xception has the highest top-1 accuracy at 68.71%. Its top-2 accuracy (87.51%) and F1 score (0.71) are also the highest of all networks, between both standard and pre-processed images. Precision (0.75) is the same as that VGG16, but recall is a lot higher (0.69), leading to the best F1 score. The difference between Xception's and MobileNet's performance increases by quite a large margin, as the top-1 accuracy is almost 7% higher than MobileNet's.

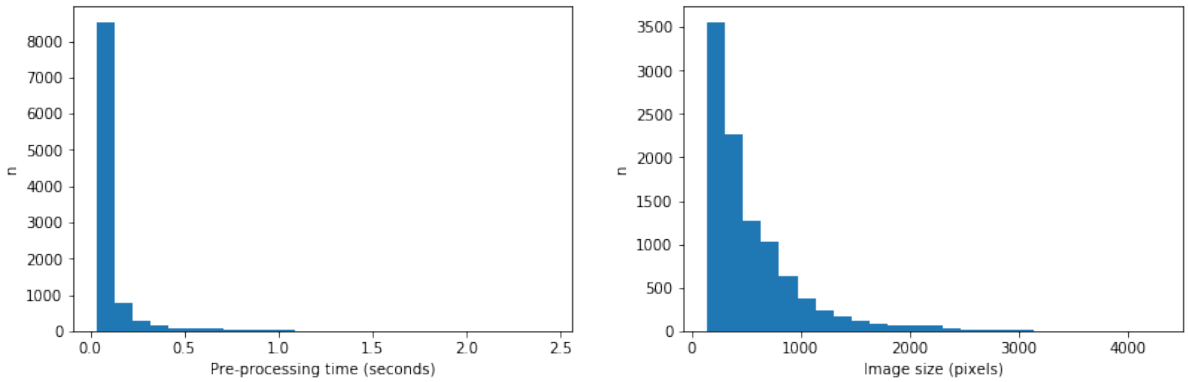
7.3.4 Small CNN

While the small CNN still has the worst performance for each metric, with pre-processing it comes much closer to the other networks, especially VGG16. Its top-1 accuracy is quite good at 55.66%, as is its top-2 accuracy with 75.60%. Precision (0.72) is only slightly lower than the other networks, and while recall (0.56) is much lower than for MobileNet and Xception, it is the same as for VGG16. The F1 score is also very close to that of VGG16. The performance of this small network has increased much more between using standard images and pre-processed images than the other networks.

7.4 Pre-processing time

The time it takes to pre-process an image was calculated by loading in images and pre-processing them, while measuring the duration of this process. This was measured for 10000 images (see Figure 12). The average pre-processing time of these images was 0.0907 seconds, with a standard deviation of 1.270

seconds. The fastest processing time was 0.03175 seconds, while the maximum processing time (on the largest image) was 2.4445 seconds. 79.58% of the images take 0.1 second or shorter to pre-process.



(a) (b)

Figure 12: *Histograms of (a) duration of pre-processing an image in seconds for 10,000 images and (b) image width of these images in pixels. The images were all square. the y-axes show the number of images.*

The size of the images that were pre-processed was also recorded, see Figure 12b. The average size of the images was 548.08 by 548.08 pixels, with a standard deviation of 441.70. The smallest image was 133 by 133 pixels, and the largest image was 4305 by 4305 pixels. 11.58% of the images were larger than 1000 by 1000 pixels. The correlation coefficient between pre-processing time and image size was 0.93, which means that pre-processing time scales with image size.

7.5 Inference time

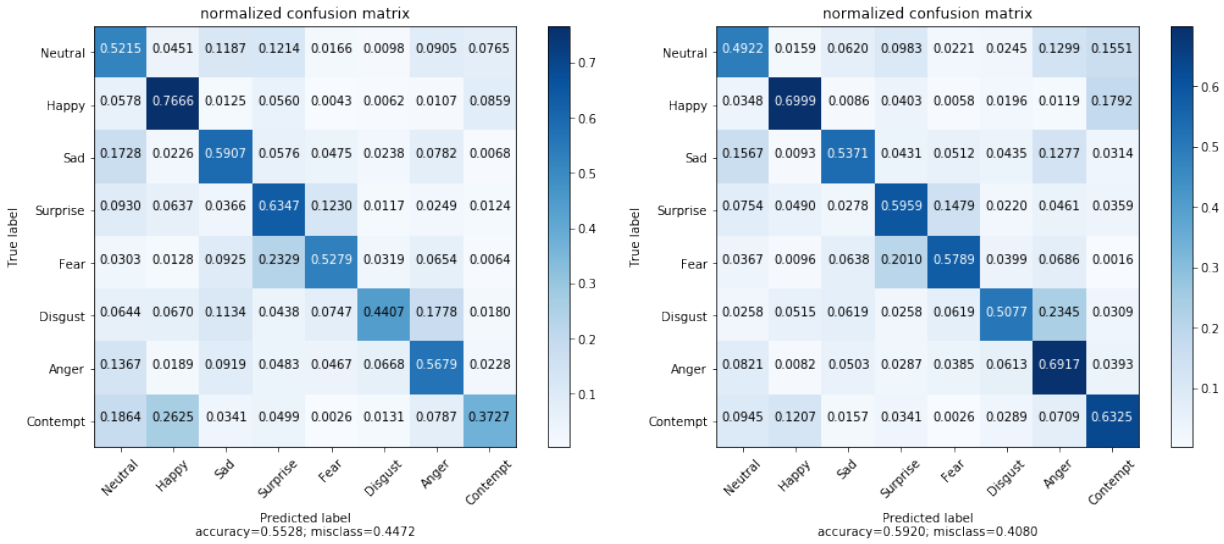
Average inference time is calculated by pre-loading a large batch of images and calculating the time it takes on average for the network to make an inference on each image (see Table 5). Between networks, the inference time varies dramatically. VGG16 and Xception have a relatively similar inference time, although Xception is still 23% slower. Mobilenet is almost a factor 8 faster than VGG16 and a factor 10 faster than Xception. The Small CNN provides an enormous decrease in inference time, being a factor 50 faster than MobileNet, a factor 400 faster than VGG16 and a factor 500 faster than Xception.

Table 5: *Average network inference time per image (in seconds).*

Network	Inference time (seconds)
VGG16	0.3466
MobileNet	0.043535
Xception	0.4263
Small CNN	0.00082165

7.6 Confusion between classes

See the appendix (Section 11) for all confusion matrices, normalized confusion matrices, and classification reports that contain precision, recall and F1 scores for individual expressions. Figure 13 shows the normalized confusion matrices for MobileNet, with standard images and pre-processed images, as examples. For all networks, there is clear confusion between surprise and fear, and when they are wrongly classified, they are much more likely to be classified as each other than as any other class. Happy seems to be the easiest class to predict, with the most correctly classified images.



(a) (b)

Figure 13: *Normalized confusion matrices for MobileNet, trained with (a) standard images and (b) pre-processed images. All confusion matrices can be found in the appendix, section 11.*

For VGG16, confusion between classes is very general. All classes have similar amounts of confusion. However, neutral is the hardest class to predict correctly, both with standard and pre-processed images.

For MobileNet with standard images, disgust and contempt are hard to classify correctly. Images labeled as contempt are classified as happy as often as they are predicted to be contempt. It is remarkable that, with pre-processing enabled, the performance on disgust increases to be twice as accurate. The performance on contempt only improves marginally. Accuracy on sad goes down quite a lot, although precision, recall and the F1 score are higher with pre-processing.

Xception, with standard images, has trouble keeping the neutral facial expression separated from all the others, often predicting each emotion as neutral. When using pre-processed images, this confusion between neutral and all other expressions becomes much less. However, accuracy for contempt, disgust and neutral goes down, while it goes up for the other facial expressions.

Apart from having a low accuracy in general, the major flaw for the Small CNN without pre-processing is classifying disgust. It is much more often classified as anger than as disgust. With pre-processing, this confusion disappears, and misclassification occurs more generally, without a clear pattern. It seems pre-processed images help this network to recognize every facial expression more clearly.

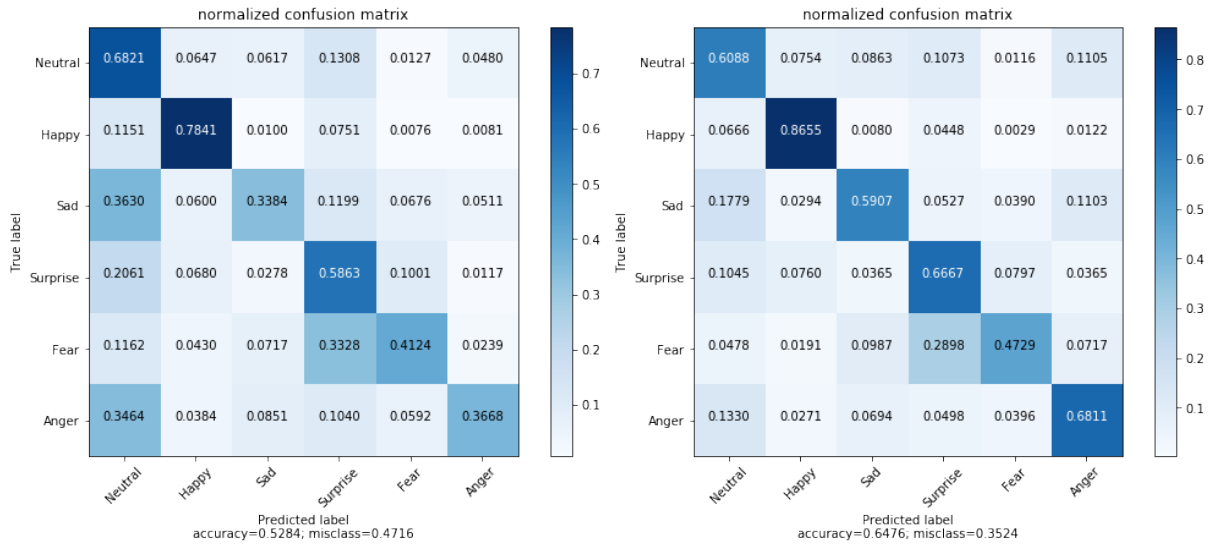
7.7 Optimizing class grouping

7.7.1 6 classes

As described in section 6.3, the facial expressions contempt and disgust do not occur often in the dataset (see Table 1), but from the original confusion matrices with 8 classes (section 7.6), it becomes clear that they still cause some confusion. The classes have been left out here to see how much accuracy improves. Table 6 shows that accuracy for both standard and pre-processed images has increased, with 1.23% and 11.73% respectively. For standard images, precision is worse, while recall has improved, leading to the same F1 score. For pre-processed images, precision is the same, while recall has improved quite significantly, leading to an F1 score of 0.75 with 6 classes as compared to 0.67 with 8 classes. It is, however, hard to say how much of the improvement can be attributed to a clearer distinction between classes, or to just the lower number of classes. From the confusion matrices (Figure 14), it can be seen that there is clearly less confusion between neutral and the other classes when images are pre-processed (compare this to Figure 13).

Table 6: *Mobilenet without disgust and contempt: accuracy, precision, recall and F1 score.*

Metric	standard	pre-processing
Top-1 accuracy	0.6613	0.7373
Precision	0.72	0.77
Recall	0.66	0.74
F1 score	0.68	0.75



(a) (b)

Figure 14: *Normalized confusion matrix for the MobileNet trained with 6 classes on (a) standard images and (b) pre-processed images.*

7.7.2 3 classes

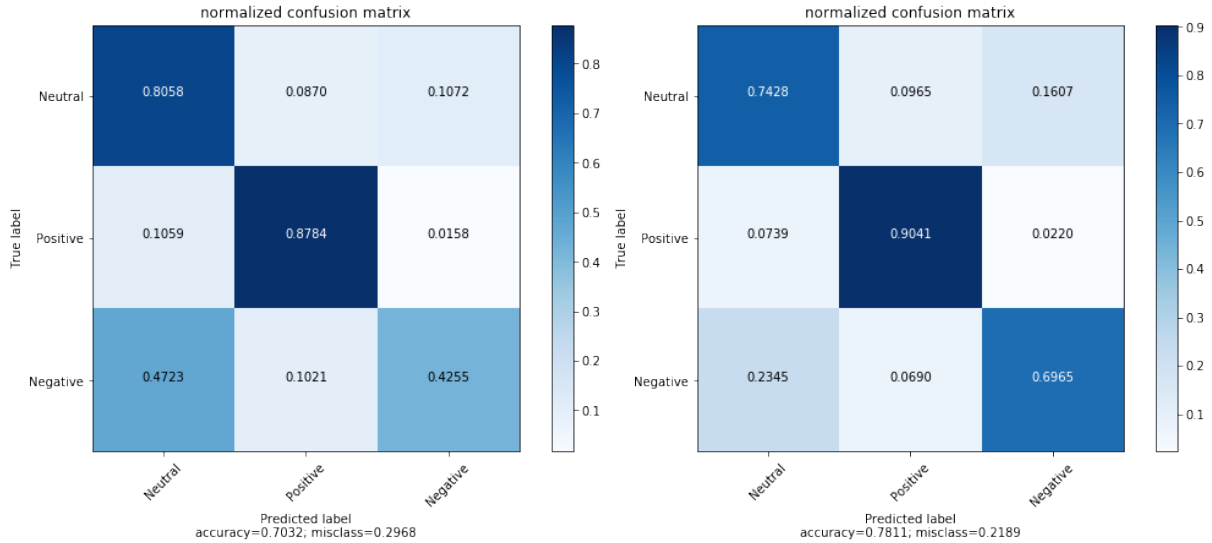
As also described in section 6.3, grouping together the original 8 facial expressions in three broader classes seemed beneficial. Classes are grouped as:

- Positive: happy.
- Neutral: neutral, surprise.
- Negative: anger, contempt, disgust, fear, sad.

Table 7: *Mobilenet with 3 classes (positive, neutral, negative): accuracy, precision, recall and F1 score.*

Metric	standard	pre-processing
Top-1 accuracy	0.7500	0.8085
Precision	0.76	0.81
Recall	0.75	0.81
F1 score	0.74	0.81

Table 7 shows the top-1 accuracy, precision, recall and F1 score for the MobileNet trained on the 3 classes, both for standard and pre-processed images. As expected, the performance of the network is much better on 3 classes than it is on 8. Accuracy for standard images has increased by 10.1%, to 75%. Accuracy for pre-processed images has greatly increased, by 18.85%, to 80.85%. Precision, recall and F1 score have all increased for both types of images. Figure 15 shows the confusion matrix for both



(a) (b)

Figure 15: *Normalized confusion matrix for the MobileNet trained on 3 classes with (a) standard images and (b) pre-processed images.*

standard and pre-processed images. It is clear that negative is confused much less with neutral when using pre-processed images than it is with standard images.

8 Discussion

8.1 Network accuracy and speed

MobileNet and Xception have the best performance out of the four networks, and they are on similar footing in this regard. MobileNet with standard images has better top-1 and top-2 accuracy than Xception with standard images, but Xception has better top-1 and top-2 accuracy than MobileNet with pre-processed images. Their precision, recall and F1 score are also close to each other for both types of images. The fact that both of these networks have approximately the same performance means that anything in the data that allows the networks to distinguish between the different facial expressions, can already be captured by the MobileNet. The large number of parameters of Xception are, apparently, not necessary for facial expression recognition, or they can not be used to their full effectiveness on this dataset. In terms of efficiency, MobileNet is likely the better choice between the two, even for pre-processed images, as its inference time is much faster and its size is much smaller than that of Xception. While Xception does have a higher inference time and a better performance with pre-processed images, its number of parameters and size are much lower than those of VGG16. VGG16 has a reasonable performance, but it is outperformed by MobileNet and Xception in almost every way (it only has slightly higher precision for standard images), which shows its age. For standard images, the small CNN has by far the lowest accuracy and F1 score. Although using pre-processed images grants it the biggest boost in performance out of all networks. It almost equals VGG16's top-1 accuracy, and its F1 score comes very close to that of VGG16 and to the other networks' F1 sore with standard images. Because the Small CNN has by far the fastest inference time and smallest network size, the best network in terms of efficiency for this dataset must lie somewhere in between a small CNN and MobileNet. It might be possible, with some trade-offs to inference time and network size, to improve the simple model to be more accurate, while remaining smaller than MobileNet. This is another sign that the difference between the classes in this dataset can be captured by a simple network, which bodes well for use in an online application such as that employed by Resnap.

8.2 Pre-processing: increase in accuracy

Pre-processing provides a significant ($p<.001$, tested with a chi-square test) increase in performance for the networks for which an increase in accuracy was observed (Xception and the Small CNN). However, the decrease in performance for MobileNet and VGG16 was also significant ($p<.001$. For every network, precision increased. While recall improved for Xception and the small CNN, it decreased for VGG16 and MobileNet. Pre-processing images improves precision, but often decreases performance on recall. It seems training with pre-processing makes the networks more conservative in their classification, creating less false positives, but more false negatives.

VGG16's top-1 accuracy is lower with pre-processed images, while its top-2 accuracy is higher in the same scenario. It is possible that VGG16's training did not go as well with pre-processed images as it did with standard images, or there is some interaction between VGG16's structure and performance on pre-processed images. The same can be said for MobileNet. It is hard to pinpoint why these networks' performances are worse.

Most of the differences in network performance are small, at only a couple of percentage points between the different types of images. The largest increase in accuracy is shown in the small CNN, which had the lowest accuracy to begin with. The top-1 accuracy of this network increased by 7.83% and its top-2 accuracy increased by 11.51%. The combination of CLAHE, rotation correction and spatial normalization significantly benefits the ability of this network to classify facial expressions in certain scenarios. This

counts for Xception as well. However, since not all networks improved in performance, it is hard to definitively conclude whether pre-processing helps in the classification of facial expressions in general.

8.3 Pre-processing: duration

As described in section 7.4, pre-processing can take a very long time, for very large images. On smaller images, it does not take very long per image (on average about 0.1 seconds). Because the images at Resnap are resized to 224 by 224 before doing any inference, pre-processing will be on the lower bound of the spectrum of pre-processing times. The time it takes to pre-process is not very long compared to the inference time of Xception and VGG16, but it would be a considerable speed sacrifice to use pre-processed images for MobileNet or the small CNN Pre-processing takes longer than these networks' inference times. For the small network, certainly, pre-processing is not necessarily better to use, even if the small CNN's performance is good enough. Pre-processing takes about 1000 times the inference time to do.

8.4 Confusion between classes

Disgust and contempt are the hardest classes to classify correctly (see the classification reports in the Appendix, section 11). They have the lowest F1 scores for every network, for both pre-processed and standard images. These emotions are also hard to classify for humans. In a laboratory-constructed data set, it may be easy to recognize these expressions, but in the AffectNet data set, the classes are not always clearly cut for the human labelers (as can be seen in section 8.5). Besides being very hard facial expressions to classify for observers, leading to additional confusion, disgust and contempt are also very underrepresented in the data set, which makes it harder for the networks to get enough examples of what they actually look like. Contempt is most often misclassified as happy, while disgust is most often misclassified as anger.

The confusion between fear and surprise is the largest and most consistent between networks. After disgust and contempt, they are the hardest classes to classify correctly for the networks. They are also most often classified as each other. Intuitively, this makes sense, given that fear is a more negative form of surprise in terms of valence and arousal, but is otherwise very close to it. When it is not clear how negative the expression is, different labelers use different (and confusing) labels for the same expression.

Without pre-processing, for all networks, many images are often misclassified as neutral. This makes sense, given that every other facial expression in toned down form could be tagged as the 'neutral' expression. The border between a neutral expression and an 'expressive' expression may well have been different for different annotators, leading to inconsistency in labeling images that were on the border of neutral and the other expressions. With pre-processing, the confusion is still visible in the normalized confusion matrices, but it is much less. Maybe, the strength of the expression is amplified by the pre-processing, creating a clearer notion of the difference between an actual expression and weak expression. It seems that the line between a neutral expression and the other expressions is now more clearly recognized by the networks in accordance with the way the annotators saw the difference. Happy was clearly the easiest class to recognize for all networks, and its F1 score is consistently the highest. This also makes sense given that the other expressions are more negative and happy has the highest amount of images associated with it. This makes it much easier to distinguish from the other emotion classes.

8.4.1 Optimizing class grouping

Since disgust and contempt were so hard to classify and may be the least relevant classes for Resnap, MobileNet was fine-tuned again with the remaining 6 classes instead of 8. As can be seen from the results section, the accuracy improved slightly. It remains hard to say if this is because of the increased distinction between classes, or because there is inherently less confusion because of the lower number of classes. Since the number of images of disgust and contempt was comparatively low, their removal can not have a significant direct impact on the overall accuracy of the network. Clearly, some of the same patterns remain as with 8 classes. Happy remains the easiest class to recognize. Fear and surprise remain confusing, and many images are still misclassified as neutral. The confusion with neutral is again decreased with pre-processed images. Whether using 6 classes is better to do in practice depends wholly on the importance that is given to contempt and disgust.

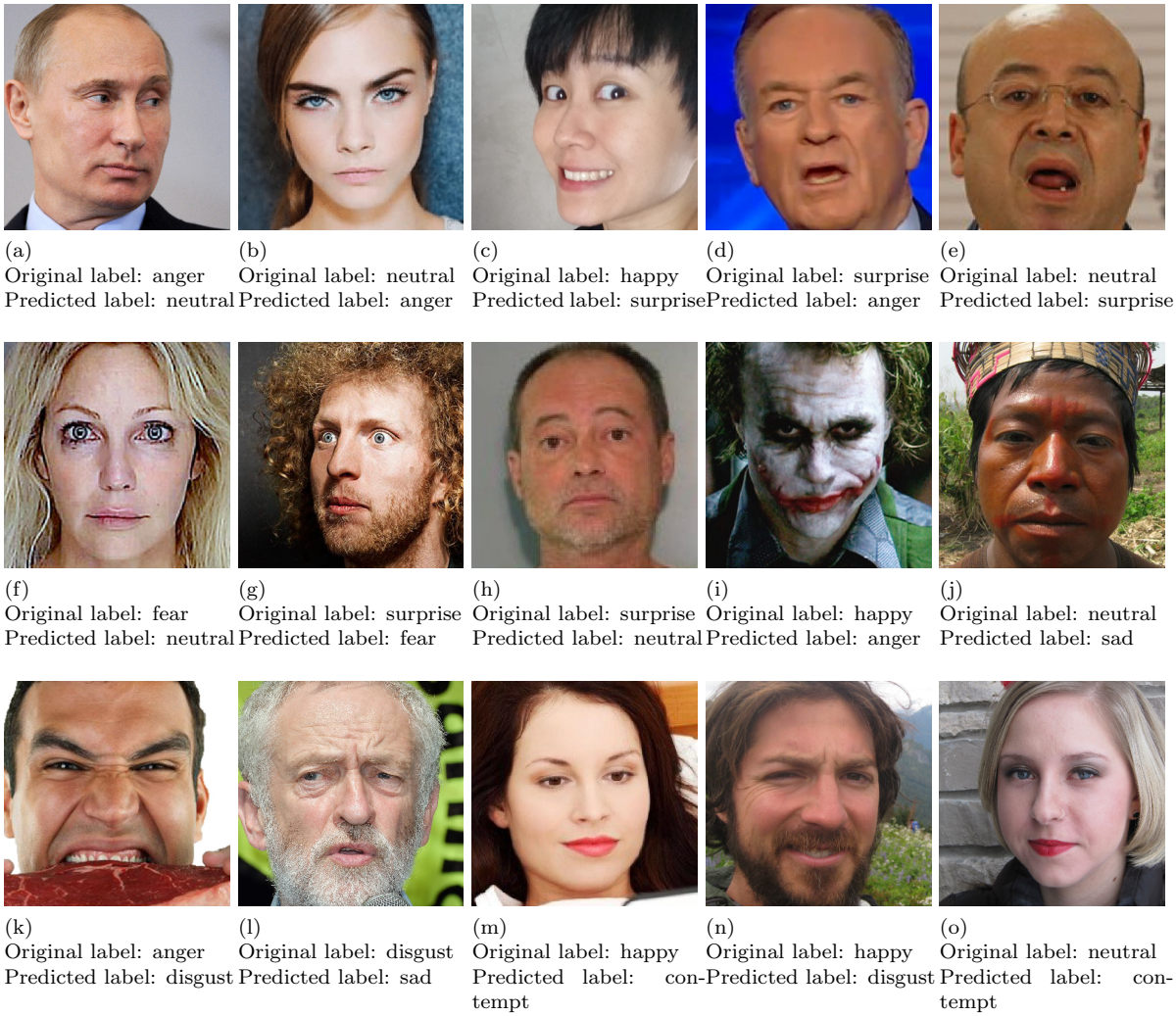
Using 3 categories improves accuracy much more. Of course, choosing the correct class given some knowledge of the classes' features is much easier now because of the lower number of classes. Some of the earlier confusion with more classes can still be seen. With standard images, a high percentage of non-neutral images are classified as neutral, while they are not with pre-processed images. The confusion between the negative and the neutral class may be amplified by the fact that surprised and fear images are in the different classes. This makes the classes less distinct. Happy remains the easiest facial expression to classify. The increased simplicity means that grouping the facial expressions into 3 classes can be a great way to improve accuracy in the actual application of face expression recognition at Resnap.

8.5 Wrongly classified images

The confusion between certain classes can be easily justified (intuitively) when looking at some of the wrongly classified images themselves, examples of which are shown in Figure 16. When looking at predictions for images that are wrong, the predictions of the network and the original label often make sense. This may be due to the limitations of the dataset, whose labels are inherently confusing. Arguably, image 16b contains an angrier expression than image 16a. Still, 16b is labeled as neutral, while 16a is labeled as angry. These images are clearly on the border between angry and neutral, and many facial expressions could be classified as in-between 2 emotions. The border between classes is not clear for the network, because it is not clear for a human either. Of course, these observations are based on conjecture, as not all images could be observed. The images were already classified by professionals, and the problem of confusion between classes is inherent to the problem of facial expression recognition. Although the network predictions are not the same as those of the human labelers, the alternative label suggested by the network is justifiable for every image in Figure 16.

8.6 Visualization of network filters

To get an understanding of what parts of the images the network is actually looking at and why classes are confusing, the optimal input for some MobileNet neurons was visualized, as described in section 5.6. This was done for MobileNet, with 8 facial expression classes, with standard and pre-processed images, as well as for MobileNet with 3 classes (positive, neutral and negative). Figure 5 also shows that earlier layers show concrete features like eyes, lips and teeth.



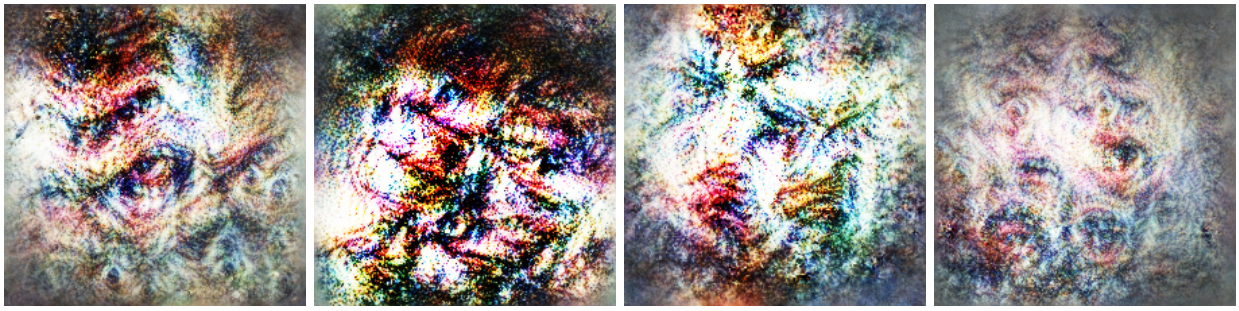
(a) Original label: anger Predicted label: neutral
(b) Original label: neutral Predicted label: anger
(c) Original label: happy Predicted label: surprise
(d) Original label: surprise Predicted label: anger
(e) Original label: neutral Predicted label: surprise
(f) Original label: fear Predicted label: neutral
(g) Original label: surprise Predicted label: fear
(h) Original label: surprise Predicted label: neutral
(i) Original label: happy Predicted label: anger
(j) Original label: neutral Predicted label: sad
(k) Original label: anger Predicted label: disgust
(l) Original label: disgust Predicted label: sad
(m) Original label: happy Predicted label: contempt
(n) Original label: happy Predicted label: disgust
(o) Original label: neutral Predicted label: contempt

Figure 16: *Examples of wrongly classified images from MobileNet with 8 classes and standard images. Altough the network predictions are not the same as those of the human labelers, the alternative label suggested by the network is justifiable for each image.*

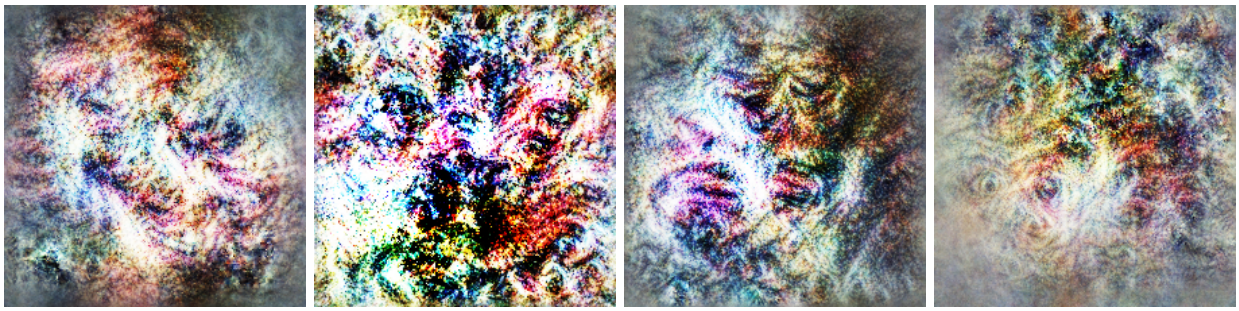
8.6.1 8 classes, standard images

Figure 17 shows examples of the generated optimal inputs for each neuron of MobileNet's last layer when trained on standard images. Since the accuracy of the network is not very high and the images are randomly initialized, the results are not perfect. However, after studying many of these images for some time, some patterns can be noticed that correspond to the classes each neuron represents.

Figure 17a, representing anger, shows one clear, very angry-looking eye in the middle of the image, as well as many v-shapes. This v-shape can also be found in other visualizations of the class. It is likely this shape represents the middle between the eyebrows, which is a clear v-shape in an archetypal angry person. Contempt (17b) also shows an eye that looks angry (in the bottom middle), as well as sets of



(a) Anger (b) Contempt (c) Disgust (d) Fear



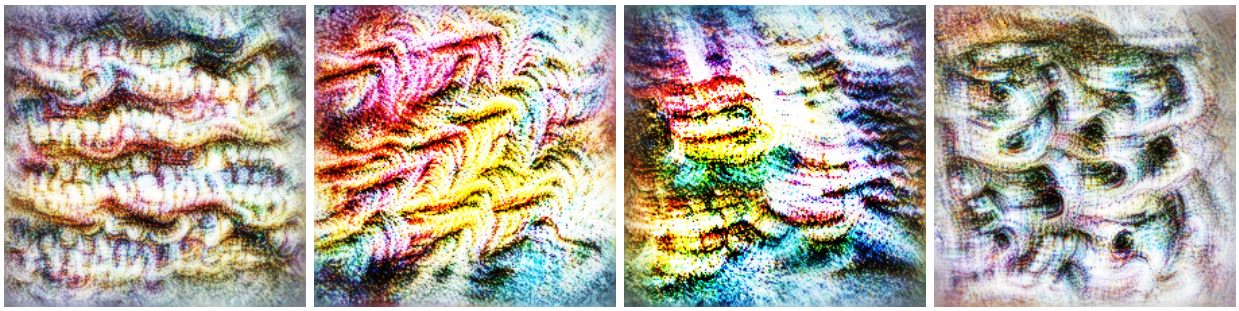
(e) Happy (f) Neutral (g) Sad (h) Surprise

Figure 17: *Visualization of the optimal input for the MobileNet neurons in the final layer (as described in section 5.6). This MobileNet was trained with standard images. Each neuron corresponds to one of the 8 classes: (a) Anger, (b) Contempt, (c) Disgust, (d) Fear, (e) Happy, (f) Neutral, (g) Sad and (h) Surprise.*

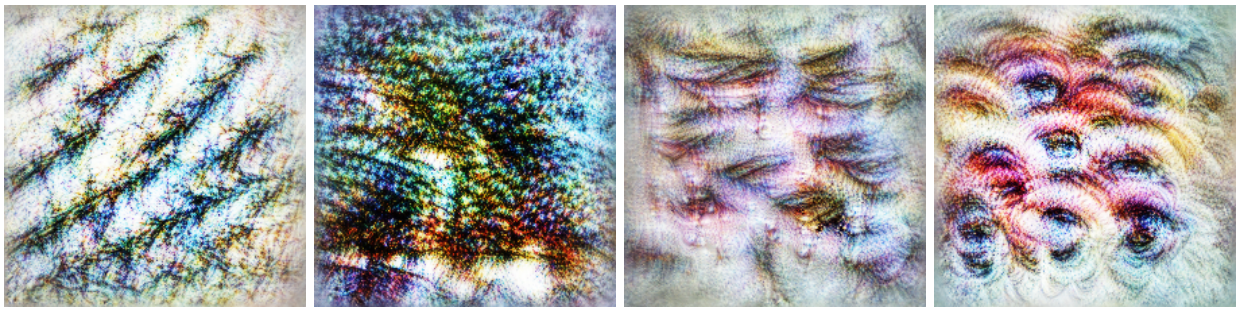
bared teeth and with lips. Disgust (17c) shows the same v-shapes as anger does, but with closed eyes, together with lips that are pressed together. This may be hard to see. The similarity in the v-shapes explains some of the confusion the networks may have between anger and disgust. The generated optimal input for fear (17d) shows neutral-looking open mouths, which are sets of an upper lip, teeth, a tongue and a lower lip. One fearful-looking eye can be seen towards the top-left. Fearful people most often have an open mouth and wide-open eyes. Happy 17e quite clearly shows grinning mouths: a red line curving upward, then a row of white teeth and another red curving line below it. The fact that both happy and contempt have show teeth in the visualization explains some of the confusion between these two classes. For neutral (17f), it is very hard to say anything about the image. It seems like eyes are showing in the image, but not much else can be seen. This makes sense given that all classes in less expressive form could be classified as neutral, and their expressions are not clear. This means that there is no uniform pattern to these images. Sad (17g) shows some patterns which could be recognized as sad faces, with droopy mouths and eyebrows curving upwards. The clearest thing that can be seen in image 17h that represents surprise are wide-open eyes, some with raised eyebrows (especially to the left in the middle of the image). This would be logical, because it is one of the clear distinguishing features of surprise. Of course, fearful people also show raised eyebrows and can have their eyes and mouth wide open. It seems the MobileNet has learned that images with an open mouth should be classified as fear and images showing wide-open eyes are most often labeled as surprise. Of course, if this is true, it will often lead to a wrong prediction, if a surprised person clearly shows an open mouth or a fearful person clearly shows wide-open eyes. From this visualization, it can already be concluded that the networks do not necessarily

look at the all features from a class that seem important to humans. They specifically seem to focus on the feature(s) from each class that uniquely distinguish the class from the other classes, not the entire face. This means that for some classes, the network only specifically looks at one feature, such as a wide, smiling mouth (curving upwards) with bare teeth for the happy class. If an image has this feature, the image is immediately very likely to be classified as happy. Intuitively, this makes sense, because in almost every other class, teeth are barely visible. Only for anger are teeth also often visible. However, an angry person has a different clear distinguishing feature, namely strongly frowning eyebrows which creates a clear v-shape in between the eyebrows.

8.6.2 8 classes, pre-processed images



(a) Anger (b) Contempt (c) Disgust (d) Fear



(e) Happy (f) Neutral (g) Sad (h) Surprise

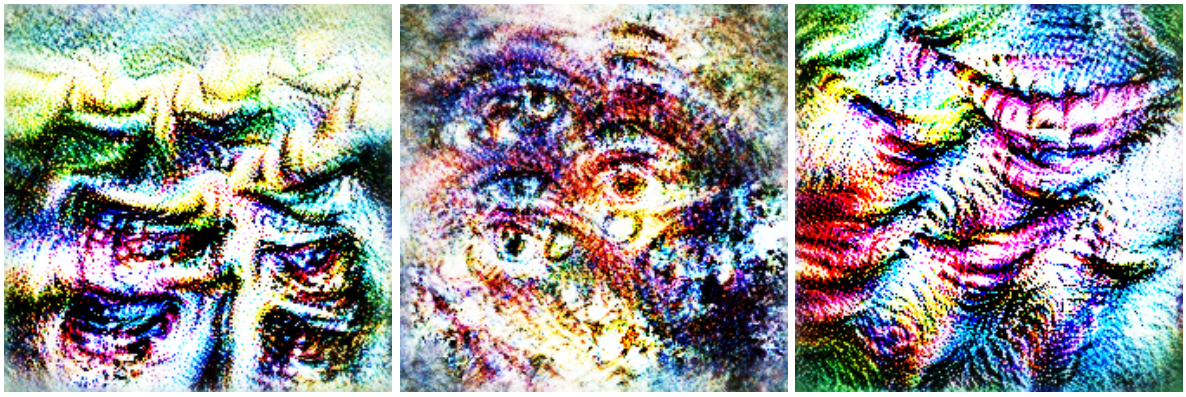
Figure 18: *Visualization of the optimal input for the MobileNet neurons in the final layer (as described in section 5.6). This MobileNet was trained with pre-processed images. Each neuron corresponds to one of the 8 classes: (a) Anger, (b) Contempt, (c) Disgust, (d) Fear, (e) Happy, (f) Neutral, (g) Sad and (h) Surprise.*

For the MobileNet trained on pre-processed images, the optimal inputs are much less clear than for the MobileNet trained on standard images. Anger shows rows of teeth, and some of the v-shape can be seen. Disgust also shows patterns resembling teeth. Fear and surprise both show eyes, but this is much more clear for surprise. These eyes are very clear and wide open. Sad shows lines that could represent eyebrows. The other images do not show patterns that are recognizable as real-life facial features (although the pattern seen in contempt might represent one raised eyebrow as this is often present in contempt images). The visualization does show very different and distinct patterns for each class. Apparently these patterns are important for the network to recognize these classes. Of course,

this network was trained on pre-processed images, which sometimes look less realistic than the standard images, with squashed proportions and amplified features.

8.6.3 3 classes

In the visualization with 3 classes (see figure 19), the differences between classes are more exaggerated and more clear. The negative classes (anger, contempt, disgust, fear and sad) all have one thing in common: the frowning eyebrows (see Figure 19a). The grouping of neutral and surprise (19b) has led to neutral looking, but wide open eyes in the filter visualization. The distinguishing factor for positive facial expressions (19c) (including only happy) is clearly the smiling mouth with bare teeth.



(a) Negative (b) Neutral (c) Positive

Figure 19: *Visualization of the optimal input for the MobileNet neurons in the final layer (as described in section 5.6). Each neuron corresponds to one of the 3 classes: (a) Negative (including Anger, Contempt, Disgust, Fear, Sad), (b) Neutral (including Neutral, Surprise) and (c) positive (including only Happy).*

9 Future work

Since the MobileNet and CNN showed a trade-off between speed and accuracy, the ideal network for this problem likely lies somewhere in between. As a follow-up a new model that lies somewhere in between MobileNet and the small CNN that was used could be constructed, with a faster inference time than MobileNet, but with similar accuracy and F1 score. MobileNet has been constructed with two simple global hyperparameters that can be tuned to change the network's latency and accuracy [19]. A good starting point would be tuning MobileNet with these hyperparameters. It could lead to a model that is both fast and accurate enough.

For the categorical model, the data was quite inconsistently labeled. This is of course somewhat inherent to the problem of facial expressions. However, it would help to have a dataset that was labeled by multiple people. It would likely create a clearer distinction between classes. It would also help to have more data on the less frequently occurring classes in the dataset. Contempt and disgust are by themselves already hard expressions to judge, but the lack of data in these classes makes them even harder to judge for a neural network. Construction of an even more expansive dataset could help neural network performance. Another idea that might help in the future is to combine lab-controlled and 'in the wild' data sets.

Adding the Cohn-Kanade or Jaffe data sets could provide the network with some very clear examples of the archetypal, exaggerated expression. These datasets are small, however, and the expressions in them do not truly represent real expressions. Still, it could be that the neural networks trained with such data would be directed more towards better recognition of the distinct classes.

Some additional techniques (such as those mentioned in [48]) could be applied to increase the networks' ability to distinguish the classes from each other. One such technique, mix-up [50], could be applied during training to make the line between classes clearer, as it increases the robustness of the network to wrong labels and adversarial examples. This could help especially well to solve much of the confusion between fear and surprise.

Another idea to better train the model for recognition of emotions could be to leave out the neutral expression. This would clear up the confusion between the neutral class and the other classes. One would either have to re-label all images containing the neutral expression to one of the other classes that is closest to its expression, or entirely remove the class and its images from training. The first scenario would create a lot of ambiguity in labeling: when a person has clearly no expression, it is equally close to multiple other expressions. Which expression should then be the label? The second scenario would mean that a value is missing for classification of some images. Faces with neutral expressions would be mostly classified based on random guesses. And still, the network would be forced to make a decision. In reality, when deciding whether to select a photo for a photo album or not, a neural network should not be forced to categorize a neutral photo into one of the other categories. The facial expression value could be mostly ignored if it is neutral.

A better idea in this case would be to use the valence and arousal values provided in the AffectNet dataset and train a regression model on this continuous data. The valence value allows for a good metric to classify images as being positive, neutral, or negative, or even make photo selection continuously scale with the valence and arousal values in the image. Still, the value of valence and arousal in different facial expressions remains as subjective as the classification of their emotion class. Here again the limited amount of labelers per image may prove to be a difficulty.

Pruning could help to reduce network size and inference time after training, while retaining the network's performance. During this process, networks are compressed by pruning unnecessary connections. A pruned MobileNet or even Xception could possibly be much faster and still accurate, as unnecessary connections within the networks are removed. Combined with other methods such as trained quantization and Huffman coding, the size and inference time can be lowered [18].

10 Conclusion

From the results in this project, I conclude that it is possible to create efficient facial expression recognition with substantial constraints on image processing duration. Especially MobileNet is suited for this purpose. Classification performance was quite good, and is usable in practice, but it does leave something to be desired. AffectNet is a hard dataset to train a categorical model on. Whereas lab-controlled datasets have clear borders between different classes, AffectNet does not. The dataset itself seems well-made, but the nature of the ambiguity in emotions creates much confusion between classes for the networks. This confusion seems to be inherent to facial expressions themselves, which can be said to be on a continuous scale. It is hard to definitively conclude whether pre-processing helps in the classification of facial expressions in general. It does seem to affect a very simple model positively. More research is necessary to truly capture the essence of (the continuous spectrum of) all possible facial expressions, and with it the facial expressions in AffectNet and the data used by Resnap. A network that lies somewhere in between the MobileNet and small CNN used in this project is likely ideal for this purpose. Additional

techniques could be used as well to improve network accuracy and speed. However, ground-work has been laid to classify facial expressions for photo selection, according to the universal emotions [14] using a hard data set, simple pre-processing and efficient network structures.

References

[1] Affectiva, 2018. https://www.affectiva.com/.

[2] Emotient, 2018. https://imotions.com/emotient/.

[3] Emotiw challenge, 2018. https://sites.google.com/site/emotiwchallenge/.

[4] Face++, 2018. https://www.faceplusplus.com/.

[5] Microsoft azure, 2018. https://azure.microsoft.com/en-gb/services/cognitive-services/emotion/.

[6] Emad Barsoum, Cha Zhang, Cristian Canton Ferrer, and Zhengyou Zhang. Training deep networks for facial expression recognition with crowd-sourced label distribution. In *Proceedings of the 18th ACM International Conference on Multimodal Interaction*, pages 279–283. ACM, 2016.

[7] Marian Stewart Bartlett, Gwen Littlewort, Mark Frank, Claudia Lainscsek, Ian Fasel, and Javier Movellan. Recognizing facial expression: machine learning and application to spontaneous behavior. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 2, pages 568–573. IEEE, 2005.

[8] H Chan and WW Bledsoe. A man-machine facial recognition system: some preliminary results. *Panoramic Research Inc., Palo Alto, CA, USA1965*, 1965.

[9] François Chollet. Xception: Deep learning with depthwise separable convolutions. *arXiv preprint*, pages 1610–02357, 2017.

[10] Ira Cohen, Nicu Sebe, Ashutosh Garg, Lawrence S Chen, and Thomas S Huang. Facial expression recognition from video sequences: temporal and static modeling. *Computer Vision and image understanding*, 91(1-2):160–187, 2003.

[11] Prudhvi Raj Dachapally. Facial emotion detection using convolutional neural networks and representational autoencoder units. *arXiv preprint arXiv:1706.01509*, 2017.

[12] Abhinav Dhall, Roland Goecke, Simon Lucey, and Tom Gedeon. Static facial expression analysis in tough conditions: Data, evaluation protocol and benchmark. In *Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on*, pages 2106–2112. IEEE, 2011.

[13] Dan Duncan, Gautam Shine, and Chris English. Facial emotion recognition in real time, 2016.

[14] Paul Ekman and Erika L Rosenberg. *What the face reveals: Basic and applied studies of spontaneous expression using the Facial Action Coding System (FACS)*. Oxford University Press, USA, 1997.

[15] E Friesen and P Ekman. Facial action coding system: a technique for the measurement of facial movement. *Palo Alto*, 1978.

[16] Wenfei Gu, Cheng Xiang, YV Venkatesh, Dong Huang, and Hai Lin. Facial expression recognition using radial encoding of local gabor features and classifier synthesis. *Pattern Recognition*, 45(1):80–91, 2012.

[17] Hatice Gunes and Maja Pantic. Automatic, dimensional and continuous emotion recognition. *International Journal of Synthetic Emotions (IJSE)*, 1(1):68–99, 2010.

[18] Song Han, Huizi Mao, and William J Dally. Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. *arXiv preprint arXiv:1510.00149*, 2015.

[19] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017.

[20] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[21] Agata Kołakowska, Agnieszka Landowska, Mariusz Szwoch, Wioleta Szwoch, and Michal R Wrobel. Emotion recognition and its applications. In *Human-Computer Systems Interaction: Backgrounds and Applications 3*, pages 51–62. Springer, 2014.

[22] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[23] Oliver Langner, Ron Dotsch, Gijsbert Bijlstra, Daniel HJ Wigboldus, Skyler T Hawk, and AD Van Knippenberg. Presentation and validation of the radboud faces database. *Cognition and emotion*, 24(8):1377–1388, 2010.

[24] Sing Lau. The effect of smiling on person perception. *The Journal of Social Psychology*, 117(1):63–67, 1982.

[25] Joseph E LeDoux. Emotion, memory and the brain. *Scientific American*, 270(6):50–57, 1994.

[26] André Teixeira Lopes, Edilson de Aguiar, Alberto F De Souza, and Thiago Oliveira-Santos. Facial expression recognition with convolutional neural networks: coping with few data and the training sample order. *Pattern Recognition*, 61:610–628, 2017.

[27] Patrick Lucey, Jeffrey F Cohn, Takeo Kanade, Jason Saragih, Zara Ambadar, and Iain Matthews. The extended cohn-kanade dataset (ck+): A complete dataset for action unit and emotion-specified expression. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2010 IEEE Computer Society Conference on*, pages 94–101. IEEE, 2010.

[28] Michael J Lyons, Shigeru Akamatsu, Miyuki Kamachi, Jiro Gyoba, and Julien Budynek. The japanese female facial expression (jaffe) database. In *Proceedings of third international conference on automatic face and gesture recognition*, pages 14–16, 1998.

[29] Masakazu Matsugu, Katsuhiko Mori, Yusuke Mitari, and Yuji Kaneda. Subject independent facial expression recognition with robust face detection using a convolutional neural network. *Neural Networks*, 16(5-6):555–559, 2003.

[30] Gerald R McDermott. *Seeing God: Jonathan Edwards and Spiritual Discernment*. Regent College Publishing, 1996.

[31] A. Mollahosseini, B. Hasani, and M. H. Mahoor. Affectnet: A database for facial expression, valence, and arousal computing in the wild. *IEEE Transactions on Affective Computing*, PP(99):1–1, 2017.

[32] Maja Pantic and Leon JM Rothkrantz. Expert system for automatic analysis of facial expressions. *Image and Vision Computing*, 18(11):881–905, 2000.

[33] Rajat Raina, Anand Madhavan, and Andrew Y Ng. Large-scale deep unsupervised learning using graphics processors. In *Proceedings of the 26th annual international conference on machine learning*, pages 873–880. ACM, 2009.

[34] Henry A Rowley, Shumeet Baluja, and Takeo Kanade. Neural network-based face detection. *IEEE Transactions on pattern analysis and machine intelligence*, 20(1):23–38, 1998.

[35] David E Rumelhart, Geoffrey E Hinton, Ronald J Williams, et al. Learning representations by back-propagating errors. *Cognitive modeling*, 5(3):1, 1988.

[36] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015.

[37] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015.

[38] Ashok Samal and Prasana A Iyengar. Automatic recognition and analysis of human faces and facial expressions: A survey. *Pattern recognition*, 25(1):65–77, 1992.

[39] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. *arXiv preprint arXiv:1801.04381*, 2018.

[40] Neeta Sarode and Shalini Bhatia. Facial expression recognition. *International Journal on computer science and Engineering*, 2(5):1552–1557, 2010.

[41] Caifeng Shan, Shaogang Gong, and Peter W McOwan. Facial expression recognition based on local binary patterns: A comprehensive study. *Image and Vision Computing*, 27(6):803–816, 2009.

[42] Muhammad Hameed Siddiqi, Maqbool Ali, Mohamed Elsayed Abdelrahman Eldib, Asfandyar Khan, Oresti Banos, Adil Mehmood Khan, Sungyoung Lee, and Hyunseung Choo. Evaluating real-life performance of the state-of-the-art in facial expression recognition using a novel youtube-based datasets. *Multimedia Tools and Applications*, 77(1):917–937, 2018.

[43] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[44] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826, 2016.

[45] Yaniv Taigman, Ming Yang, Marc'Aurelio Ranzato, and Lior Wolf. Deepface: Closing the gap to human-level performance in face verification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1701–1708, 2014.

[46] Michel F Valstar, Bihan Jiang, Marc Mehu, Maja Pantic, and Klaus Scherer. The first facial expression recognition and analysis challenge. In *Automatic Face & Gesture Recognition and Workshops (FG 2011), 2011 IEEE International Conference on*, pages 921–926. IEEE, 2011.

[47] Paul Viola and Michael J Jones. Robust real-time face detection. *International journal of computer vision*, 57(2):137–154, 2004.

[48] Junyuan Xie, Tong He, Zhi Zhang, Hang Zhang, Zhongyue Zhang, and Mu Li. Bag of tricks for image classification with convolutional neural networks. *arXiv preprint arXiv:1812.01187*, 2018.

[49] Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. How transferable are features in deep neural networks? In *Advances in neural information processing systems*, pages 3320–3328, 2014.

[50] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2017.

[51] Xing Zhang, Lijun Yin, Jeffrey F Cohn, Shaun Canavan, Michael Reale, Andy Horowitz, Peng Liu, and Jeffrey M Girard. Bp4d-spontaneous: a high-resolution spontaneous 3d dynamic facial expression database. *Image and Vision Computing*, 32(10):692–706, 2014.

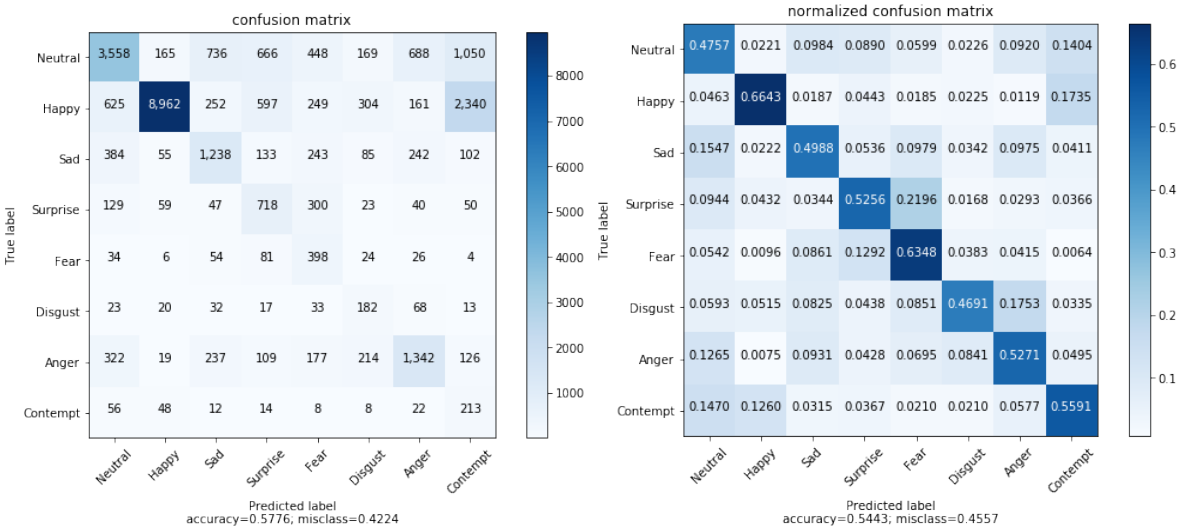
[52] Yisu Zhao, Nicolas D Georganas, and Emil M Petriu. Applying contrast-limited adaptive histogram equalization and integral projection for facial feature enhancement and detection. In *Instrumentation and Measurement Technology Conference (I2MTC), 2010 IEEE*, pages 861–866. IEEE, 2010.

11 Appendix

11.1 VGG16

Table 8: VGG16 standard images classification report.

	precision	recall	F1 score	Support
Neutral	0.69	0.48	0.56	7480
Happy	0.96	0.66	0.79	13490
Sad	0.47	0.50	0.49	2482
Surprise	0.31	0.53	0.39	1366
Fear	0.21	0.63	0.32	627
Disgust	0.18	0.47	0.26	388
Anger	0.52	0.53	0.52	2546
Contempt	0.05	0.56	0.10	381
micro avg	0.58	0.58	0.58	28760
macro avg	0.43	0.54	0.43	28760
weighted avg	0.74	0.58	0.63	28760

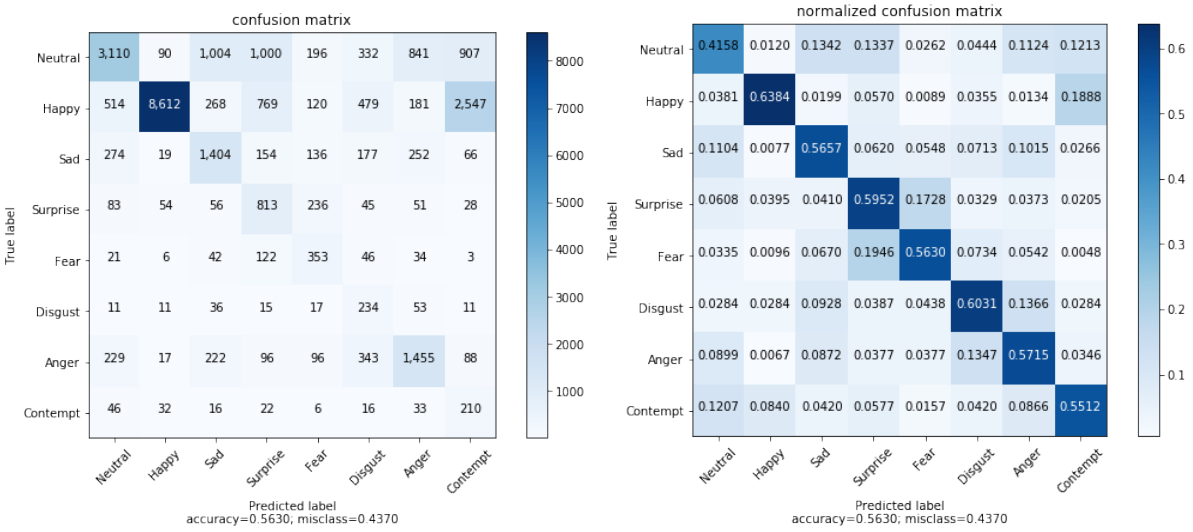


(a) *Confusion matrix for vgg16.* (b) *Normalized confusion matrix for vgg16.*

Figure 20: VGG16 with standard images.

Table 9: VGG16 preprocessed images classification report.

	precision	recall	F1 score	Support
Neutral	0.73	0.42	0.53	7480
Happy	0.97	0.64	0.77	13490
Sad	0.46	0.57	0.51	2482
Surprise	0.27	0.60	0.37	1366
Fear	0.30	0.56	0.40	627
Disgust	0.14	0.60	0.23	388
Anger	0.50	0.57	0.53	2546
Contempt	0.05	0.55	0.10	381
micro avg	0.56	0.56	0.56	28760
macro avg	0.43	0.56	0.43	28760
weighted avg	0.75	0.56	0.62	28760



(a) *Confusion matrix for vgg16 with pre-processing.*

(b) *Normalized confusion matrix for vgg16 with pre-processing.*

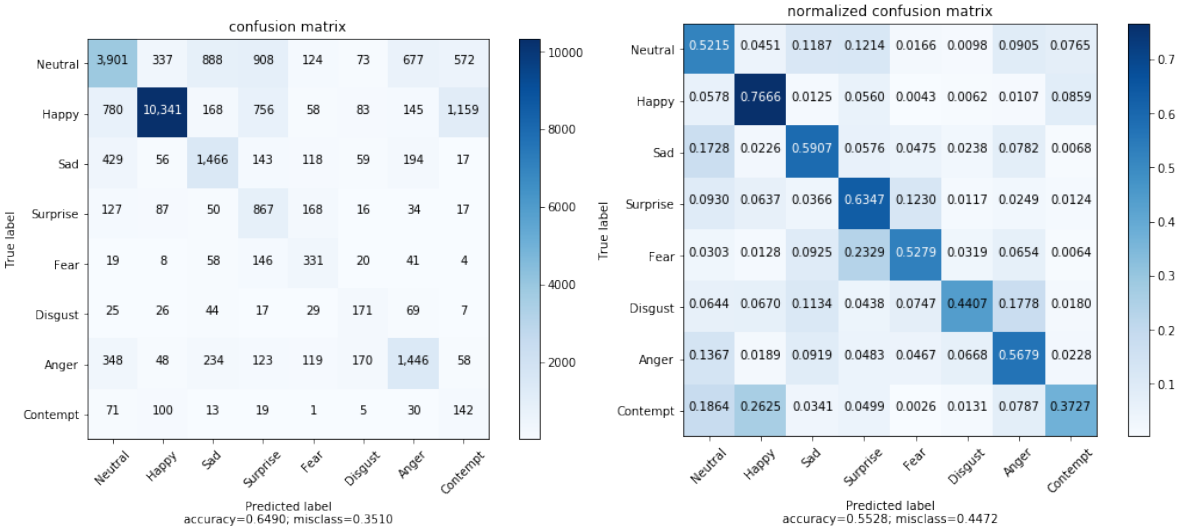
Figure 21: VGG16 with pre-processed images.

11.2 MobileNet

11.2.1 8 classes

Table 10: MobileNet standard images classification report

	precision	recall	f1-score	support
Neutral	0.68	0.52	0.59	7480
Happy	0.94	0.77	0.84	13490
Sad	0.50	0.59	0.54	2482
Surprise	0.29	0.63	0.40	1366
Fear	0.35	0.53	0.42	627
Disgust	0.29	0.44	0.35	388
Anger	0.55	0.57	0.56	2546
Contempt	0.07	0.37	0.12	381
micro avg	0.65	0.65	0.65	28760
macro avg	0.46	0.55	0.48	28760
weighted avg	0.74	0.65	0.68	28760

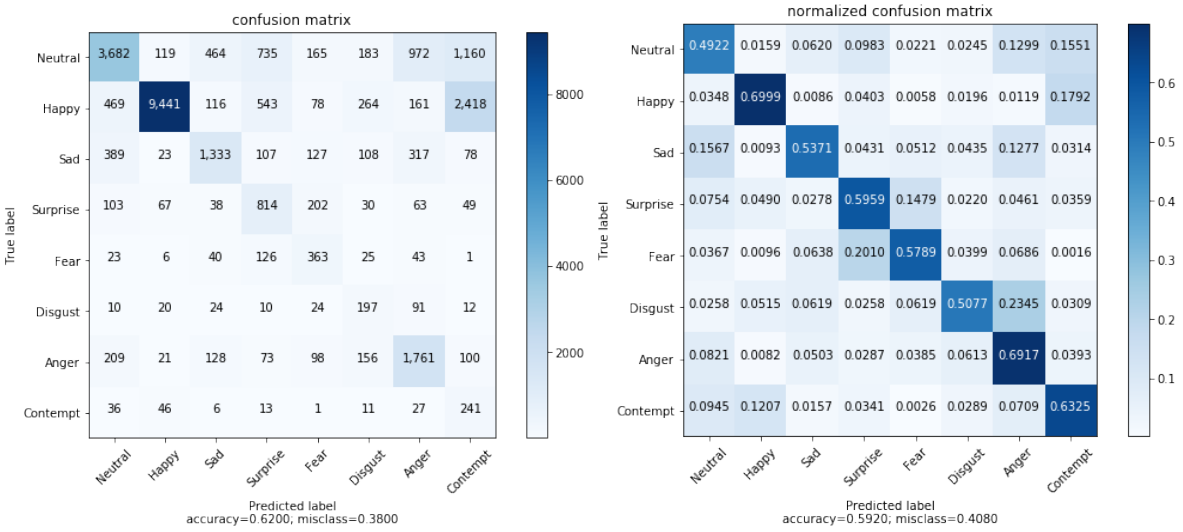


(a) *Confusion matrix for MobileNet.* (b) *Normalized confusion matrix for MobileNet.*

Figure 22: MobileNet with standard images.

Table 11: MobileNet preprocessed images classification report.

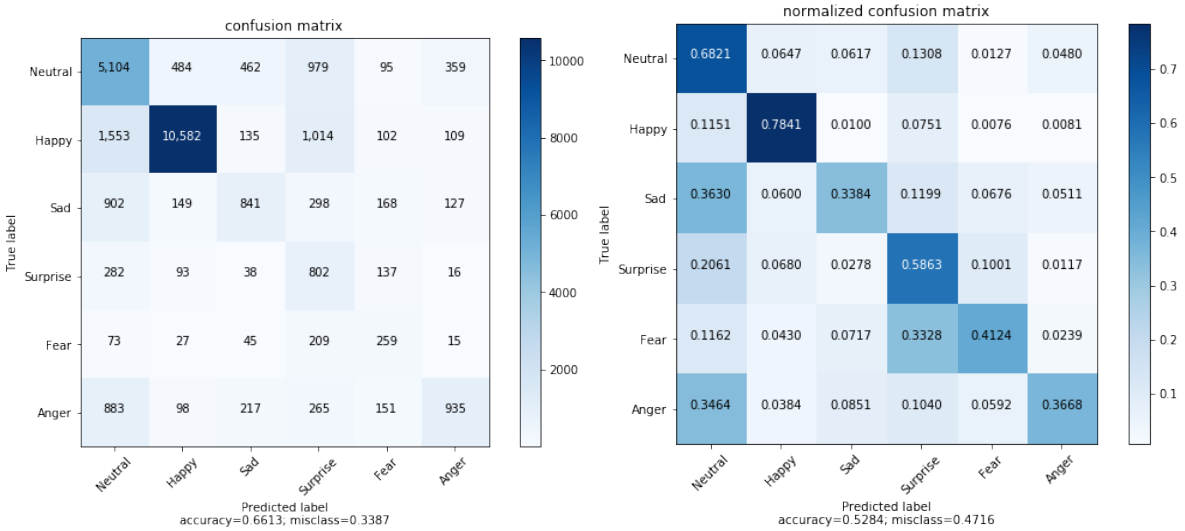
	precision	recall	F1 score	Support
Neutral	0.75	0.49	0.59	7480
Happy	0.97	0.70	0.81	13490
Sad	0.62	0.54	0.58	2482
Surprise	0.34	0.60	0.43	1366
Fear	0.34	0.58	0.43	627
Disgust	0.20	0.51	0.29	388
Anger	0.51	0.69	0.59	2546
Contempt	0.06	0.63	0.11	381
micro avg	0.62	0.62	0.62	28760
macro avg	0.47	0.59	0.48	28760
weighted avg	0.77	0.62	0.67	28760



(a) *Confusion matrix for mobilenet with pre-processing.* (b) *Normalized confusion matrix for mobilenet with pre-processing.*

Figure 23: MobileNet with pre-processed images.

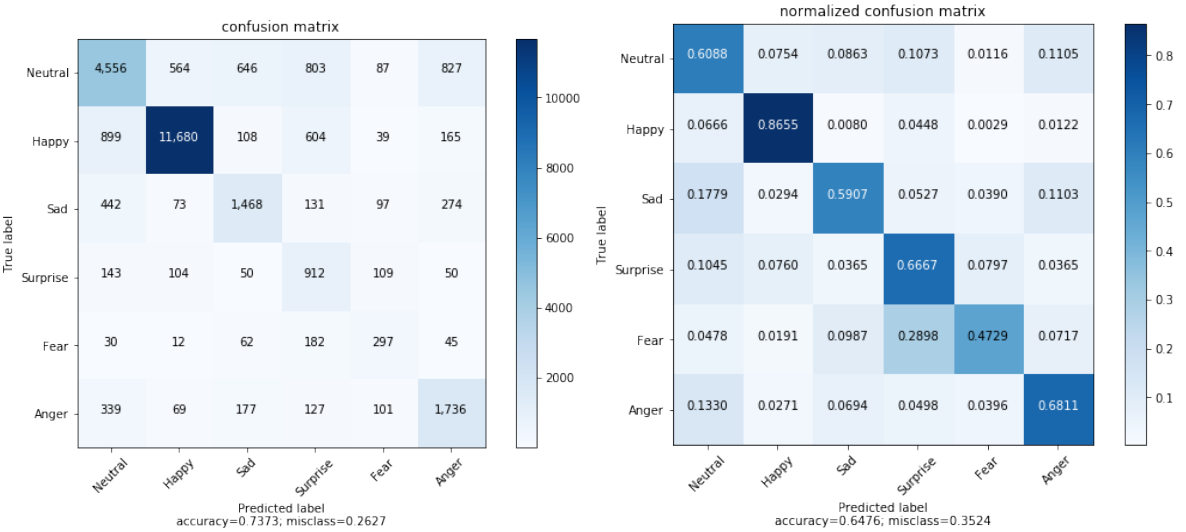
11.2.2 6 classes



(a) *Confusion matrix for mobilenet with 6 classes with standard images.*

(b) *Normalized confusion matrix for mobilenet with 6 classes with standard images.*

Figure 24: MobileNet with 6 classes with standard images.

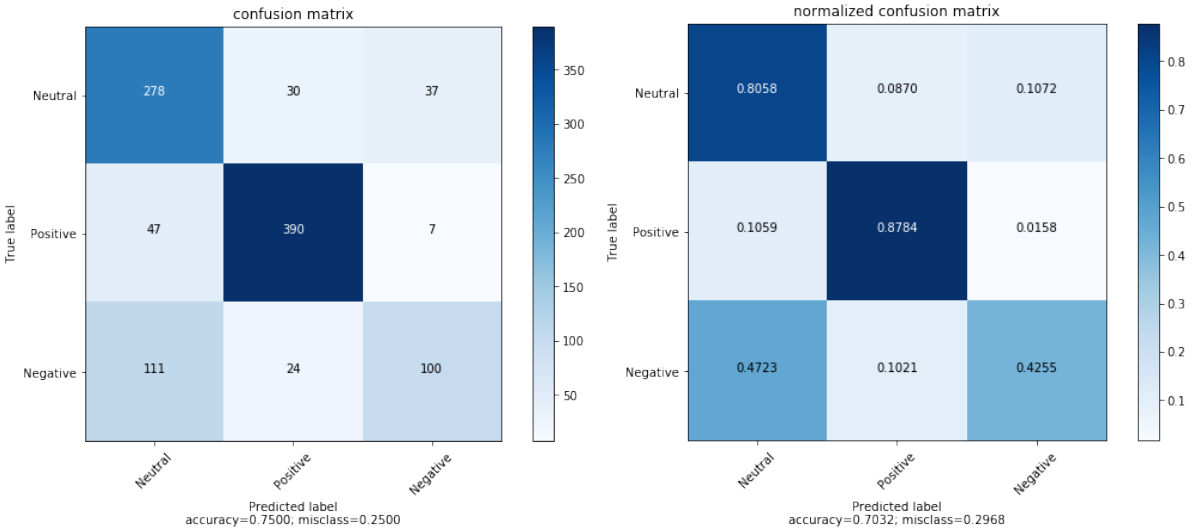


(a) *Confusion matrix for mobilenet with 6 classes with pre-processed images.*

(b) *Normalized confusion matrix for mobilenet with 6 classes with preprocessed images.*

Figure 25: MobileNet with 6 classes with pre-processed images.

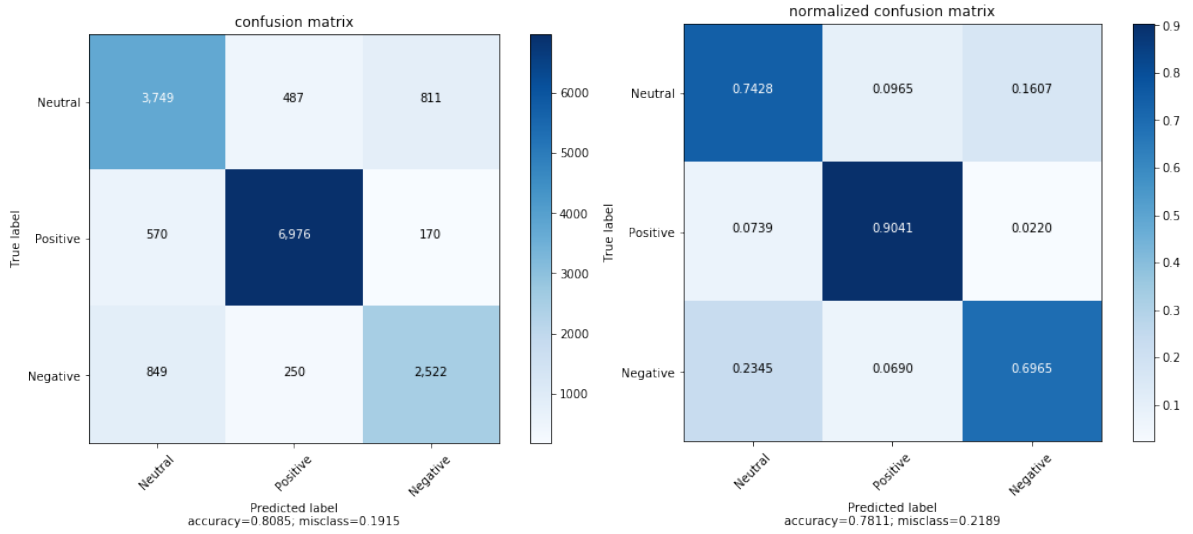
11.2.3 3 classes



(a) *Confusion matrix for mobilenet with 3 classes with standard images.*

(b) *Normalized confusion matrix for mobilenet with 3 classes with standard images.*

Figure 26: MobileNet with 3 classes with standard images.



(a) *Confusion matrix for mobilenet with 3 classes with pre-processed images.*

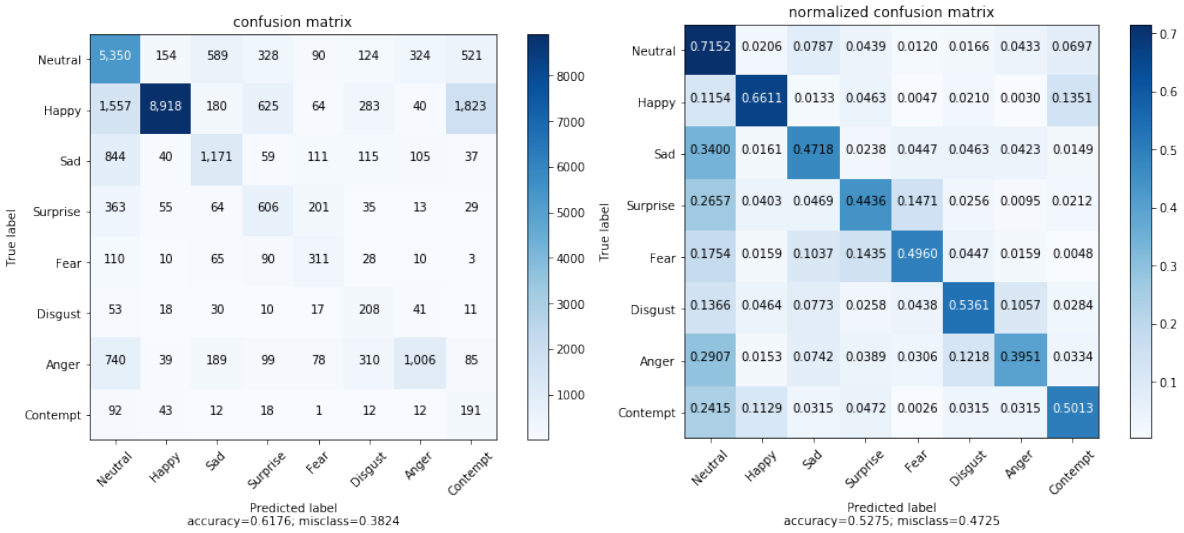
(b) *Normalized confusion matrix for mobilenet with 3 classes with pre-processed images.*

Figure 27: MobileNet with 3 classes with pre-processed images.

11.3 Xception

Table 12: Xception standard images classification report

	precision	recall	F1 score	Support
Neutral	0.59	0.72	0.65	7480
Happy	0.96	0.66	0.78	13490
Sad	0.51	0.47	0.49	2482
Surprise	0.33	0.44	0.38	1366
Fear	0.36	0.50	0.41	627
Disgust	0.19	0.54	0.28	388
Anger	0.65	0.40	0.49	2546
Contempt	0.07	0.50	0.12	381
micro avg	0.62	0.62	0.62	28760
macro avg	0.46	0.53	0.45	28760
weighted avg	0.73	0.62	0.65	28760



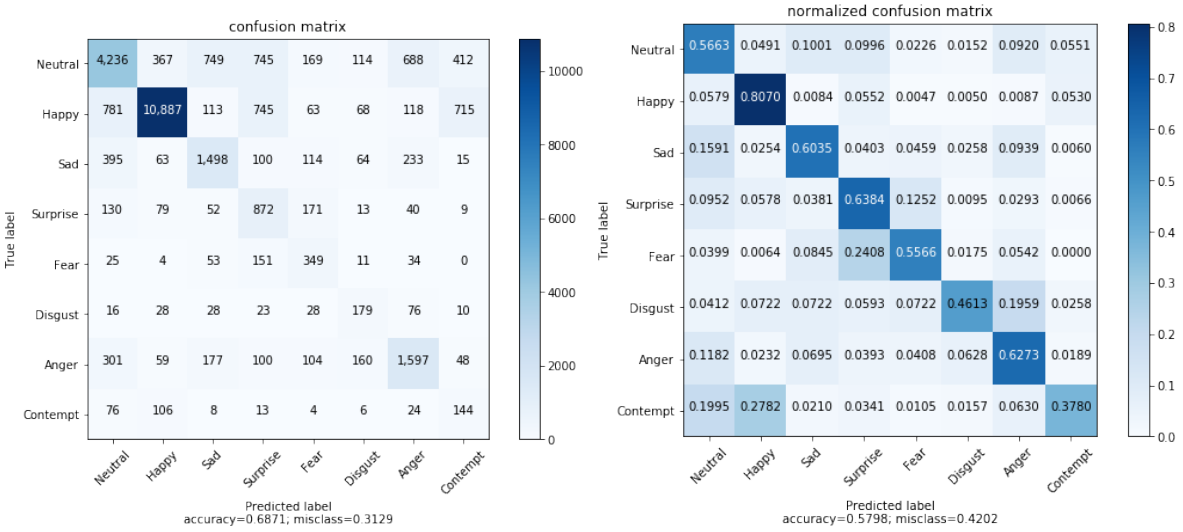
(a) *Confusion matrix for xception.*

(b) *Normalized confusion matrix for xception.*

Figure 28: Xception with standard images.

Table 13: Xception preprocessed images classification report

	precision	recall	F1 score	Support
Neutral	0.71	0.57	0.63	7480
Happy	0.94	0.81	0.87	13490
Sad	0.56	0.60	0.58	2482
Surprise	0.32	0.64	0.42	1366
Fear	0.35	0.56	0.43	627
Disgust	0.29	0.46	0.36	388
Anger	0.57	0.63	0.60	2546
Contempt	0.11	0.38	0.17	381
micro avg	0.69	0.69	0.69	28760
macro avg	0.48	0.58	0.51	28760
weighted avg	0.75	0.69	0.71	28760



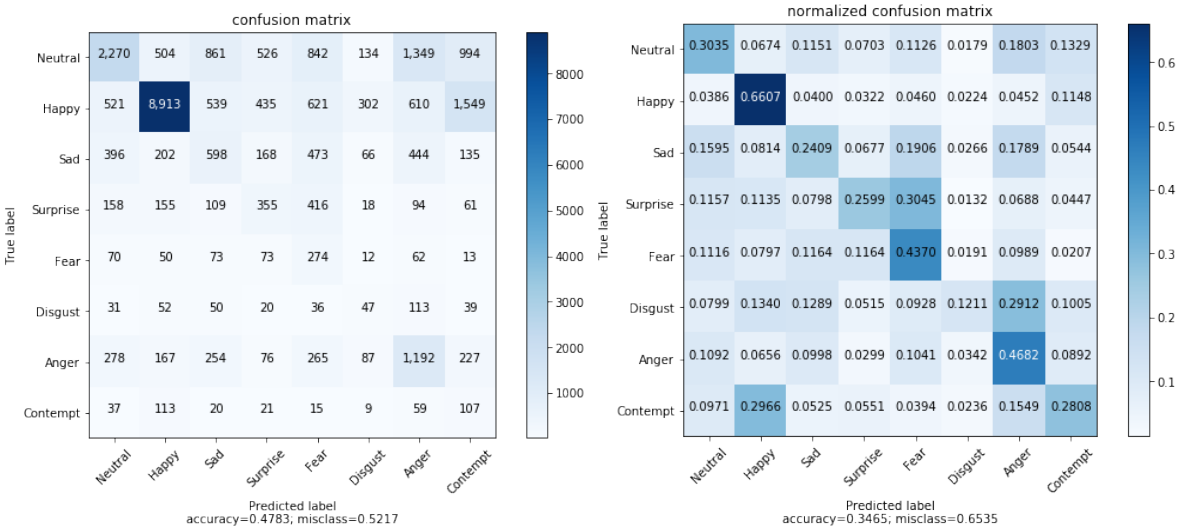
(a) *Confusion matrix for xception with pre-processing.* (b) *Normalized confusion matrix for xception with pre-processing.*

Figure 29: Xception with pre-processed images.

11.4 Small CNN

Table 14: Small CNN standard images classification report.

	precision	recall	F1 score	Support
Neutral	0.60	0.30	0.40	7480
Happy	0.88	0.66	0.75	13490
Sad	0.24	0.24	0.24	2482
Surprise	0.21	0.26	0.23	1366
Fear	0.09	0.44	0.15	627
Disgust	0.07	0.12	0.09	388
Anger	0.30	0.47	0.37	2546
Contempt	0.03	0.28	0.06	381
micro avg	0.48	0.48	0.48	28760
macro avg	0.30	0.35	0.29	28760
weighted avg	0.63	0.48	0.53	28760

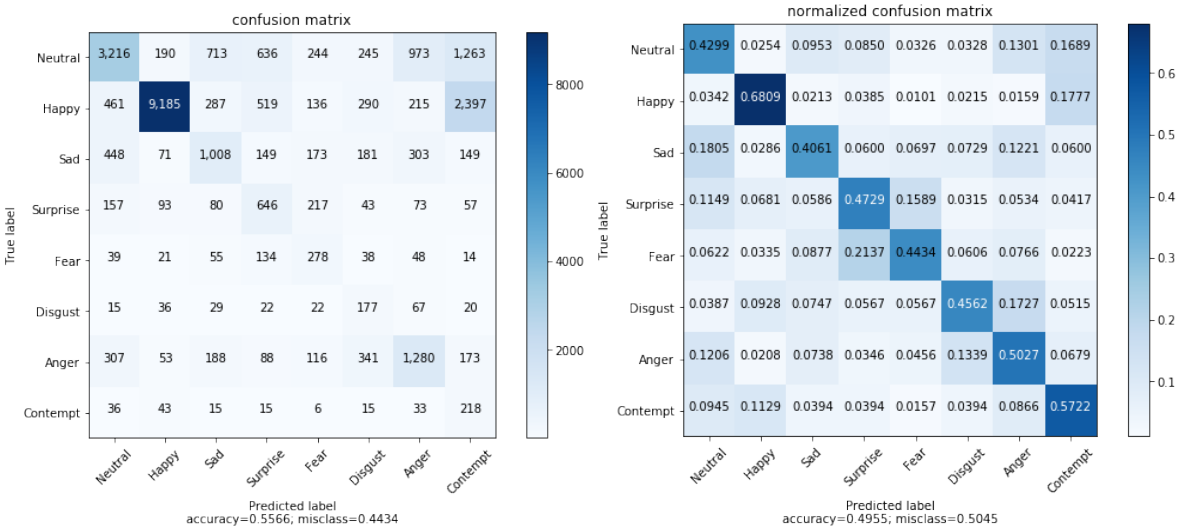


(a) *Confusion matrix for the small CNN.* (b) *Normalized confusion matrix for the small CNN.*

Figure 30: Small CNN with standard images.

Table 15: Small CNN Preprocessed images calssification report.

	precision	recall	F1 score	Support
Neutral	0.69	0.43	0.53	7480
Happy	0.95	0.68	0.79	13490
Sad	0.42	0.41	0.42	2482
Surprise	0.29	0.47	0.36	1366
Fear	0.23	0.44	0.31	627
Disgust	0.13	0.46	0.21	388
Anger	0.43	0.50	0.46	2546
Contempt	0.05	0.57	0.09	381
micro avg	0.56	0.56	0.56	28760
macro avg	0.40	0.50	0.40	28760
weighted avg	0.72	0.56	0.61	28760



(a) *Confusion matrix for the small CNN with pre-processing.*(b) *Normalized confusion matrix for the small CNN with pre-processing.*

Figure 31: Small CNN with pre-processed images.