Comparing the performance of facial expression classification between a human agent and a Convolutional Neural Network

Alva Liu
August Bergman
Sasha Hellstenius

Abstract. The ability for humans to be able to perceive the feelings and intentions of others has been enormously important in human evolution. With the aid of algorithms that can recognize facial expressions, important advances can be made in the field of artificial intelligence and robotics. The aim of this study was to examine how a convolutional neural network performs in a facial expression classification task compared to manual classification done by a human agent. A convolutional neural network was implemented in Lasagne, a library for building and training neural networks in Theano, and trained on the FER-2013 dataset. FER-2013 is a labelled dataset that originates from the ICML 2013 Workshop on Challenges in Representation Learning, and contains 35887 grayscale images of size 48x48 pixels distributed over the seven emotion classes anger, disgust, fear, happy, sad, surprise and neutral. The results showed that the neural network could not achieve human-like performance on the classification task; it generated an overall accuracy of 55.7 % on the testing data whereas the human agent was able to achieve 63.7 %. In the absence of sufficient time and computational power the complexity of the architecture was limited. Even though the results from this study did not show that the convolutional neural network could perform better than the human agent, earlier studies indicate that this is possible.

1 Introduction

Facial recognition is an important field within computer vision with many applicational uses. Recent advances in deep learning models have led to the development of models that perform well compared to the facial recognition capabilities of humans [1] [2].

The potential benefits of facial recognition software are vast. One such benefit is the ability for software to classify human expression and to identify the subtle differences of facial expressions that constitutes one of the ways humans use non-verbal communication to express their emotions. The ability for humans to be able to perceive the feelings and intentions of others has been enormously important in human evolution, making the ability of algorithms to correctly recognize facial expressions an important marker for advances in artificial intelligence.

Convolutional Neural Networks (CNN) have been successfully applied to many different visual recognition tasks. By exploiting different attributes of the spatial structures of images, CNNs can be used to train image classifiers that require vastly less network parameters than more traditional approaches [3].

This study explored how a CNN can be taught to recognize facial expressions using image pixel data. After training the image classifier using labelled data, a manual classification is performed using the same set of image data in order to benchmark the classification capabilities of a human on this dataset. The classification accuracy of the CNN model and the manual classification is then compared in order to determine whether a CNN classifier can achieve human-like performance in the task of recognizing facial expressions.

1.1 Problem formulation

How does the classification performance of a convolutional neural network compare to that of a human agent in recognizing human facial expressions?

2 Background

In regard to image classification, Neural Network architectures tend to perform better by increasing the number of layers or the number of neurons in the network, as this allows the network to learn more complex functions. This increase in complexity, however, tends to make the networks more prone to overfitting, and require more computational power [1]. CNNs take advantage of spatial locality between pixels within an image to vastly reduce the amount of parameters necessary for the model, and the use of CNNs have resulted in significantly improved performances in image classification [4].

Goodfellow et al. [5] introduced the Facial Expression Recognition 2013 (FER-2013) dataset, consisting of photos of human faces expressing one of seven different emotions. Introducing a completely new dataset for this well-known problem was expected to have the benefit of avoiding overfitting on the test set for a commonly benchmarked dataset. Goodfelow et al. held a contest in designing facial expression recognition systems, the top three best performing participants all used CNNs.

Deep CNNs have been used for many image recognition tasks. In the widely acclaimed study by Krizhevsky et al. [6] a deep CNN was used to classify 1.3 million images from the ImageNet dataset into 1000 different classes and was able to achieve considerably better performance than the previous state-of-the-art classifiers. The network used consists of five convolutional networks, some followed by max-pooling layers, and two dense layers with a final application of a softmax layer.

Tang, Y. [2] proposed a network replacing the commonly used softmax layer with a linear support vector machine, and minimizing a margin-based loss during learning instead of the cross-entropy loss. Tang was able to achieve a classification accuracy of 69.4 % on the FER-2013 dataset.

Mollahosseini et al. [1] used a network consisting of two convolutional layers followed by max pooling and inception layers to classify facial images into six different expressions, and performed experiments on six different publicly available facial expression datasets. Using the proposed network, they achieved a classification accuracy for the FER-2013 dataset of 66.4 %.

3 Approach

A literature study was conducted in order to explore the state-of-the-art for the facial expression recognition problem. An appropriate dataset with pre-labelled image data was then selected. A network architecture was selected and trained using the Lasagne library for Python, after which the best performing model was selected. Manual classification was also performed on the testing data.

3.1 The dataset

The image dataset used in this study originates from the ICML 2013 Workshop on Challenges in Representation Learning, where a completely new dataset for the facial expression recognition problem was introduced. This dataset was created by searching for images of faces that match emotion-related search strings using Google Image Search. The search queries were then processed by letting humans disqualify images that had been incorrectly labelled by the query process. The remaining images were resized to a pixel size of 48x48, converted to grayscale and then labelled with one of the seven emotions Angry, Disgust, Fear, Happy, Sad, Surprise or Neutral. The complete dataset consists of 35887 images in total, with 28709 images in the training set, 3589 in the validation set and 3589 in the testing set [5]. Figure 1 shows the distribution of the emotions in the training set.

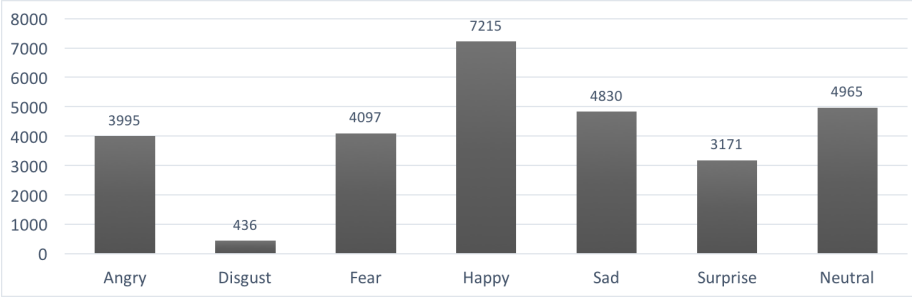

Fig. 1. The distribution of the six emotions angry, disgust, fear, happy, sad, surprise, and neutral in the training set.

3.2 The network architecture

The network architecture chosen for this study is presented in Figure 2 and Table 1. The choice of architecture was limited due to time- and computational constraints.

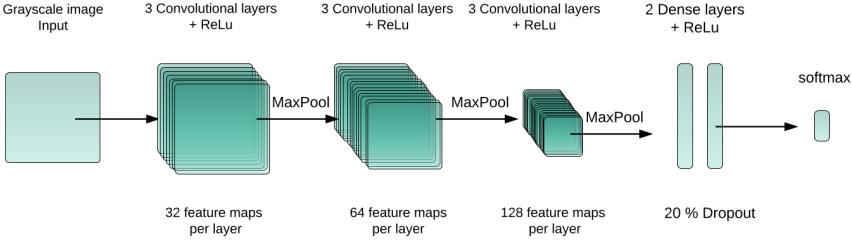

Fig. 2. An overview of the network architecture. The network begins with an input layer followed by 3 sequential convolutional layers with 32 filters and a max-pooling layer. This sequence is repeated with an increasing amount of filters. The network ends with two dense layers with 20% dropout followed by an output layer.

Type	Filter size/stride	Depth	Output size	Params (W+b)
input			48 x 48 x 1	0
convolution	3 x 3/ 1	32	48 x 48 x 32	(3*3*1)*32 + 32
convolution	3 x 3/ 1	32	48 x 48 x 32	(3*3*32)*32 + 32
convolution	3 x 3/ 1	32	48 x 48 x 32	(3*3*32)*32 + 32
max-pool	2 x 2/ 2	0	24 x 24 x 32	0
convolution	3 x 3/ 1	64	24 x 24 x 64	(3*3*32)*64 + 64
convolution	3 x 3/ 1	64	24 x 24 x 64	(3*3*64)*64 + 64
convolution	3 x 3/ 1	64	24 x 24 x 64	(3*3*64)*64 + 64
max-pool	2 x 2/ 2	0	12 x 12 x 64	0
convolution	3 x 3/ 1	128	12 x 12 x 128	(3*3*64)*128 + 128
convolution	3 x 3/ 1	128	12 x 12 x 128	(3*3*128)*128 + 128
convolution	3 x 3/ 1	128	12 x 12 x 128	(3*3*128)*128 + 128
max-pool	2 x 2/ 1	0	6 x 6 x 128	0
dropout (20 %)		0	6 x 6 x 128	0
dense		1	64	(6*6*128)*64 + 64
dropout (20 %)		0	64	0
dense		1	64	64*64 + 64
softmax		0	7	64*7 + 7

Table 1. Description of the network architecture. Each layer is presented with corresponding filter sizes, stride, depth, resulting output size and the number of parameters.

The architecture presented in Figure 2 consists of an input layer, two repetitive sequences and an output layer. The first sequence is repeated three times and consists of the layers convolution, convolution, convolution and max-pool. The second sequence is repeated two times and consists of the layers dropout and dense.

All convolutional layers were each succeeded by a ReLu operation. The choice of using ReLu over other non-linearity operations was based on the simplicity of computing the gradient as well as the reduced likelihood of the gradient vanishing. The input layer and response maps that were input to the convolutional layers were zero padded to maintain dimensions. To make the data representation smaller and more manageable max-pooling was chosen. The filter size of the convolutional layers was 3x3 with a depth corresponding to the previous layer. The stride was set to 1. The size of the max-pool region was 2x2 and the stride was set to 2.

The dropout layers in the second repetitive sequence were added to combat overfitting [6]. The dropout layers were followed by dense layers that were succeeded by ReLu operations. To accelerate learning and dampen oscillation momentum was used. To avoid overshooting local optimums and picking up too much speed Netrov Accelerated Gradient (NAG) was the chosen method.

The hyper-parameters that were subject to a coarse grain search were learning rate and dropout. A feasible learning rate was first found by testing values

in the range of 0.001 to 0.1. Once the learning rate was established the dropout parameter was tested in the range of 10-40%. The filter sizes, number of filters and stride values were fixated due to time constraints. With the selected parameters the network was trained for 50 epochs, overfitting started after 12 epochs.

3.3 Computational complexity

The limiting computational factor of a CNN is memory constraints [3]. The network architecture was thus selected due to its time and computational requirements. The highest memory requirements were from the early convolutional layers. The most amount of parameters arises from the dense layers. The total amount of learnable parameters in the network was 779,783. The memory complexity for the network during the forward-pass was calculated to be roughly 1,7 MB memory per image.

3.4 Manual classification process

The manual classification of the images was performed by letting one human agent classify the testing data.

4 Results

The network was trained with 20 % dropout before each of the two dense layers, and the best performing learning rate 0.005 for 12 epochs. The training and validation loss for each epoch during the training process of the CNN is shown in column 2 and 3 in Table 2. Column 4 shows the ratio between the training and validation loss. After training for a few epochs, the ratio started to decrease which implies that the decrement of validation loss is decelerating in comparison to the decrement of training loss. The fifth column shows that the validation accuracy increments as the network is trained, and the last column states the training time for each epoch.

Epoch	Training loss	Validation loss	Training/Validation	Validation Acc	Duration
1	1.82905	1.81899	1.00553	0.25122	1006.67s
2	1.76800	1.66873	1.05949	0.34993	1007.94s
3	1.60555	1.51782	1.05780	0.41713	1011.65s
4	1.48993	1.42295	1.04707	0.45665	1006.29s
5	1.39959	1.35395	1.03371	0.48294	1009.20s
6	1.31576	1.29310	1.01752	0.50836	1009.48s
7	1.25740	1.25950	0.99833	0.52228	1007.42s
8	1.20078	1.24313	0.96593	0.52037	1006.55s
9	1.14897	1.23225	0.93242	0.53081	1009.51s
10	1.10524	1.26091	0.87654	0.53151	1008.39s
11	1.06381	1.26210	0.84289	0.53430	1010.17s
12	1.03020	1.24039	0.83054	0.54161	1006.38s

Table 2. Details of the training process.

The resulting classification accuracy of the CNN and the human agent were 55.7 % and 63.7 % respectively. The accuracy scored for each emotion is presented in Figure 3 and Table 3. The human agent performed better when classifying the emotions Angry, Disgust, Happy, Surprise and Neutral. The CNN performed better when classifying the emotions Fear and Sad. The discrepancy of the classifications was the greatest when classifying the Disgust emotion and the smallest with the Fear emotion. This is seen in Figure 3 as the distance between the corresponding stacks. The difference in accuracy per emotion compared to the total accuracy of each classifier is presented in Table 3. The CNN performs worse on every emotion except Happy when compared to the total accuracy. The human agent performs better than the total accuracy on a majority of the emotions.

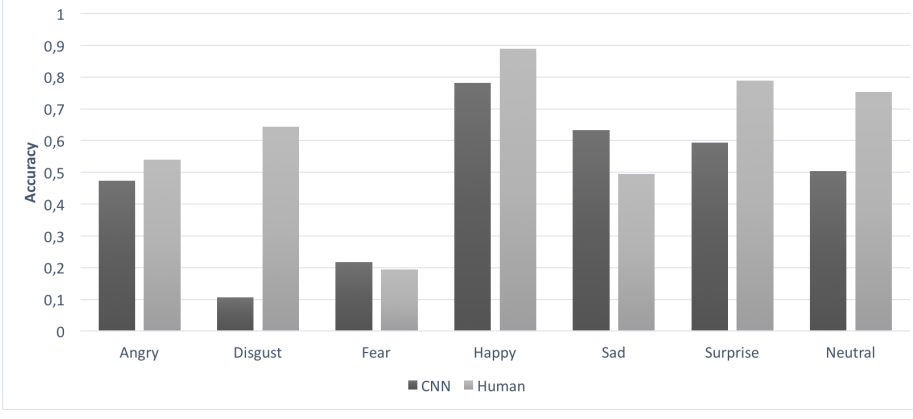

Fig. 3. The classification accuracy of human agent (light gray) and CNN (dark gray), class by class.

Emotion	Human Accuracy	CNN Accuracy	Difference Human	Difference CNN
Angry	54.0	47.3	-1.7	-16.4
Disgust	64.3	10.7	8.6	-53.0
Fear	19.4	21.8	-36.3	-41.9
Happy	88.8	78.2	33.1	14.5
Sad	49.5	63.2	-6.2	-0.5
Surprise	78.8	59.3	23.1	-4.4
Neutral	75.9	50.4	20.2	-13.3

Table 3. Classification accuracy. Column two and three present the accuracy of the human agent and CNN per emotion. Column four and five present the difference in accuracy per emotion compared to the respective classifiers total accuracy.

5 Discussion

Related works have utilized CNNs in order to perform FER classification and there exists no real consensus as to which CNN architecture works best as, for example, the image pre-processing is different and many different datasets are used for benchmarking. The network architecture used in this study was chosen because of its relative ease of implementation and as a compromise between recommendations from earlier studies and computational limitations.

The best performing network achieved an overall classification accuracy of 55.7 %, compared to 63.7 % for the manual classification. These results show that the CNN was not able to achieve a human-like performance in regards to classifying facial expressions. However, the state-of-the-art analysis [5] shows that networks from earlier studies have been able to achieve accuracies that surpass that of the human agent, implying that it indeed is possible to train a CNN that exceeds human capabilities for facial recognition. Additional computational resources and time is necessary for this study to achieve similar results.

The CNNs failure to classify the emotion Disgust is likely due to the fact that the training set only has 436 images labelled Disgust. Hence, the CNN was not able to learn disgust. For the emotions Fear and Sad, the CNN outperformed the human agent. A reason to this could be that the human agent interprets the emotions fear and sad differently than the pixel patterns in the images labelled with Fear and Sad. This problem does not occur for the CNN since there is no actual recognition of emotion, simply pattern recognition.

5.1 Dataset

The way the labelling process was done could potentially have led to labelling errors. For the purpose and scope of this study however, the dataset was deemed to be sufficient. Such a dataset inevitably leads to a discussion of how to retrieve a significant amount of training data of adequate image quality, and how to discretely label the human expression of emotions, which undoubtedly is a highly

complex and subjective process.

The results from this study also indicate that classification of emotions is a difficult task for both human agents and CNNs. One reason that this type of classification is problematic lies in the subjective nature of the labelling.

The labels of the original data could be incorrect since they do not all correspond to human classification performed in this study. However it is most likely that several emotions can be found in a single image and the most prominent emotion experienced differs from individual to individual. If the classification of emotions is inconsistent when performed by human agents the labeling and training of the CNN will suffer. If the CNNs classification was flawless there would be many practical implementations that would be of great benefit for society. An example were major changes could be made is health care. With an aging population the need for innovation within this field is crucial.

5.2 Approach

The network architecture was limited by the available computational power. A better performing network could have been designed with more resources. This implied a time-constraint that limited the scope of the hyper-parameter search.

As the manual classification was performed by the authors, the risk of introduced bias to the result was high because of the prior knowledge of the dataset. In particular, knowledge of the distribution of the different labels affected the classification. The manual classification was also subject to basic human error, for example missclassfication.

6 Conclusions

Although the results from this study show that the CNN does not perform better than a human agent previous research indicates that this can be achieved [5]. The lower accuracy of the CNN in this study is thought to be a result of the chosen architecture. A deeper network may have increased the accuracy, however this was not an alternative for this study due to the limited time and computational constraints.

References

1. Mollahosseini, A., Chan, D., Mahoor, M.H.: Going deeper in facial expression recognition using deep neural networks. In: Applications of Computer Vision (WACV), 2016 IEEE Winter Conference on, IEEE (2016) 1–10
2. Tang, Y.: Deep learning using linear support vector machines. arXiv preprint arXiv:1306.0239 (2013)
3. Karpathy, A.: Cs231n: Convolutional neural networks for visual recognition. Online Course (2016)
4. Pramerdorfer, C., Kampel, M.: Facial expression recognition using convolutional neural networks: State of the art. arXiv preprint arXiv:1612.02903 (2016)
5. Goodfellow, I.J., Erhan, D., Carrier, P.L., Courville, A., Mirza, M., Hamner, B., Cukierski, W., Tang, Y., Thaler, D., Lee, D.H., et al.: Challenges in representation learning: A report on three machine learning contests. In: International Conference on Neural Information Processing, Springer (2013) 117–124
6. Srivastava, N., Hinton, G.E., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.: Dropout: a simple way to prevent neural networks from overfitting. Journal of Machine Learning Research **15**(1) (2014) 1929–1958