Towards Human-like Performance Face Detection: A Convolutional Neural Network Approach

Joshua van Kleef
University of Twente
P.O. Box 217, 7500AE Enschede
The Netherlands
j.a.vankleef-1@student.utwente.nl

ABSTRACT

Face detection is a widely discussed topic in the field of computer vision. The application of face detection varies from convenient HMI- to government-scale surveillance applications. The face detectors perform well when a image is provided that is in a perfect condition, such as an official portrait. In this condition a face is perfectly aligned to the lens of the camera, not occluded and it has no pose. The challenge for a face detector is to detect a face when it is not in a perfect condition and this situation happens more often than not in real life situations. The goal of this research is to find out what the impact of providing more training data to a convolutional neural network (CNN)-based face detector is. The face detector is constructed by the deep learning framework Caffe. The model used for the face detection is AlexNet. There were 10 versions of face detectors created, varying in size and how many iterations it gone through. There is a performance gain from 39 to 64 correct accepts, between the versions with the least and most training data, so the performance is increased when more training data is presented.

Keywords

Deep learning, face detection, convolutional neural network, computer vision

1. INTRODUCTION

Deep learning is a concept that is impacting our daily lives more often. We do not notice it, because deep learning is used in the background of the applications that we use. An example of deep learning that is used in our daily lives is the search algorithm of Google Search\(^1\). Google finds patterns in data they have acquired and try to come up with relevant search results specific for the person who executed the query. Google uses deep learning to acquire business intelligence and become more valuable for their users, but there are also applications of deep learning that are used in life-threatening situations. It is for example used in the medical world to classify skin cancer\(^2\). The fundamental idea of deep learning dates back to 1980, when Fukushima established the Neocognitron\(^5\). This is an algorithm that is based on neurophysical characteristics that can be executed by a computer. So the learning algorithms that are used in deep learning are based on how a human learns things. The scope of the application of deep learning in this paper is face detection. Face detection is used in several applications, such as convenient HMI applications, but also more serious topics, such as surveillance software. This surveillance software scans camera footage for faces and tracks these through facial recognition software to recognize a face. The process of detecting faces in camera footage is a challenge, because face detection software performs poorly in poor conditions. These poor conditions are when faces are not well-posed, occluded, not well-lit, over-exposed, blurry, grainy or the resolution of the camera footage is too low. When a face is not detected, it cannot run through facial recognition software, a demand for a good face detector is high.

1.1 Convolutional Neural Network

A convolutional neural network is a deep learning algorithm that is used in object recognition. This algorithm is proposed by Vaillant et al.\(^13\). CNN is a successor of Fukushima’s Neocognitron\(^5\). The philosophy of a CNN is to train the network, the same way as a human learns things. A convolutional neural network is a deep learning algorithm that is based on neurophysical characteristics of the object that needs to be located by the CNN. Figure 1 shows an example of the architecture of a CNN. All the layers have specific algorithms for certain feature extraction. Examples of feature extraction are the contrast of a picture or what colors are present in the pictures. These layers are connected to each other to combine all the feature extractions. In the training phase will be looked when certain features are present in images and the CNN gives a weight to these features when they occur more often. When someone wants to classify an image by a CNN, it applies the feature extraction, applies the weights that are calculated in the training phase and eventually outputs the result of the classification of an image. And because a human face can look like a face of a dog, there also need to be images supplied of objects that are not faces. Because the memory size of modern day computers is not sufficient enough to load all the images in its memory, the CNN performs the calculations in iterations to calculate the features.

\(^1\)www.google.com

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

\(^2\)\(^4\)th Twente Student Conference on IT January 22\(^{nd}\), 2016, Enschede, The Netherlands.

Copyright 2016, University of Twente, Faculty of Electrical Engineering, Mathematics and Computer Science.
1.2 Face Detection

There are several face detection algorithms that are already present. The face detection algorithms can be subdivided in three categories. These categories are cascade based, DPM-based and neural network-based[3].

1.2.1 Cascade-Based

The primary element of cascade based face detectors are haar-features. These haar-features are simple elements that describe how certain pixels are aligned. An array of haar-features describes how a feature can look like in a picture. Viola Jones[15] is the most well-known cascade-based face detection algorithm.

1.2.2 DPM-based

DPM stands for deformable parts model [4]. This model is based on parts that construct a face based on their geometric position relative to each other. For example a left eye (a part) needs to be left of an nose (another part). This model is robust against occlusion, because when a mouth and a nose is discovered in the right geometric place, then the algorithm can pretend that there are also eyes, even if the subject has sun glasses on.

1.2.3 Neural Network-based

These face detectors are based on self-learned features, where the DPM-based and Cascade-based face detectors have certain type of features that can be detected in an image, a neural network-based face detector is free of these features. So, a neural network-based face detector can learn new patterns and be more agile in difficult environments.

1.3 Problem Statement

Viola-Jones is the school book example of face detection [15]. Because of its fast computational time, this face detection algorithm is popular for analyzing video and fast face detection. It is also used in devices that have limited resources and where the computation of the face detector needs to happen on the device itself, instead of doing the computation on powerful servers. The trade-off with Viola-Jones is that it does not detects faces when it is in a perfect condition. For example, figure 1 is a snapshot of Cognitec’s promotional video[2]. In this video is demonstrated how their object recognition software FaceVACS[3] performs. Overall it performs well, but when a face is occluded or their pose is turned to a different angle, the face detection software does not perform well. Of the ten true positives face the face detection can return, it now returns two. To not bring Cognitec in discredit, this is a snapshot of their video to illustrate the problem a face detector can face. In other frames more faces are detected when the situation is more optimal for the face detector.

1.4 Research Questions

To find out if a convolutional neural network-based face detection system can actually outperform a state-of-the art face detector such as Viola-Jones, is too time-consuming in the scope of this research. Therefore the scope of this research will be the performance gain of a CNN-based face detector in relation to the presented size of the training data. The research question is as follows:

To what extent does the performance of a CNN-based face detector increase when more training data is presented?

1.5 Related Work

A lot of research has been done in the field of face detection. But because the scope of this paper is face detection based on CNN, the related work-section will be limited to articles that are subject both to face detection and deep learning. The most recent research is by Farfade et Al. [3]. This research proposed a new face detection method called Deep Dense Face Detector. This new face detector performed overall better than state-of-the-art face detectors. Facebook has also done research with neural network based face detectors, their method is called DeepFace [12]. In 2014, Girshick et Al. developed a method called R-CNN [6], what is a object detector based on a CNN. This method uses selective search [14] to split a image in multiple regions.

2. RESEARCH METHOD

The CNN model that has been used for this research is AlexNet [10]. The reason that this model has been chosen is that Farfade et al. [3] also used this model in their research on CNN-based face detection. The results of this research are very good, so AlexNet is a proven method on face detection. The framework that is used to implement AlexNet is Caffe [9]. This is an open source software package developed by the Berkeley vision and learning center. The images used to train the model are the Labeled Faces in the Wild (LFW) dataset[7, 11] and ImageNet[1]. LFW is a dataset of crops of faces of famous people, Figure 3 is an example of how the faces could look like. Because these images are shot “in the wild”, this dataset provides excellent examples of how a face could look like in a real life situation. So the LFW dataset provides the positive examples for the training of the model. ImageNet provides the negative examples for the model. It consists of a couple of million categorized images. These images are categorized by a community of people for the sake of training an applying object recognition algorithms. Because the total ImageNet dataset is more than 1,5 TB and to download this amount of data took about 40 days, a sub sample of ImageNet is downloaded. This sub sample was downloaded as a sub sample of ImageNet is downloaded.

To what extent does the performance of a CNN-based face detector increase when more training data is presented?
contains 700,000 images. The categories that are present in this dataset are subsets of: animals, art, artefacts, fungus, geological, miscellaneous, natural objects and plants. The excludes categories are: persons and sports. Sports is also excluded, because there is a chance faces are present in these images. Figure 4 is an example of how these images could look like.

2.1 Instances of the model
There are two instances created of the model to look if the ratio of training data impacts the performance of the training data. The first instance has a ratio of 10,000 positive examples and 500,000 negative examples. The second instance has a ratio of 5,000 positive examples and 500,000 negative examples. These positive and negative examples are chosen randomly by the computer for both instances. The negative examples of both instances are a size of 500,000, but because these sets are randomly chosen, the actual training data is not the same. Both instances trained till 5000 iterations. In an interval of 1000 a snapshot is taken of the instances, so 5 versions of both instances are saved, which results in 10 versions of the model. On these 10 versions of the model the research is performed.

2.2 Training Phase
The GPU that is provided for this research is a NVIDIA GeForce GT 430 with an internal memory of 1GB. The required memory size for training this model was at least 3 GB. Because its memory was insufficient to train the CNN, the training has been done on an Intel Core i7 processor with 8 GB of RAM, this was sufficient to train a single model till 5000 iterations for 7 days.

2.3 Detector
To detect faces in images, it is needed to create bounding boxes for the image to let the CNN detect faces in these prepared bounding boxes. The algorithm that was responsible for creating the bounding boxes was selective search [14]. This algorithm produces 2000 boxes for the image, no matter what the size of the image is, the boxes vary in size and position in the image, such that even small faces can be detected. Figure 5a is an example of all the bounding boxes that are produced by selective search. In this figure there are so many bounding boxes produced that the image itself can not be seen anymore. In figure 5b the CNN is applied on the bounding boxes to detect which bounding box contains a face. Because the CNN is applied on small bounding boxes as well as big ones, the output of the CNN detector produces multiple matches on the same face, as can be seen in figure 5b. So a new algorithm was developed to figure out which of these overlapping bounding boxes produces the highest probability of containing a face to reduce redundant overlapping results of the detector. The result of this algorithm can be seen in figure 5c. This algorithm does not simply return only one bounding box with the highest probability, but it can return multiple bounding boxes corresponding to how many faces are present in the image, such as in figure 6.

3. Evaluation
Despite the fact that the memory of the GPU was insufficient to train the CNN, it was sufficient to test the CNN with the GPU enabled. The reason that it could be used was due to the fact that in the training phase it loads 256 images in its memory and in the test phase it just loads one image at the same time, so a lot less memory is required.

3.1 Dataset
The Face Detection Data Set and Benchmark (FDDB) Dataset [8] was used for the evaluation phase. This dataset...
is originally a benchmark test to test against other state-of-the-art face detectors, such as Viola Jones [15] or DDFD [3]. Because the focus of this research was not to outperform state-of-the-art face detectors, a custom benchmark was designed. The images in this dataset are real life examples of pictures where faces are present. 50 random images of the total of 28,000 images of the FDDB dataset were selected to perform the tests on. The reason so less images were selected to perform the test on was the fact that the time to test the 10 instances on these 50 images took about half a natural day. So when something went wrong, the time to restore the error was costly, because the test need to be done all over again.

3.2 Procedure
Firstly the evaluation dataset was checked by a human on how many faces are present in the images. The decision rule for classifying a face was: “If the facial keypoints can be identified for potential facial identification software, then it is a face”. So a face printed on a t-shirt will be counted as a correct face detection, the scope of this research is not to detect spoofs. The next phase in the evaluation procedure was to detect the faces in the evaluation dataset by the 10 versions of the face detectors. The results of these detection can be split up in two categories: true accepts and false accepts. A true accept is when the face detector recognized a face in a certain area of the image and in this same area there is also a face detected by a human. A false accept is when the face detector returns a face recognition for a certain area, but there is no face recognized in this area by a human. The plotted results of how many true and false accepts the detector returns is one of the results of this research.

3.3 Histograms
The second result of this research is to find out if the probability changes when there is more training data presented to the face detector. A histogram is created for all the versions of the face detectors. The histograms show how the probability changes for the different face detectors and if there is a progress in presenting more data to the face detector.

4. RESULTS
In appendix A the figures are present that corresponds to the results of this research. These results include three figures. Figure 7 shows the true and false accepts of every version of the face detectors. Figure 8 is almost similar to figure 7, except in this figure the difference between the true and false accepts are plotted, to show the performance gain of the different face detectors. Figure 9 shows the probability distribution of all the 10 versions of face detectors. The total faces counted by a human in the evaluation dataset was set at 75. As one can see in figure 7, the true accepts do not very much get affected when more data is presented. For example, the face detector with 10,000 positive examples in its training set, at 1000 iterations it detects 68 faces and at 5000 iterations it detects 69 faces. This is a total gain of one face, what is not a major gain. This can also be said of the face detector with 5,000 positive examples in its training set; at 1000 iterations it detects 57 faces and at 5000 iterations it detects 67, which is a gain of 10 faces. But the result that is more significant is the decrease of the false accepts. At 1000 iterations this model detects 28 false accepts and when it has done 5000 iteration, it detects only 8 false accepts. This is an absolute decrease of 20 false accepts. An interesting trend in both models is that the performance drops around 3,000-4,000 iterations. But overall, as can be seen in figure 8 the performance gains when more data is presented. For this benchmark test the correct accepts increases from 39 to 65 correct accepts. The model with 5000 positive examples outperforms the other model, possibly because the ratio positive/negative examples is smaller, so the detector knows better what especially is not a face. In figure 9 can also be seen that the histogram tends to a higher count of a higher probability as the training size increases. After 1000 iterations both models have a higher count of medium probabilities(0.4-0.6) than at 5000 iterations. After 3000 iterations the model with a training size of 10,000, the count of medium probabilities increases. This is related to the performance drop in figure 7. Because the CNN is a sort of black box, it can not be traced what happened at 3000 iterations to explain the performance drop.

5. CONCLUSIONS
In this research a CNN-based face detector is used to look if the size of the training data is of impact on the performance of a CNN-based face detector. There has been chosen to measure the training size in two dimensions if the face detector performs better. The first dimension is the training size of the batch that is presented to the face detector. The first batch consists of 5,000 positive and 500,000 negative examples and the second batch size consists of 10,000 positive and 500,000 negative examples. The second dimension that is used to measure the performance gain is how many iterations the face detector has gone through. An interval of 1,000 iterations was picked to a maximum of 5,000 iterations. The iterations can be seen as the more examples the face detector has used to train itself. A performance gain is observed when there is more training data presented as well as in the true and false accept, as in the probability distribution of the detectors. However, it does not outperforms the human eye or does not even reach status quo.

6. DISCUSSION
The result of this research is fairly limited to use the conclusions for a more general purpose. Because the training size of the different versions of the face detectors did not vary that much to really conclude if in most cases more training data results in a performance gain for CNN-based face/object detectors. The LFW dataset [7] did provide faces that were taken “in the wild”, but the pose vary much less than intended to. Also the ImageNet [1] dataset provided examples of random non-faces, this could be extended to not-randomized non-face examples. This can be reached by using non-face examples that are present in environments where faces are present, such as paintings, bodies or furniture. Also the non-face examples were not checked by hand if there were no faces present, so it could be that there were faces present, what contaminated the negative examples. Also the benchmark was limited with a size of 50 images, what influences the significance of the conclusions.

6.1 Future Work
For further research, there needs to be experimented with bigger differences of training data. On a quantitative and qualitative base, the dataset need to be changed, such that the type of negative examples also are present in real life situations where faces are present. But also with a bigger ratio difference between the datasets.
7. REFERENCES


APPENDIX

A. RESULTS

Figure 7. False and true Accepts of the instances

Figure 8. Difference between true and false Accepts of the instances
A.1 Histogram

Figure 9. Histograms of all the instances