Understanding the features a CNN uses to perform a classification between Caucasian and African-American faces by creating perturbated examples

Irene Font Peradejordi
Tilburg University
i.fontperadejordi@tilburguniversity.edu

Abstract

Neural Networks are capable of expressing complex patters in data. But as patterns are automatically discovered they can be difficult to interpret by humans. Understanding how neural networks learn has become a major concern among the scientific community. In this paper, I aimed to conduct a very modest experiment to understand what features a convolutional neural network uses to perform a simple binary classification task: classify between Caucasian and African-American faces extracted from the Face Place 3.0 public dataset. I reduce the range of possible features to three: (1) the facial expressions, (2) the shape and head position, (3) and the color. To identify which of these three is more relevant for the classification I created eight perturbating filters that pretend to challenge the features. By analyzing how the probability of each prediction varies in each case I concluded that the color is the main information the neural network learns to perform the task correctly. However, it is not the distribution of colors of the while image what it learns but the colors situated in the center of the image, where the faces are. Other minor conclusions are found and stated in at the end of the paper.

1 Introduction

Neural networks (NN) achieve high performance because of their ability of expressing complex patterns by computing a lot of parallel non-linear steps. But, as these patterns are automatically discovered by backpropagation via supervised learning, they can be difficult to interpret by humans. Understanding how neural networks learn has become a major concern among the scientific community.

A classic parable is often used to warn about the limitations of neural networks and the importance of controlling their input data to avoid possible biases. The parable goes as follows: The Army trained an artificial intelligence algorithm (aka CNN) to classify between American and Russian tanks. This algorithm happened to classify the task perfectly, with a 100% accuracy, only later analysts realized that American tanks had been photographed on a sunny day and the Russian tanks had been photographed on a cloudy day. The CNN was not using the tanks itself as features for the classification, as it was desired, but the brightness of the image. Although this parable is sometimes said to be an *urban legend* [The Neural Net Tank Urban Legend, 2011] as it is usually never sourced nor dated, the phenomenon has been a reason for concern for variety of researches. In 2017, Dr. Kosinski and Mr. Wang from Stanford University, created an algorithm that did successfully classify between gay and heterosexual people only using images from their faces extracted from an online dating platform. They used a deep neural network to extract features from 35,326 facial images and entered these into a logistic regression model. They achieved an accuracy of 91% for man and 83% for women, whereas the human judgement is only 61% and 54% accurate respectively [Wang & Kosinski, 2017]. Leaving aside the ethical issues raised and the potential negative consequences of this classifier, we will focus on another concern, more relevant for the task at hand, spotted by Dr. Cox, an Assistant Scientists at the University of Wisconsin. He stated that gay people tend to post higher-quality photos [Murphy, 2017] and that could be what the algorithm is truly looking at when performing the classification. There is an agreement in the scientific community that it is easy to teach a machine to see but not to understand what it has seen.

Not only data training quality is a concern, but also intentional attacks. In 2014, Szegedy and Goodfellow discovered the possibility to intentionally harm neural networks by applying a certain hardly perceptible perturbation or noise in the image. This noise is found by maximizing the network's prediction error. When this perturbated image is feed into the neural network, the algorithm misclassifies it with an extremely high probability. The perturbed examples are named "adversarial examples" [Szegedy & Goodfellow, 2014].

In this paper, I aimed to perform a very modest experiment to understand what features a convolutional neural network uses to perform a simple binary classification task: classify between Caucasian and African-American faces extracted from the Face Place 3.0 [Tarr, 2008] public dataset. Therefore, the research question addressed in this pa-

per reads as follows: Which features does a CNN classifier use when distinguishing between faces belonging to the Caucasian and the African-American race? I addressed this research question with three hypothesize in mind of what the CNN could be looking at: (1) the facial expressions, (2) the shape and head position, (3) and the color.

This study has been done by training a CNN and used it to predict perturbated images. I extracted the conclusions by analyzing how the probability of each prediction varies in each case. Eight different perturbations have been created in form of image filters, and some of them are image-level distortions while other are face-level distortions. Each perturbation is created to challenge an initial hypothesis. Some of these filters are inspired by the paper Unravelling Robustness of Deep Learning based Face Recognition Against Adversarial Attacks [Goswami, Ratha, Agarwal, Singh & Vatsa, 2018], which created a framework to evaluate robustness of deep learning-based face recognition engines. Others are product of my own creation to address this specific task.

2 The deep learning architecture

To start, I trained a convolutional neural network to classify between Caucasian and African-American faces. A CNN has been the chosen because they are a specialized kind of neural networks for professing data that has a known, grid-like topology. Image data can be thought of as a 2D grid pixels [Goodfellow, Bengio, Courville, 2017].

As the task at hand is very similar than the well-known classification of Cats vs Dogs, I used similar parameters than the hands-on-exercise we did in class. I constructed a neural network of seven hidden layers, five of which are convolutional layers with a rectified linear activation function (ReLu), followed by a max pooling layer. I used pooling layers to reduce the size of the representation, to speed the computation as well as to make the features that it detects a bit more robust [Andrew Ng, 2017].

The convolutional layers have either 32, 64, or 128 filters of 3x3. The last two layers are densely connected. One has 512 neurons to receive the output coming from the convolutional layers and the last one has just 1 neuron with a sigmoid activation function, as it is a binary classification and we are interested in one output per example.

A complex neural network structure is used to solve what could be considered as a simple binary classification problem. The deeper the neural network goes the more sophisticated the filters become. Therefore, increasing the level of complexity of the neural network means that it has more expressiveness and it is more capable of representing complex patterns. However, this does not avoid the CNN taking a shortcut and learning a very simple feature instead, having unnecessarily a complex structure.

3 Experimental method

3.1 Data

The Face Place 3.0 public database is used for the task. It was created in 2012 by the Tarrlab at Brown University, that is now at Carnegie Mellon University. The database includes multiple images for over 200 individuals of many different races with consistent lightening, multiple views, real emotions, and disguises. The images were taken with a standard resolution video camera. The images themselves are in jpeg format, 250x250 72 dpi 24-bit color. The database consists of images divided in five groups depending on the race: Asian, African-American, Caucasian, Hispanic, and Multiracial. All the images in the dataset had been preprocessed [Tarr, 2008]. All the images have had the background eliminated and only include the face and the hair of the person. That is why it was very appropriate for the task at hand.

The goal of the experiment was not to challenge a neural network to perform good in a difficult classification task but to understand how the classification is performed, the simple the problem is the better. Therefore, I used only two type of race groups: the Caucasian and African-American faces.

The Caucasian class had 3362 examples and the African-American class had 936 examples. In order to make them balanced, I choose 936 random examples from the Caucasian class.

I saved each class in two different folders. I exported each example into my code and convert them into an image array. After that, I stacked both arrays vertically, the one with the Caucasian examples and the other one with the African-American examples. Then, I created the arrays of labels (1 for Caucasians and 0 for African-Americans) following the same procedure.

To randomly mix the examples and split the data between the training and the test set, I used the *train_test_split* module from the Sklearn library. I set the test set to be 20% and the training set to be 80%. Therefore, the test had 375 examples and the train 1497 examples in total.

Before feeding this data to the CNN, I convert the pixels number, that varied between 0 and 255, to a number between 0 and 1, as the ReLu activation function performs better when the numbers are small.

Figure 1: Two examples from the Face Place 3.0 dataset. One example from each class. The Caucasian and the African-American class.

3.2 Experiment

The goal of this experiment is to identify which of the features are more relevant when classifying Caucasian and African-American faces. In other words, I aim to identify what is the CNN truly looking at. To do so, I created three hypothetical features that the CNN could be using in this particular task when trying to maximize its performance. These features go from more expressive to less expressive:

- Facial expressions
- Shape and head position
- Color

Then, I created adversarial examples that challenge these hypothesizes and extracted conclusions by observing which image perturbation(s) downsize(s) the most the CNN performance. To create the image perturbations, I gathered inspiration from the paper Unravelling Robustness of Deep Learning based Face Recognition Against Adversarial Attacks [Goswami, Ratha, Agarwal, Singh & Vatsa, 2018]. They identify that both the Grid and the Noise diminish the accuracy of a Face Recognition Classifier. I used their results as starting point.

Creating the perturbations

The perturbations were applied to 16 examples randomly selected from the test set, 8 from each class. Therefore, the neural network did never see these examples during the training process.

All the perturbations used are showcased in the Table 1. Each perturbation pretends to challenge one or two hypothesizes. The facial expressions will be challenged by the gaussian blur, the grid, the molino, the noise, and both of the permutations' filters. The shape and head position will be challenged by the gaussian blur, the *molino*, the 180º rotation, and both of the permutations` filters as well. Finally, the color will be challenged by the black and white, and the ellipse filters.

Perturbation / Features affected	Facial Expressions	Shape & position	Color
Black and White			x
Ellipse			x
Gaussian Blur	x	x	
Grid	x		
Molino	x	x	
Noise	x		
180º Rotate		x	
Permutation	x	x	
Double Permutation	x	x	
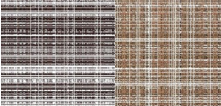
Table 1: Visual example of each perturbation, their names, and the features affected.

All the filters have been created using the software Adobe Photoshop, except for the permutations, which I used the NumPy's permutation(x) function. For further reference, I will detail the technical aspects for the creation of each filter. The black and white filter is a simple transformation from RGB to gray scale. The ellipse is a white circle (255, 255, 255) for African-American examples or a black circle (0, 0, 0) for Caucasian examples, with a transparency of 60%, positioned on top of the faces' examples. The gaussian blur filter has a constant radius of 10,0 pixels. The grid are random lines, both black and white, situated on top of the image. All the examples have the same grid. The *molino* deforms the image rotating it 150º starting from the center. The noise is a gaussian noise type of 15%. And lastly, a simple 180º rotation.

The premutation(x) function randomly permutes the pixels, eliminating all possible facial expressions and shape. Only the color information stays intact. The Figure 2 shows the histogram of the original image and the permutated image. Both of them have the same color distribution.

However, if x is a multi-dimensional array, it is only shuffled along its first index. That is why the pixels are allocated following a patter in the horizontal axis. As the intention of this filter is to eliminate all possible shape and head position, this pattern could be prejudicial for the task. Therefore, I rotated 90º the permutated images and I applied a second permutation. After the double permutation the pixels are distributed in both, the vertical and the horizontal axis.
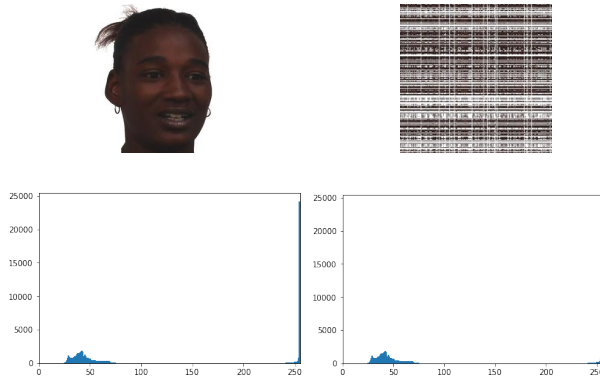
Figure 2: Histogram of an original image (left) and the double permuted image (right).

3.3 Hyper-parameter settings

The hyper-parameters chosen were motivated by the in-class hands-on-exercise Cats vs Dogs. The number of the batch size is 64 and 5 epochs are enough to achieve a high accuracy and minimize the risk of overfitting.

3.4 Evaluation criterion

The evaluation metrics used is accuracy. This is the number of correct predictions made divided by the total number of predictions made, multiplied by 100 to turn it into a percentage. Accuracy is a good metric due to both classes are balanced.

The model outputs a probability. This probability is transformed into binary, either 1 if $p > 0.5$ or 0 if $p < 0.5$. The test accuracy is calculated by comparing these binary predictions with the true classes.

However, the metrics I will be using to compare the results of the perturbated images are the raw predictions that we get before the binary conversion, due to the predictions are more informative.

4 Results

After the training, the neural network successfully achieved a 95.46% accuracy on the test set. Only 17 examples from the 375 belonging to the test set were misclassified. Interestingly enough, 16 of these 17 examples belonged to the African-American class.

In the Figure 3 it can be seen that the distribution of these classifications is quite homogeneous in both classes.
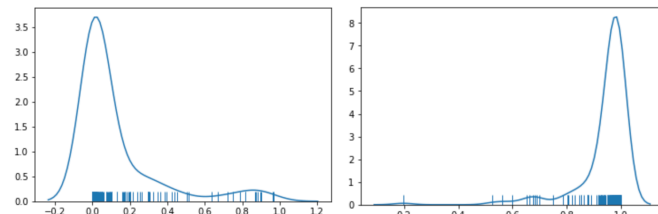

Figure 3: Distributions of predictions. African-American class (left). Caucasian class (right).

After, 16 examples where randomly selected from the test set. 8 from each class. I ensured that all of them were predicted correctly. The Table 2 show the results.

Then, the perturbations defined in the section 3.2 were applied to all these 16 images and predicted again. The results are showed in the Table 3.

5 General discussion

From the results presented in the previous section, we can conclude different things.

First, the hypothesis that the CNN learns the facial expressions is not true. Neither the *molino*, nor the noise have a significant impact on the predictions.

Secondly, we can conclude that the hypothesis that the CNN is looking at the shape and head position could be true. The gaussian blur filter has an impact at the shape of the head, and the 180º rotation modifies its position. Both of these filters have significant misclassifications.

Thirdly, we can conclude that the hypothesis that the CNN is looking at the color is partially true. While the ellipse filter is changing the color of the face only, it misclassifies significantly both of the classes –7 misclassifications out of 16–, concluding that this filter is the most adversarial of all. However, when the pixels are randomly permuted, the predictions for both classes shift to 0. Therefore, the CNN is not learning to predict one class or another depending on the colors of the whole picture but the colors where the face is located. This hypothesis is also supported by the fact that the images just permuted once have more color in the center of the image and, therefore, they are better classified than the image that suffered a double permutation.

It is also interesting to notice that the gaussian blur filter has a higher impact on the African-American class than on the Caucasian class. While the Caucasian class stays intact after the gaussian blur is applied –in some cases the prediction even improves–, the African-American class suffers significant variation, and even two out of eight misclassifications. However, the opposite happens with the black and white filter. While the Caucasian class suffers significantly, the black and white examples are sometimes classified with a higher probability.

It is important to mention that one of the weaknesses of this analysis is the number of examples that have been perturbated. Due to time and resources constraints, only 16 images have been edited. More tests are required in order to achieve a generalizable conclusion.

6 Conclusions

After the results obtained, I can conclude that the convolutional neural network uses the color as main classification feature to perform the task with high accuracy. However, it does not learn to look at the distribution of colors of the whole image but the colors in the center of the image, where the faces are. This is proved by the highly ratio of misclassifications when an ellipse of the opposite color is applied on the top of the examples faces and the results obtained after

	0	1	2	3	4	5	6	7
	correct label: 1.0 predict label: [1.] confidence: [0.9961946]	correct label: 1.0 predict label: [1.] confidence: [0.9851918]	correct label: 1.0 predict label: [1.] confidence: [0.9916613]	correct label: 1.0 predict label: [1.] confidence: [0.9891031]	correct label: 1.0 predict label: [1.] confidence: [0.97008824]	correct label: 1.0 predict label: [1.] confidence: [0.9912465]	correct label: 1.0 predict label: [1.] confidence: [0.96609384]	correct label: 1.0 predict label: [1.] confidence: [0.9894252]
	8	**9**	**10**	**11**	**12**	**13**	**14**	**15**
	correct label: 0.0 predict label: [0.] confidence: [0.02494373]	correct label: 0.0 predict label: [0.] confidence: [0.00061859]	correct label: 0.0 predict label: [0.] confidence: [0.00742696]	correct label: 0.0 predict label: [0.] confidence: [0.13385017]	correct label: 0.0 predict label: [0.] confidence: [0.00017722]	correct label: 0.0 predict label: [0.] confidence: [0.01679216]	correct label: 0.0 predict label: [0.] confidence: [0.00136057]	correct label: 0.0 predict label: [0.] confidence: [0.34570342]

Table 2: The 16 examples randomly selected from the test set and its predictions after feeding them into the model.

	ORIGINAL	bw	elipse	gaussian	grid	molino	noise	rotate	permutation1	permutation2
0	0.9962	0.9404	0.9789	0.9937	0.9952	0.9946	0.9897	0.9902	0.5955	0.0014
1	0.9852	0.6919	0.1472	0.9906	0.9848	0.9949	0.9818	0.9961	0.9109	0.0123
2	0.9917	0.8611	0.5872	0.991	0.9917	0.991	0.9912	0.9895	0.6863	0.0186
3	0.9891	0.93	0.7606	0.9883	0.9879	0.9966	0.9794	0.9911	0.3151	0.0003
4	0.9701	0.391	0.3484	0.976	0.98	0.9764	0.9806	0.7814	0.0379	0
5	0.9912	0.9528	0.7217	0.9898	0.9912	0.9928	0.9949	0.9954	0.9882	0.1761
6	0.9661	0.3883	0.0809	0.9902	0.9674	0.9832	0.911	0.78	0.0028	0.0002
7	0.9894	0.9319	0.6475	0.9873	0.9905	0.9924	0.9882	0.9818	0.4286	0.0038
8	0.0249	0.0133	0.9721	0.4062	0.0114	0.0874	0.0011	0.1467	0	0
9	0.0006	0.0005	0.0243	0.3795	0.0007	0.0012	0.0003	0.0404	0.0002	0
10	0.0074	0.0052	0.1538	0.1577	0.0033	0.0303	0.0002	0.0019	0	0
11	0.1339	0.0371	0.9919	0.6498	0.137	0.3782	0.0281	0.9313	0.0117	0.0011
12	0.0002	0.0002	0.0712	0.0282	0.0002	0.0002	0.0001	0.0076	0	0
13	0.0168	0.0096	0.9655	0.4032	0.0195	0.0041	0.0014	0.0215	0	0
14	0.0014	0.0008	0.0019	0.8578	0.0005	0.0171	0.0004	0.0773	0.0001	0
15	0.3457	0.0539	0.9878	0.8488	0.3189	0.82	0.0609	0.9345	0.005	0.0002

Table 3: Results of the 16 images after feeding them into the model with the perturbations applied.

the permutations, when the pixels are shuffled throughout the image.

Predictions made on African-American examples are in general easier to attack. In other words, they are less robust than the predictions made on the Caucasian class. However, when feeding black and white images to the neural network the probability on the African-American class increases.

The CNN is also sensitive to all the other perturbations but in a smaller scale.

References

The Neural Net Tank Urban Legend - Gwern.net [Video file]. (2019, March 27). Retrieved April 5, 2019, from https://www.gwern.net/Tanks

Andrew, N. G. (2017, November 7). *C4W1L09 Pooling Layers* [Video file]. Retrieved April 5, 2019, from https://www.youtube.com/watch?v=8oOgPUO-TBY

Goodfellow, I. J., Jonathon, J., & Christian, C. (2014, December 20). Explaining and Harnessing Adversarial Examples. Retrieved April 5, 2019, from https://arxiv.org/abs/1412.6572

Goswami, G., Ratha, N., Agarwal, A., Singh, R., & Vatsa, M. (2018, February 22). Unravelling Robustness of Deep Learning based Face Recognition Against Adversarial Attacks. Retrieved April 5, 2019, from https://arxiv.org/abs/1803.00401

Kurakin, A., Goodfellow, I. J., & Bengio, S. (2016, July 8). Adversarial examples in the physical world. Retrieved April 5, 2019, from https://arxiv.org/abs/1607.02533

Marcus, G. (2018, January 2). Deep Learning: A Critical Appraisal. Retrieved April 5, 2019, from https://arxiv.org/abs/1801.00631

Righi, G., Peissig, J. J., & Tarr, M. J. (2008). Face Place - The CNBC Wiki. Retrieved April 5, 2019, from http://wiki.cnbc.cmu.edu/Face_Place

Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Dumitru, E., Goodfellow, I. J., & Fergus, R. (2013, December 21). Intriguing properties of neural networks. Retrieved April 5, 2019, from https://arxiv.org/abs/1312.6199

The New York Times Company. (2017, October 20). *Why Stanford Researchers Tried to Create a "Gaydar" Machine* [Video file]. Retrieved April 5, 2019, from https://www.nytimes.com/2017/10/09/science/stanford-sexual-orientation-study.html

The New York Times Company. (2018, November 9). *Artificial Intelligence Hits the Barrier of Meaning* [Video file]. Retrieved April 5, 2019, from https://www.nytimes.com/2018/11/05/opinion/artificial-intelligence-machine-learning.html