An Approach to Facial Recognition Technology

Aarushi Upadhyaya
TJHSST
May 2019

Overview

This project focuses on creating facial recognition technology using advanced Artificial Intelligence concepts. Similar facial recognition technology has been implemented successfully in Google and Facebook. In the future, I plan to incorporate this technology into a facial recognition-based attendance system.

Motivation

In recent years, facial recognition technology has made groundbreaking strides and some facial recognition programs today even have success rates that rival that of a human. The creation of such astounding technology has numerous applications in our society and has the potential to revolutionize everyday life as we know it. Specifically, an implementation of such technology which I was interested in was a facial recognition-based attendance system. My intent for this project was to create the necessary facial recognition technology which will eventually be implemented into an attendance system. Throughout this process, I intended to familiarize myself with advanced Artificial Intelligence concepts, such as Convolutional Neural Networks (CNNs), and with handling image data.

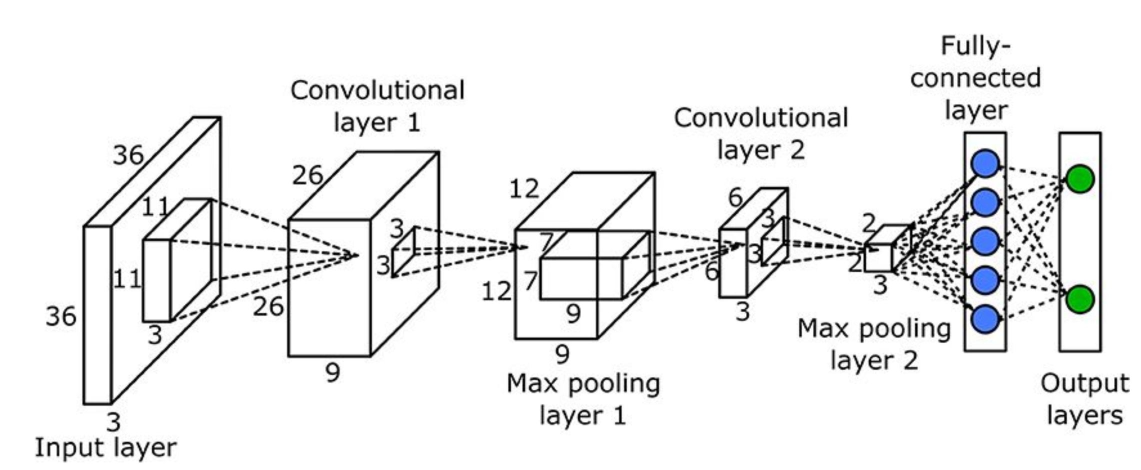
Background

Technologies similar to my method have been attempted before, the majority of which deal with machine learning algorithms. A previous solution to this is the DeepID method, implemented by researchers Yi Sun, Xiaogang Wang, and Xiaoou Tang [1][2]. The method I created incorporates some of the earlier steps involved in that method with a simpler face classification algorithm, to ensure the technology could be developed under my projected timeline of 10 months. Furthermore, throughout the year, all of my code was developed and tested inside of an anaconda environment. The data used was obtained from the Olivetti Research Laboratory. The dataset contained 400 images total in 92x112 dimensions

Methodology

The approach I took to solving this problem is mainly divided into 4 steps, outlined below:

1. **Identify key points**
   These points focused around easily distinguishable regions of the face, such as the eyes, nose and mouth, and were identified through the implementation of a pre-trained facial landmark predictor using a python image processing library, dlib.
2. **Implement a similarity transform**
   In this step, the main goal is to have a template image, with the key points identified, and to transform a sample image, such that its key points match up to that of the template image. I was able to create this code, using parts of an implementation of a similar transformation project, which I found on GitHub while researching.
3. **Extract patches**
   I chose to extract 8 patches, using OpenCV, from the facial image. Each patch that was extracted focused on a different aspect of the face and a different set of key points, such as the nose, or multiple sets of key points, such as an eye and the nose. This allowed for more variance to occur and also ensured that each section of the face was being represented equally.
4. **Use Convolutional Neural Networks to confirm face's identity**
   A CNN, or a deep learning algorithm, was used to analyze the input images and made deductions regarding the identity of the facial image input. The CNN was implemented using the Python libraries TensorFlow and Keras. The data to train and test the CNN was passed in using another Python library, Numpy. After a vigorous training process, the CNN was able to successfully determine a facial image's identity.



Results

The main results finding and accuracy determination occurred at the end on the project. Ultimately, after training, the technology had about a 95 percent accuracy rate and around a 0.17 loss rate - a value measuring inconsistencies between the identity generated by the CNN and the image's actual identity - on previously seen data, and approximately 75 accuracy rate and about a 1.5 loss rate on new data which the CNN hadn't seen before. In this case, the measured loss is relatively low. Furthermore, in the context of the future application of this technology, the 75 percent statistic is more relevant as new pictures will have to be taken in and corresponded with its identity.
The results are quite similar to that of the current average success rate of facial recognition technology available[3]. However, there is still a large margin for improvement in my method, as many algorithms, such as those developed by Google and Facebook mentioned in the Introduction of this paper, have almost perfect accuracy.

Conclusions

This study accomplished its original goal, which was to create facial recognition technology through the implementation of CNNs. In the future, I plan to expand upon this technology by adding many more subjects to the image dataset, as with more variance and increased training, the CNN will become more adept at distinguishing between subjects' features. This technology can be incorporated into the project idea discussed in the Motivation section - an automated attendance system - and can easily be altered to accept image input through a video feed.

References

[1]Yi Sun, Xiaogang Wang, and Xiaoou Tang. Deep learning face representation from predicting 10,000 classes.
[2]Switching eds: Face swapping with python, dlib, and opencv, 2015.
[3]The top 7 trends for facial recognition in 2019, 2019.