Gender recognition by hand with deep neural networks

Naveen Hosaagrahara Srinivas
Matriculation number: 916612
Università Degli Studi Di Milano Nov 2019

Course Assessment Project for
Informatics for industrial applications and robotics
Vision in industrial applications and robotics

prof.
Fabio Scotti

1 Abstract

Gender is the primary parameter in the identification of a user. Its uses are with the introduction of Convolutional Neural Networks in computer vision, Image analysis has found deeper dimensions[1]. CNN combined with Image processing can achieve prediction of Gender from the image of a subject [2]. In the following project, Features are extracted from a labeled hands image dataset, later these features called feature map are used as an input to a neural network to classify Gender.

2 Introduction

Gender is the primary parameter in the identification of a user. Its uses are important in the field of Security, Social Networking, etc. In recent times with increasing demand for managing security aspects, identifying users on physical parameters like age, gender, emotion with minimal information is challenging. Social Networking and e-commerce companies invest a huge amount of time and resources to estimate gender to provide appropriate suggestions of products and services to their users. The progress in the field of computer vision has been amplified with the introduction of neural networks. Convolutional Networks have introduced a new paradigm in the field of image analysis. Profiling of users, using images or video footage's can be achieved with the help of image processing and neural networks.

In this project, we have a large dataset of hand images, subsequently referred to as 11K Hands dataset, As we have a large dataset of hand images, we can effectively train a CNN. To that end, we train a deep learning-based model for gender recognition using the proposed 11K Hands dataset.
Keywords— *Gender recognition*, *neural networks*, *deep learning*

3 Development tools

3.1 Programming language

The implementation of the algorithm for recognizing gender is developed using Python as a programming language and different open-source machine learning libraries. Python is an interpreted, general-purpose high-level programming language. It has gained a lot of popularity in data science and in designing machine learning algorithms. The choice of this programming language was influenced by its simplicity, readability and the huge community contributing to providing a wide range of implementations of different algorithms. Also, a lot of libraries that were useful to this project are provided as open-source packages that can easily be included in the project.

3.2 Google Colaboratory

Google colab is a cloud-based workspace, similar to the jupyter notebook and it is designed for data science tasks. With every colab session is provided a virtual machine running 13 GB of RAM and either a CPU, GPU, or TPU processor. Colaboratory provides means to write and execute code, save and share analyses, and access to powerful computing resources, and above all, it can be accessed for free from any browser. We took advantage of the Google

Colab resources to train the neural network using GPU. After the training is completed we saved the model in a file that can be loaded later in our local machine.

3.3 Open source Libraries

A lot of libraries have been developed in Python, especially in the field of Machine Learning. They make it easier to implement different machine learning algorithms in a few lines of code. In this project, we used many open source libraries. Below we give a brief description for each of them.

3.3.1 TensorFlow

It is an open-source library developed by Google Brain team and it is designed for numerical computations that make machine learning faster and easier. TensorFlow carries out the calculations by dataflow graphs (structures) that describe how data moves through a graph or a series of processing nodes. Nodes in this graph represent mathematical operations, and edges between nodes are multidimensional data arrays or as referred by the authors of the library, tensors. In our project, TensorFlow is used to create, train and evaluate the model.

3.3.2 NumPy

It is the fundamental package for scientific computing in Python. It is a Python library that provides a multidimensional array object, various derived objects (such as masked arrays and matrices), and an assortment of routines for fast operations on arrays, including mathematical, logical, shape manipulation, sorting, selecting, I/O, discrete Fourier transforms, basic linear algebra, basic statistical operations, random simulation and much more [4]. In our project, we used it to handle images and labels that are fed to the model. TensorFlow provides a lot of APIs make it possible to integrate NumPy objects.

3.3.3 Matplotlib

It is a Python 2D plotting library that provides API to display quality figures in a variety of hardcopy formats and interactive environments across platforms. Matplotlib can be used in Python scripts, the Python and IPython shells, the Jupyter notebook, web application servers, and other graphical user interface toolkits. In our project, we used it to display figures and graphs about our model performance.

3.3.4 OpenCV - (Open Source Computer Vision Library)

It is an open-source computer vision and machine learning software library. OpenCV was built to provide a common infrastructure for computer vision applications and to accelerate the use of machine perception in commercial products. In our project, it is used for different image manipulations.

3.3.5 Pandas

It is a popular Python library for data science, and it offers powerful, expressive and flexible data structures that make data manipulation and analysis easy, among many other things. We used pandas to collect and save information about our data that later is used more easily.

4 Developed method

4.1 Dataset

The 11k Hands dataset, a collection of 11,076 hand images (1600 x 1200 pixels) of 190 subjects, of varying ages between 18 - 75 years old. Each subject was asked to open and close his fingers of the right and left hands. Each hand was photographed from both dorsal and palmar sides with uniform white background and placed approximately in the same distance from the camera.

Figure 1: A sample of the dataset Male hands (https://goo.gl/rQJndd)

Figure 2: A sample of the dataset Female hands (https://goo.gl/rQJndd)

4.2 Development Process

How machine learning works is that the model will guess the relationship between the numbers. When training, It will then calculate how good or how bad that guess is, using the loss function. Then it will use the optimizer function to generate another guess. The logic is that the combination of these two functions will slowly get us closer and closer to the correct formula. In our case, it will go through that loop 50 times, making a guess, calculating how accurate that guess is, and then using that optimizer to enhance that guess, and so on. The neural network will be initialized with random values at the starting.

The dataset we have here that has several thousand images of male and female hands. Will train a neural network with this data.

First of all, we need to download the zip files containing the data. One has a training set and the other has a testing and validation set. In python, we can unzip a file with a zip file library, and we unzip them to a temp directory. This creates folders with subfolders of each of our categories. When training in TensorFlow using an image data generator, it will automatically label the images based on the name of their parent directory. So we don't need to create labels for the images.

Then we have 4 layers of convolutions each with MaxPooling before feeding into the dense layer and we use the Dropout, the dropout is a little trick to improve the efficiency of a neural network by throwing away some of the neurons.

Behind the convolution neural network, it filters the image before training the deep neural network. After filtering the images, features within the images could then come to the forefront and you would spot those features to identify something.

What is the filter? A filter is simply a set of multipliers and it applies to image and this can be combined with something called pooling, which groups up the pixels in the image and filters them down to a subset. So where did these filters come from? That's the magic of a convolution neural network, they are learned. They are just parameters like those in the neurons of a neural network.

So as our image is fed into the convolutional layer, several randomly initialized filters will pass over the image. The results of these are fed into the next layer and matching is performed by the neural network. And over time, the filters that give us the image outputs that give the best matches will be learned and the process is called feature extraction. We flattened the input that's fed into a dense layer that in turn in fed into the final dense layer that is our output and in our case it has 2 categories of male and female.

We initiate the convolution layer first and tells it to generate 64 filters with parameter input shape (150, 150, 3). So it will generate 64 filters and multiply each of them across the image, then each epoch, it will figure out which filleters give the best signals to help match the images to their labels in much the same way it learned which parameters worked best in the dense layer. As I mentioned earlier our neural network will act like a filter, which takes in a 150*150 set of pixels and outputs male or female. Each layer has a different filter size like 64,128,512 and so on. For example, take 128? Well, think of it like this, we are going to have 128 functions, each one of which has parameters inside of it. Let's call this f0 through f127.

What we want is that when the pixels of the image gets fed into them, one by one, that the combination of all of these functions will output the correct value. To do that, the computer will need to figure out the parameters inside of these functions to get that result. And it will then extend this to all of the other items in the dataset. The logic is, once it has done this, then it should be able to recognize gender.

The max-pooling to compress the image and enhance the feature and we can stack convolutional layers on top of each other to break down the images and try to learn from very abstract features

We use many Activation functions :

A. Relu function or rectified linear unit: it simple as returning a value if it is greater than zero. So if that function had zero or less as output, it just gets filtered out.

B. Softmax: has the effect of picking the biggest number in a set. The output layer this network has 2 items in it. Representing the probability that we are looking at that specific gender. Softmax finds the largest and set to 1 and the rest is 0.

We will compile the neural network and then we can fit the data with the model.fit call. Note that we don't have a label that is because we are using the generator, it's inferring the labels from the parent directories of both the training and the validation datasets.

We train the model with fit function and input train images and train labels with 50 epochs. We test how well our model performs with the data model never seen before. We can do that test by passing them to the evaluation method. Then we can get prediction back for new images by calling model.predict.

5 Classification Results

Classification Results				
CNN Model	Training Accuracy	Training Loss	Validation Accuracy	Validation Loss
(Average)	0.83	0.42	0.85	0.30

Figure 3: Classification Results

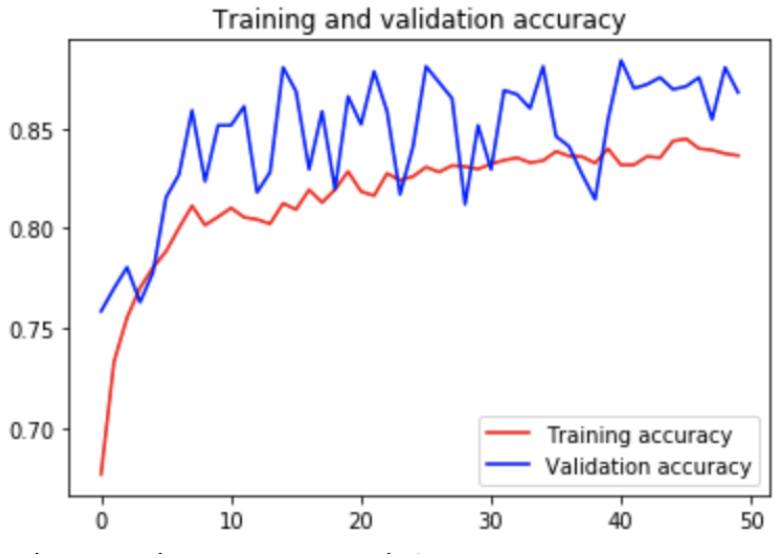

Figure 4: Training and Validation Accuracy

6 Conclusion

I am able to achieve an average of 85 percent validation accuracy and 83 percent of training accuracy. I retied to vary the number of filters in the convolution neural network in each layer, then the result as expecting getting better by a small number on each increment but on the other hand it was too expensive in time. Sometimes training accuracy was way less then validation accuracy on that time image augmentation is a good idea to match the accuracy.

As I observed some of the female hand data set has Nile polish on, in the future work we can use that as a new dimension to increase the accuracy of the model. We can even go deeper and categorize the left or right hand of the person but due to lack of data sample, I couldn't do that.

7 References

[1] A. NG, “Convolutional Neural Networks (Course 4 of the Deep Learning Specialization),” 2017. [YouTube Source].

[2] A. Anand, R. D. Labati, A. Genovese, E. Munoz, V. Piuri, and F. Scotti, “Age estimation based on face images and pre-trained convolutional neural networks,” in 2017 IEEE Symposium Series on Computational Intelligence (SSCI), pp. 1–7, Nov 2017.

[3]TensorFlow Documentation – https://www.tensorflow.org/overview

[4] NumPy documentation - https://docs.scipy.org/doc/numpy-1.13.0/user/whatisnumpy.html

[5] Yaw Chang, Yishi Wang, Karl Ricanek, Cuixain Chen, ”Feature selection for improved automatic gender classification”, Computational Intelligence in Biometrics and Identity Management (CIBIM) 2011 IEEE Workshop on, pp. 29-35, 2011.

[6] Robust Biometric Recognition From Palm Depth Images for Gloved HandsNguyen, Binh P.Tay, Wei-LiangChui, Chee-KongIEEE Transactions on Human-Machine Systems

[7] Single-sensor hand-vein multimodal biometric recognition using multiscale deep pyramidal approachBhilare, ShrutiJaswal, GauravKanhangad, VivekNigam, AdityaMachine Vision and Applications 2018

[8] Robust Multi biometric Recognition Using Face and Ear Images-Boodoo, Nazmeen BibiSubramanian, R. K.arxiv 2009

[9] https://link.springer.com/article/10.1007

[10] Deep learning with python - francois chollet - Text book