International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 07 Issue: 04 | Apr 2020 www.irjet.net p-ISSN: 2395-0072

GENDER PREDICTION WITH MINIMAL FEATURES USING A CNN

Francis S Xavier[1], DVS. Praharshitha[2], T. Harini Veena[3], N. Chaitanya Srikar[4]

[1]*Assistant Professor, Dept. of EECE, GITAM, Hyderabad, India*
[2]*Student, Dept. of EECE, GITAM, Hyderabad, India*
[3]*Student Dept. of EECE, GITAM, Hyderabad, India*
[4]*Student, Dept. of EECE, GITAM, Hyderabad, India*

---***---

Abstract - *This paper aims to analyze the fingerprints of individuals and determine whether or not they belong to a male or a female. This paper also highlights the importance of convolutional neural network over other neural networks in simplifying the process of identifying the gender using fingerprints, by trying to fixate the number of features extracted from it to a minimal amount. Convolutional Neural Networks have proven to work effectively in the domain of image recognition, processing and image classification. Using CNN we can subtly identify faces or other objects easily. These extracted features are ridge endings and ridge bifurcations which after extraction from the fingerprints are passed to the CNN. The fingers used for the accomplishment of this project are specifically index fingers, ring fingers and middle fingers. The data sets and test sets include a total number of 1505 of male fingerprints and 1146 female fingerprints. The gender was correctly predicted with an accuracy of 71% when tested which suggests the implementation and possible usage in forensic, security and other related departments.*

1. INTRODUCTION

Analyzing a fingerprint and extracting information from it is very important for crime fighting, security and in several other fields. Although fingerprints are widely used for identification, research has proved that fingerprints can also be used for gender prediction. Gender prediction using a fingerprint opens up a wide range of applications such as if the gender is predicted from a fingerprint obtained from a crime scene it can narrow down the search by 50%. Gender prediction from fingerprints is carried out in general by ridge densities which is found to be more in female than the male counterparts. There are several features that are much easier to extract than the ridge density. We have tried to establish gender based on minimal number of features that are extremely easy to extract. These features are “Ridge endings” and “Ridge Bifurcations”, validation studies have proven that there is a noticeable difference in these features between the genders [1].

These features are extracted from the fingerprint and passed on to “MobileNet” neural network. The model was trained on the images with the features marked. Overall accuracy of nearly 60% was achieved when just using the two features, ridge endings and ridge bifurcations.

1.1 Methodology:

The entire process is illustrated as in Fig 1. The dataset is first collected, this data set is pre-processed, to remove bad quality images and then enhanced. These images are then sent for feature extraction.

Finally a neural network is trained using these images and finally tested.

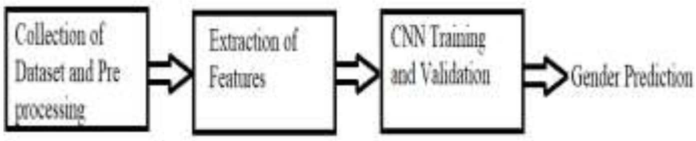

Fig -1 Gender Prediction

2. MODIFICATION AND ENHANCEMENT OF FINGERPRINTS

The fingerprints utilized are acquired from two sources. One was from the openly available Fingerprint data from Kaggle. Another dataset was obtained from our fellow students in the college. The reason for obtaining a secondary dataset for was for consistency. The dataset was quality controlled and fingerprints were obtained in a much better condition. This secondary dataset only had 80 fingerprints. These included Index, Middle and Ring fingers. 35 female and 45 male fingerprints were collected. The dataset from Kaggle has 1117 Female fingerprints and 1460 Male fingerprints. The fingerprints were sorted and some extremely deformed ones were removed.

A fingerprint taken from the sensor consists of lot of redundant data. This data makes it harder to achieve the desired results. The fingerprints are made to undergo a certain preprocessing steps before the Neural Network is trained with them. These are

2.1 Binarization:

The various levels of grey, white and black are converted to either pure black or pure white. That is each pixel is given a value of either 0 or 255. This results in the image being much clearer and noise is removed. Fig-2 Shows an example of image before and after binarization.

International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 07 Issue: 04 | Apr 2020 www.irjet.net p-ISSN: 2395-0072

Before After

Fig - 2 Binarization

2.2 Enhancement:

Although binarization does help in reducing the size and improving the visibilities of features, there is still a lot of redundant information, as in, the ridges are several pixels wide. The ridges in the images are thinned down to a single pixel, reducing the size of the image, while maintaining the information unaltered.

Before After

Fig-3 Enhancement

2.3 Ridge Endings:

After the image has been enhanced, the image is sent for feature extraction. The features that are being used are ridge endings and ridge bifurcations. Ridge endings are determined by looking through the image as 3x3 matrix, where one ridge ending would have no more than 2 adjacent elements, colored white and remaining all should be colored black.

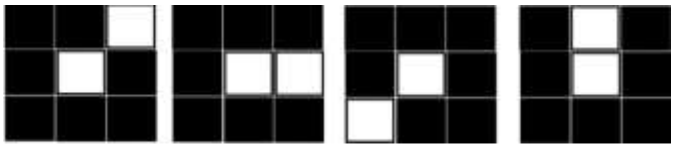

Fig-4 Some possible ridge ending combinations

2.4 Ridge Bifurcations:

Ridge bifurcations are determined by a similar principle as that of the Ridge endings, ridge bifurcations are identified by finding out the white pixels that have 2 neighboring white pixels. Fig-5 illustrates the following. Fig-6 illustrates the marked feature extracted images. The ridge endings and ridge bifurcations are marked on the images, with colored rectangles. Ridge Bifurcations are marked using blue rectangles and Ridge endings are marked using the yellow rectangles.

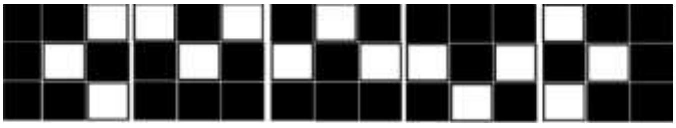

Fig-5 Ridge Bifurcation

Before After

Fig-6 Feature Extraction

3. SELECTION AND IMPLEMENTATION OF NEURAL NETWORK

The neural network selections play an important part as the number of features are being used a very highly limited. Mobile Net uses a concept called "Depth Wise Separable Convolution". This in turn is made up of two layers called "Depth wise convolutions" and "Pointwise Convolutions". These layers use Batch Normalization and RELU activation function. Fig-6 Shows the summary of the model.

Type / Stride	Filter Shape	Input Size
Conv / s2	$3 \times 3 \times 3 \times 32$	$224 \times 224 \times 3$
Conv dw / s1	$3 \times 3 \times 32$ dw	$112 \times 112 \times 32$
Conv / s1	$1 \times 1 \times 32 \times 64$	$112 \times 112 \times 32$
Conv dw / s2	$3 \times 3 \times 64$ dw	$112 \times 112 \times 64$
Conv / s1	$1 \times 1 \times 64 \times 128$	$56 \times 56 \times 64$
Conv dw / s1	$3 \times 3 \times 128$ dw	$56 \times 56 \times 128$
Conv / s1	$1 \times 1 \times 128 \times 128$	$56 \times 56 \times 128$
Conv dw / s2	$3 \times 3 \times 128$ dw	$56 \times 56 \times 128$
Conv / s1	$1 \times 1 \times 128 \times 256$	$28 \times 28 \times 128$
Conv dw / s1	$3 \times 3 \times 256$ dw	$28 \times 28 \times 256$
Conv / s1	$1 \times 1 \times 256 \times 256$	$28 \times 28 \times 256$
Conv dw / s2	$3 \times 3 \times 256$ dw	$28 \times 28 \times 256$
Conv / s1	$1 \times 1 \times 256 \times 512$	$14 \times 14 \times 256$
5× Conv dw / s1 Conv / s1	$3 \times 3 \times 512$ dw $1 \times 1 \times 512 \times 512$	$14 \times 14 \times 512$ $14 \times 14 \times 512$
Conv dw / s2	$3 \times 3 \times 512$ dw	$14 \times 14 \times 512$
Conv / s1	$1 \times 1 \times 512 \times 1024$	$7 \times 7 \times 512$
Conv dw / s2	$3 \times 3 \times 1024$ dw	$7 \times 7 \times 1024$
Conv / s1	$1 \times 1 \times 1024 \times 1024$	$7 \times 7 \times 1024$
Avg Pool / s1	Pool 7×7	$7 \times 7 \times 1024$
FC / s1	1024×1000	$1 \times 1 \times 1024$
Softmax / s1	Classifier	$1 \times 1 \times 1000$

Fig-7 Neural Network summary

3.1 Depthwise seperable convolutions:

Depthwise seperable convolutions break down normal convolutions into two separate steps. A Normal 2D convolution takes a filter of a particular size and slides it over the entire image to generate a feature map. Each Conv2d layer takes in several of these filters and repeats the process. Each image although 2D, becomes a 3D representation as each channel, the three colors are seperated. For instance if we consider a 244x244x3 image which is basically a 244x244 RGB image, convolved with a 3x3 filter, for each channel we make 3x3x3 = 27 multiplications each time the filter moves one step. In general a lot of these filters are stacked together so if 256 such filters are applied the above calculation repeats for 256 times. Thus filter array is represented as for example 3x3x3x256(Height, width, input channels, number of output images). This is changed in Depthwise seperable convolutions. It occurs in two steps instead of one large step which requires a lot of computational power. Fig- 8(a) shows the illustration of Normal Convolutions.

3.2 Depthwise Convolutions:

Depthwise convolution is responsible for creation of individual images for each channel, ie it uses a three 3x3x1 filters one for each channel and then these are stacked to form one ouput. This output of this convolution is then passed on to the Point Wise convolutions. Fig 8(b) shows the illustration of Depthwise convoltion .

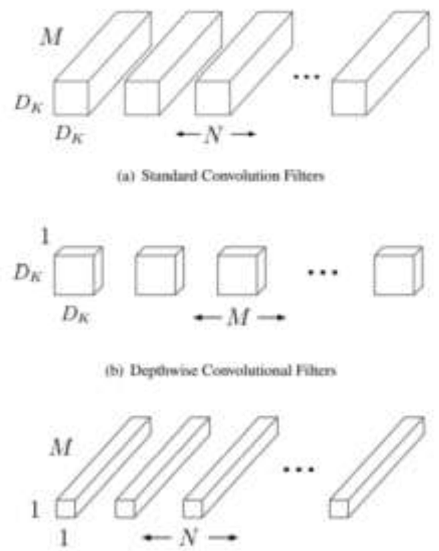

Fig-8 Convolution and Depthwise seperable convolution illustration

3.3 Point wise convolutions:

After the depth wise convolutions are done we have one output in all three channels. We need to expand this output to 256. This is done by the point wise convolutions. Pointwise convolutions are only work with one point, thus the name. These are of dimensions 1x1x256 in this case. The one output is then expanded to 256. Point wise convolutions are illustrated in Fig-8(c). Thus completing the work of one Normal Convolution. The training data is the dataset obtained from Kaggle after marking all ridge bifurcations and ridge endings on these images. A 10% of these photos are taken as Validation set and 90% acts as the Training set. The neural Network is trained with the images after the features were extracted and marked on the images. The loss function of the neural network although higher than many is still small enough to give satisfactory results.

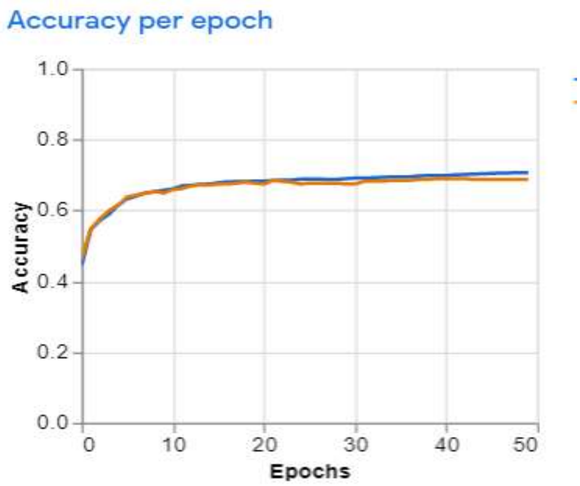

Chart-1: Training and Validation Accuracy while training per epoch.

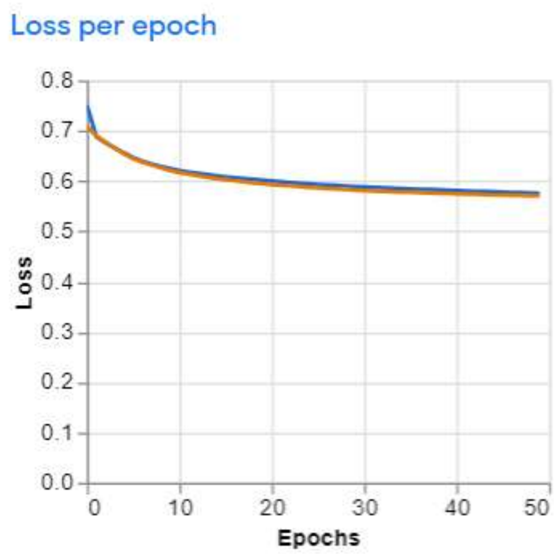

Chart-2: Loss in Validation and Training sets per epoch.

As shown in the Chart-1. The accuracy reached 72% on training data and 68% on validation data. The data showed that the predictions were accurate enough to make judgements regarding the gender based on just the two features.

Chart-2 shows the plot of loss per epoch for both training and validation sets of data.

4. RESULTS AND TESTING:

The entire project was tested using R307 Fingerprint module. This module is connected to a laptop using the USB-TTL UART converter. USB-TTL UART converter is used to create virtual COM ports and initiate Serial Communication. Due to unavailability of a USB-TTL UART device, an Arduino UNO was used instead. Fig-9 Shows the circuit for using Arduino UNO as a USB-TTL UART converter with the R307 finger print module.

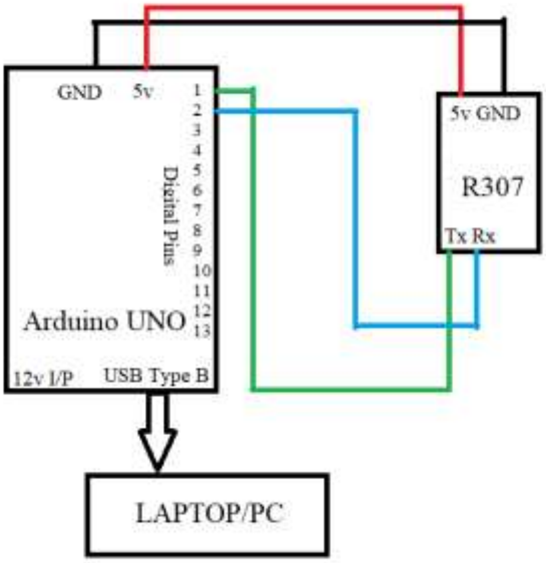

Fig-9 Arduino UNO as USB-TTL converter and R307 fingerprint module

The tool called SFG demo is used for interacting with R307 module, this tool is used to send commands to the module and extract fingerprints. The tool provides options like taking a new fingerprint image or storing a fingerprint in the module's memory. The tool can also be used to check if the fingerprint matches with one already stored in the memory. Fig-10 shows the interface of the tool.

Fig-10 SFG Demo Tool

Before testing on the new users, the model is tested on collected fingerprints, the results are as shown in Chart-3. Out of the 35 female fingerprints, 25 were accurately predicted, giving us 73% percent accuracy of prediction. Out of the 45 male fingerprints, 30 were accurately predicted, giving us a 67% accuracy. These results are inline with the Accuracy shown during the training of the neural network. Using the hardware shown in Fig-9, the model was tested with another random set of 10 people, out of which 6 were Male and 4 were Female. The model predicted 6 out of 10 cases correctly, giving an accuracy of 60%.

It is proven from the above testing that the genders can be identified with only 2 features extracted along with the power of CNNs, which are specifically designed to identify delicate features of an image and make predictions. In order to test the dependency of the model on the 2 features extracted, the model is trained on just the enhanced fingerprints without marking the features on it. The accuracy of this model came out to be below 50% proving that the features extracted made significant contribution to the efficiency of the model.

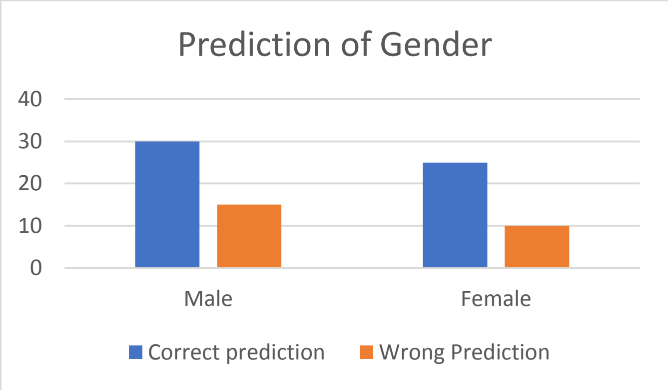

Chart-3: Correct and Wrong predictions for Male and Female

REFERENCES

[1] Thakar, M.K., Kaur, P. & Sharma, T. Validation studies on gender determination from fingerprints with special emphasis on ridge characteristics. *Egypt J Forensic Sci* **8,** 20 (2018).

[2] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. arXiv:1704.04861, (2017) .

[3] Hong, L., Wan, Y., and Jain, A. K. 'Fingerprint image enhancement: Algorithm and performance evaluation'. IEEE Transactions on Pattern Analysis and Machine Intelligence 20, 8 (1998), pp 777-789.

[4] Duth, Sudharshan & Mirashi, Megha. (2019). Fingerprint Based Gender Classification using ANN. 2249-8958.

[5] A.s, Falohun & Deborah, Fenwa & Ajala, Faten. (2016). A Fingerprint-based Age and Gender Detector System using Fingerprint Pattern Analysis. International Journal of Computer Applications. 136. 975-8887. 10.5120/ijca2016908474