https://doi.org/10.36375/prepare_u.a61

Classification of Gender from Human Facial Images using Convolutional Neural Networks

Devjyoti Saha, Diptangshu De, Pratick Ghosh, Sourish Sengupta, and Tripti Majumdar

Department of Computer Science and Engineering
Bengal Institute of Technology
1 No. Govt. Colony, P.O. Hadia, Kolkata – 700153
{Corresponding author’s email: devjyoti1998@gmail.com}

Abstract - It's a fairly simple task for humans to determine the gender of an individual using certain facial features, although it is difficult for machines to perform an equivalent task. Within the past decade, unimaginable steps have been taken to automatically predict the gender from a face image. The human face has certain distinctive features such as eyes, nose, lips, etc., which can be analyzed to classify humans into two basic genders: Male and Female. This project aims at achieving a similar goal of detecting gender from face images. The basic tool used in the project is Convolutional Neural Network (CNN) along with the use of the Programming language Python. In recent years, face detection has achieved considerable attention from researchers in biometrics, pattern recognition, and computer vision groups. There are countless security and forensic applications requiring the use of face recognition technologies which have motivated us to explore this area and start with this project.

Keywords – CNN; Gender; Machine Learning; Python; Deep Learning

I. INTRODUCTION

Speculating the gender from face images using automated techniques is becoming progressively remarkable as far as law enforcement and intelligence agencies are concerned. But, it is not an easy task to implement automated techniques as machines are not as intuitive when compared to humans in recognizing gender from face images. It has been observed that the gender of a human being can be determined by studying facial features such as eyes, nose, lips, etc. This project revolves around the idea of determining the gender by studying then above stated features employing automated techniques from the given face images. Deep learning techniques have been brought to play and one remarkable technology used is the Convolutional Neural Network (CNN).

This project has a wide range of applications ranging from countering terrorism to voter identification. The predominant technologies used in this project include Machine Learning: Supervised Learning, Image Processing, and Deep Learning: Convolutional Neural Network and Deep Learning. Supervised Learning can be defined as a machine learning technique where the input is mapped to the output with the help of training data consisting of input-output pairs. TensorFlow is an open-source library that is used for mathematical computation, dataflow programming, and various machine learning applications. Convolutional Neural Network (CNN) is one of the most prevalent algorithms that has gained a high reputation in image feature extraction [2].

II. LITERATURE SURVEY

A system that classifies the detected and aligned face images based on the gender was presented in E. Makinen et al. [1]. From this paper, it was found that the manual alignment method provides better classification rates than the automatic alignment method. It was also found that different input image sizes did not affect the classification accuracy rates. A new architecture for face image classification named unsupervised CNN was introduced by S. U. Rehman et al. [2]. A CNN that handles multitask (i.e. Facial detection and emotional classification) is made by merging CNN with other modules and algorithms. A hybrid deep CNN and RNN (Recurrent Neural Network) model was introduced by N. Jain et al. [4]. This model aims to improve the overall result of face detection. MI Facial Expression and JAFFE dataset were used to evaluate the model. A convolutional network architecture was proposed by G. Levi et al. [5] that classified the age and gender with small amounts of data. The Adience Benchmark was used to train the model. A system in which a real-time automatic facial expression system was designed was proposed by S. Turabzadeh et al. [6]. It was implemented and tested on an embedded device which could be the first step for a specific facial expression recognition chip for a social robot. MATLAB was first used to build and simulate the system and then it was built on an embedded system. The hardship of performing automatic prediction of age, gender and ethnicity on the East Asian Population using a Convolutional Neural Network (CNN) was explored by N. Srinivas et al. [3]. A fine-grained ethnicity has predictions based on a refined categorization of the human population (Chinese, Japanese, Korean, etc.). Previous results suggest that the most critical job is to predict the fine-grained ethnicity of a person, followed by age and lastly gender. An automated recognition system for age, gender and emotion was presented by A. Dehghan et al. [7] that was trained using deep neural network. At the ImageNet LSVRC-2010 contest, A. Krizhevskyetal. [8] presented a paper which suggested segregation of 1.2 million images into 1000 different categories with the help of a deep Convolutional neural network. The results which were obtained suggested that supervised learning can deliver exceptional accuracies. Some datasets have annotations on the face images which are not considered to be of any use for face recognition. Some papers have also used RNN but it is not applicable for our project as the RNN takes text or speech as an input whereas we required an image to be as the input. Hence, CNN is chosen over RNN for the sake of our project. Some papers also suggest the use of unsupervised CNN, but, for this project supervised learning is more appropriate. The UTKFace dataset is used as a dataset for the project.

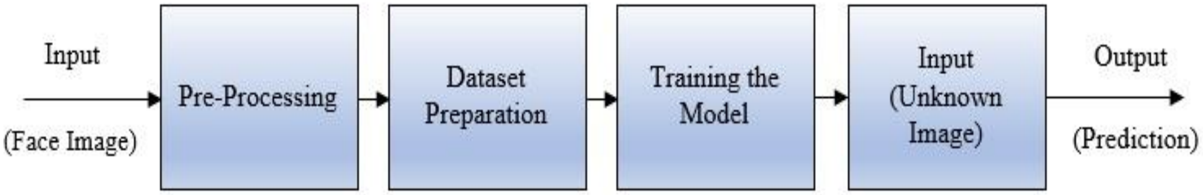


Fig-1: Basic Block Diagram

III. ARCHITECTURE

For the purpose of identification of gender, the dataset images are fed into the algorithm. For classification, the UTKFace dataset is used and Convolutional Neural Network (CNN) is used as the classifier. The purpose behind using CNN as a classifier is because it requires less amount of preprocessing when compared to other image classifiers. It was also observed from research that the rate of error was considerably low when CNN was used. CNN has multiple layers that are hidden between the input layer and the output layer. Larger amounts of data are typically required in order to avoid overfitting. As per Fig. 1, the preprocessing unit receives input in the form of face images. The features are analyzed from the images by the pre-processing unit. The process of converting raw data into a clean dataset is known as data pre-processing. After this step, the model is trained using the clean dataset. An unknown image is then inputted to predict the gender of the unknown image.

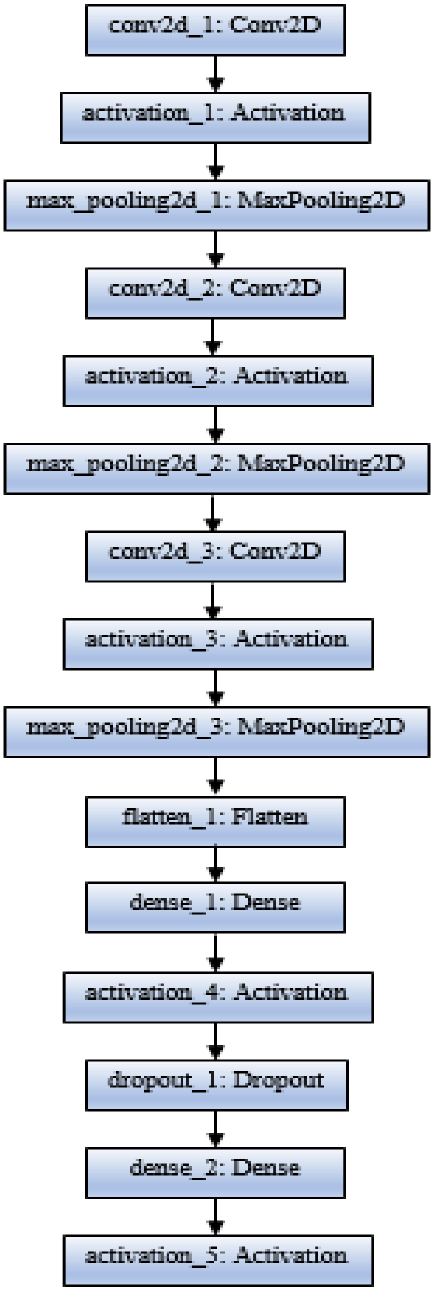


Fig-2: Model for training the dataset

As an output, we get the information regarding the gender of the unknown input face image. The model, as shown in Fig. 2, has five primary layers: The Convolutional layer, Activation layer, Max Pooling layer, Flatten layer, dense layer, and Dropout layer. The first three layers consist of the Convolutional layer, the activation layer, and the Max Pooling layer. The Rectified Linear Unit (ReLu) is used as the activation layer. The fourth layer is composed of the Flattening layer, the dense layer and again the Activation layer. The output given by the first three primary layers is in 3D and needs to be converted to 1D. The Flattening layer is responsible for the conversion of the 3D output to 1D. The Dense layer is also known as the fully connected layer. The purpose of this layer is to convert the matrix into a list and connect all the nodes with each other. The fifth layer consists of the Dropout layer, the dense layer, and the Activation layer. The duplicate images in the dataset are removed by the Dropout layer. This is done to avoid overfitting. The sigmoid classification used by the activation layer is considered best for binary classification problems. In order to optimize the network, an extra training step is added after all of the network parameters converge.

IV. APPROACH

The first step in the implementation of the project will be to create a dataset. Datasets are collections of relevant data that are used to train the system to get the required output. Datasets can be either created through data collection or by using a publicly available dataset available on the internet. An example of a publicly available dataset is the UTKFace dataset, which consists of the images of people of different ages, genders, and ethnicities. The dataset images are used as the input to the gender identification algorithm used in the project. The input facial images are analyzed based on the algorithm and the image features are identified. Then, an unknown facial image is provided to the algorithm for the purpose of its gender identification. The output generated will contain the predicted gender of the unknown image. The UTKFace dataset contains images divided into two categories, ‘Train’ and ‘Validation’, each of which contains male and female facial images. The algorithm is then trained using 8000 images of each class and validated using 1000 images of each class. The model is trained using a ConvNet (Convolutional Neural Network) consisting of 5 layers. The CNN used consists of a number of hidden layers such as the convolutional layer, the ReLu layer, the max-pooling layer, the fully connected layer, etc. Using these layers the input facial image is converted into weights and saved in the ‘.h5’ format. These weights are then used for predicting the gender of the person in an unknown image. The average accuracy achieved in the project is 90%.

The training program also includes data augmentation, which is the increasing of the number of images in the dataset. This is done because having a lot of high-quality information in the dataset is a key aspect of achieving a higher prediction accuracy in a machine learning model. This is performed by augmenting the training images via a variety of random transformations so that no two inputted images are exactly similar to each other. This helps in the prevention of overfitting and hence helps in the generalization of the model.

In the project, Keras is used to work on Tensorflow. Keras is an open-source neural network library. It is user-friendly and provides several features such as activation functions, layers, optimizers, etc. and it supports CNN as well as RNN. By using the appropriate class in Keras, deep learning models can be created on iOS and Android through the JVM (Java Virtual Machine). Keras enables the model to perform random transformations and normalization operations batches of image data by working on different attributes such as height shift, width shift, rotation range, rescale, range of shear, range of zoom, horizontal rip and fill mode. Using these attributes the system can automatically rotate, translate, rescale, and zoom into or out of images, as well as apply shearing transformations, rip images horizontally, fill in newly created pixels, etc.

For the purpose of image classification, ConvNet is used. Even though data augmentation is a way of reducing overfitting, it is not enough since the augmented samples are still highly correlated. The main focus while reducing overfitting must be the model’s entropic capability, i.e. the abundant information that it is allowed to store. A model that can store a lot of information generally turns out to be more accurate, as it is able to extract more features as compared to a model that can store only a few features. On the other hand, a model that stores a lot of information at some will start storing irrelevant features from the input data at some point, whereas a model that stores comparatively fewer amounts of information will have to focus on the most significant features found in the data. This may lead to more inaccuracies and it is observed that the model which stores fewer features is more likely to be truly relevant and easier to generalize.

The setup for the project is as follows:

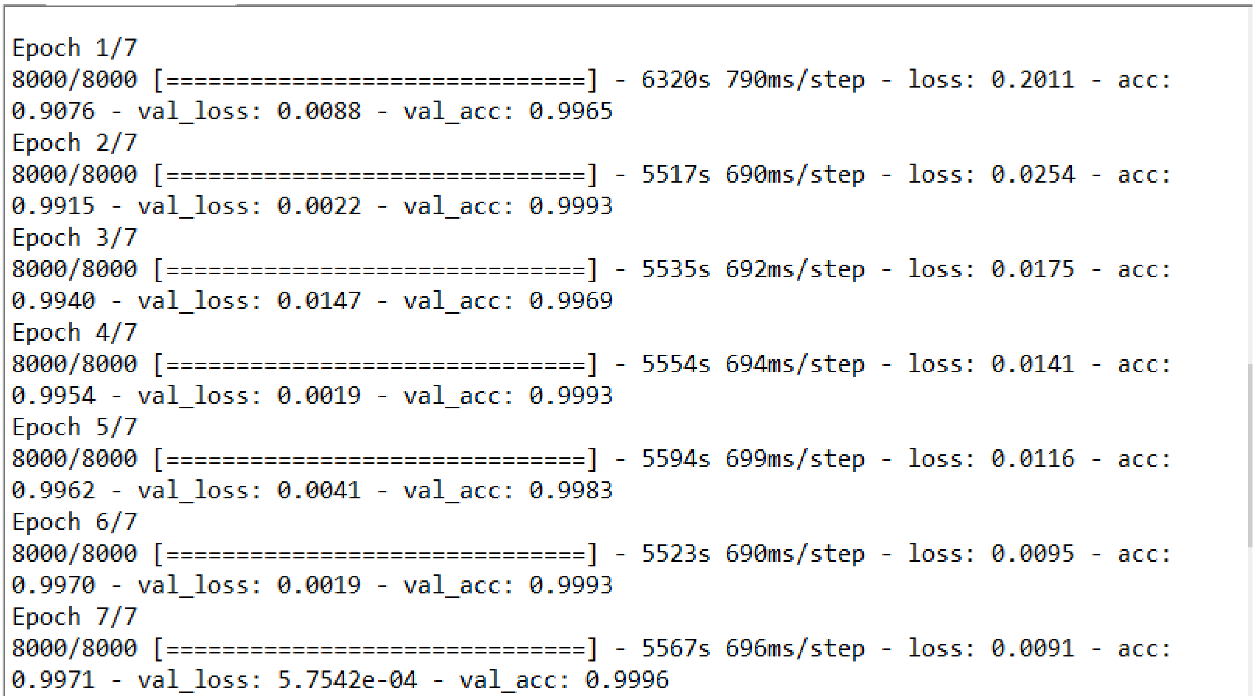
- 16000 training examples (8000 per class)
- 2000 validation examples (1000 per class)

Training Dataset: The training dataset is used as a set of examples used for training the model, i.e. to fit the different parameters. A small training dataset with less variety in its content will lead to overfitting.

Validation Dataset: A validation dataset is used to fit the hyper-parameters of the classifier. A validation dataset is necessary because it helps in the reduction of overfitting. The validation dataset is independent of the training dataset. This is so because the ultimate goal is to choose a network performing the best on unseen data.

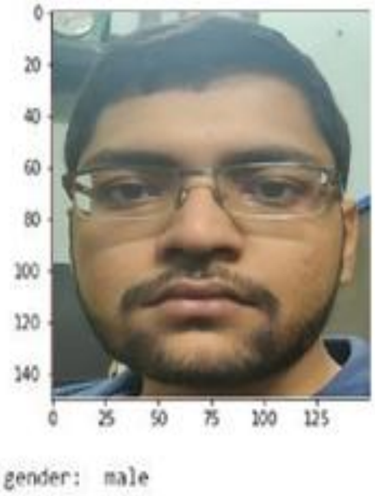
Test Dataset: The test dataset is used to test the performance of the classifier or model and to check the performance of characteristics such as accuracy, loss, sensitivity, etc. It is independent of the training and validation dataset. If a model fits both the training as well as the test dataset it achieves minimum overfitting.

The class, flowfromdirectory( ) is used here to generate groups of image data and their labels directly from jpg files in the respective folders, which are then used to train the model. Epoch is a term used to indicate the number of passes through the entire training dataset the algorithm has completed. When the entire dataset passes both forward and backward through the neural network once, it is said to complete one epoch. The approach used in this project gives a validation accuracy of 0.9965 to 0.9996 after 7 epochs.



```
Epoch 1/7
8000/8000 [==============================] - 6320s 790ms/step - loss: 0.2011 - acc:
0.9076 - val_loss: 0.0088 - val_acc: 0.9965
Epoch 2/7
8000/8000 [==============================] - 5517s 690ms/step - loss: 0.0254 - acc:
0.9915 - val_loss: 0.0022 - val_acc: 0.9993
Epoch 3/7
8000/8000 [==============================] - 5535s 692ms/step - loss: 0.0175 - acc:
0.9940 - val_loss: 0.0147 - val_acc: 0.9969
Epoch 4/7
8000/8000 [==============================] - 5554s 694ms/step - loss: 0.0141 - acc:
0.9954 - val_loss: 0.0019 - val_acc: 0.9993
Epoch 5/7
8000/8000 [==============================] - 5594s 699ms/step - loss: 0.0116 - acc:
0.9962 - val_loss: 0.0041 - val_acc: 0.9983
Epoch 6/7
8000/8000 [==============================] - 5523s 690ms/step - loss: 0.0095 - acc:
0.9970 - val_loss: 0.0019 - val_acc: 0.9993
Epoch 7/7
8000/8000 [==============================] - 5567s 696ms/step - loss: 0.0091 - acc:
0.9971 - val_loss: 5.7542e-04 - val_acc: 0.9996
```

Fig-3: Accuracy and Loss in Each Epoch



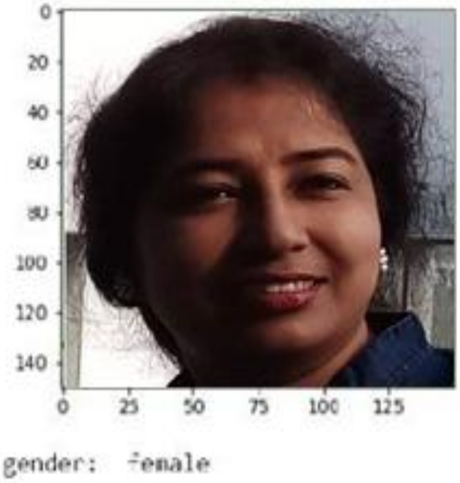


Fig-4: Output Prediction

V. CONCLUSION

Convolutional Neural Network is a supervised machine learning algorithm that gives a precise and better output when compared to other algorithms. Labeled images are used to train a model that can then determine the gender from an image. For gender classification, there are two main classes i.e., male and female. The average accuracy of 90% was achieved after training the algorithm with ten epochs. Due to the fact that fewer validation samples were used, the variation of accuracy was observed to be linear. The higher the number of samples, the higher will be the accuracy that can be achieved.

VI. FUTURE WORKS

On the basis of our analysis, it is clear that this domain has a lot of applications in real life. The ability of Convolutional Neural Networks to classify subjects at a much higher rate of accuracy can be utilized in various situations just by changing the dataset and fine-tuning the algorithm for the desired results. A lot of research work has been done in this field and a lot of different approaches have been taken into account. A fusion of two or more such approaches might yield better results and might turn out to be more efficient. Thus, our next approach would be finding out such methods and combine them to observe how they perform under the influence of various data sets.

ACKNOWLEDGMENT

Foremost, we would like to express our sincere gratitude to our guide, Mrs. Tripti Majumdar, Assistant Professor, Bengal Institute of Technology, for her valuable guidance and continuous support during the project; her patience, motivation, enthusiasm, and immense knowledge. Her direction and mentoring helped us a lot to work successfully on the project.

REFERENCES

[1] E. Makinen, and R. Raisamo, Evaluation of Gender Classification Methods with Automatically Detected and Aligned Faces," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 30, no. 3, pp. 541547, 2008.

[2] S. U. Rehman, S. Tu, Y. Huang, and Z. Yang, Face recognition: A Novel Un-supervised Convolutional Neural Network Method, IEEE International Conference of Online Analysis and Computing Science (ICOACS), 2016.

[3] N. Srinivas, H. Atwal, D. C. Rose, G. Mahalingam, K. Ricanek, and D. S. Bolme, Age, Gender, and Fine-Grained Ethnicity Prediction Using Convolutional Neural Networks for the East Asian Face Dataset, 12th IEEE International Conference on Automatic Face and Gesture Recognition (FG 2017), 2017.

[4] N. Jain, S. Kumar, A. Kumar, P. Shamsolmoali, and M. Zareapoor, Hybrid Deep Neural Networks for Face Emotion recognition, Pattern Recognition Letters, 2018.

[5] G. Levi, and T. Hassner," Age and Gender Classification Using Convolutional Neural Networks," IEEE Workshop on Analysis and Modeling of Faces and Gestures (AMFG), IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), Boston, 2015.

[6] S. Turabzadeh, H. Meng, R. M. Swash, M. Pleva, and J. Juhar, Realtime Emotional State Detection From Facial Expression On Embedded Devices, Seventh International Conference on Innovative Computing Technology (INTECH), 2017.

[7] A. Dehghan, E. G. Ortiz, G. Shu, and S. Z. Masood, Dager: Deep Age, Gender and Emotion Recognition Using Convolutional Neural Network, arXiv preprint arXiv: 1702.04280, 2017.

[8] A. Krizhevsky, I. Sutskever, and G. E. Hinton, ImageNet classification with deep convolutional neural networks, Communications of the ACM, vol. 60, no. 6, pp. 8490, 2017.