

Gender predictions using Convolution Neural Networks



Kurshid Madina, Saksham Mansotra

Abstract: Nowadays Deep learning was advanced so much in our daily life. From 2014, there is massive growth in this technology as there is a vast amount of data present. We are even getting better results from whatever we may do. In my work, I have used Convolution Neural Networks as my project depends on image classification. So what I'm trying to do is I'm using two classes in which one class is male and one class is female. I'm classifying both the classes and trying to predict who is male and who is female. For this, I have been using layers like Sequential, Convolution2D, Max-pooling, Flattening, and finally Dense. So, I connect all of these layers. I have been using two more extra layers which are Convolution2D and max-pooling connected as one layer for better classifications. In my model, I'm using Adam optimizer as I'm having only two classes and in my experiments, I found Adam as a good optimizer and I use binary cross entropy as my loss function as I'm using only two classes if we have more than two classes we can use categorical loss function and the images which I use for predictions will be converted into 64*64 matrix form. In the end, I will be getting predictions as 1 for male and 0 for female.

Keywords: Computer Vision, Gender Classification, Human-computer interaction, Convolution Neural Network (CNN).

I. INTRODUCTION

HCI- Human-computer interaction is the new name, where the user can access and control the digital devices on the bases of the human body like an eye, head, and hand gestures which makes the physical world and digital world come nearer. The aim of this is to make computers recognize human interactions and develop user-friendly Human-computer interfaces (HCI). Some steps related to this are making a computer understand speech, facial expressions, and human genders. Therefore, the idea of gender recognition research is to form a system that can detect and understand specific human genders automatically and employ them to convey information or to control devices[1]. Computer vision is a field of study that works on how computers can gain a high-level understanding of digital images and videos[2]. Normally we use images and videos to train the computer about a specific matter. The aim of this is to make a computer understand specific data which we have provided. Deep learning is based on neural networks in which they are connected as layers[3]. A basic neural network contains an input layer, a hidden layer, and an output layer connected to each other. In the same way,

I have used the CNN model in which the 5 layers are connected to each other and the training is done. Male and female images have been split into 80:20 ratio in which I have provided 10,000 images so 8000 images go for training and 2000 images for testing.

Normally we can train more than 80% but I have chosen a specific value of 80% so I can get a better overfitting model. If we do more training the model may get more overfitted and the model doesn't work better. A Gender classification system uses the face of a person from a given image to tell the gender of the given person i.e either male/female. A successful gender classification approach can boost the performance of many other applications including face recognition and smart human-computer interface. The two classes which I used are male and female in this project. The dataset is been collected from Kaggle where I have a certain amount of images and I have added necessary images based on the requirement of the size of images. So, after arranging the necessary images we start the training process and for the training process, I have used a CNN model where I connect each layer which is Sequential, Convolution2D, Max-pooling, Flattening, and finally Dense layer. After a certain amount of time, I got a good amount of accuracy. The optimizer I used is Adam as I'm having only two classes. Based on my research and trying experiments on various optimizers I found Adam as a good optimizer and I'm using categorical data as I have only two classes and if I have more than two classes I have to use other data[4].

II. METHODOLOGY

In this section, we explain our hierarchical architecture that helps the CNN models help to predict genders as efficiently as possible. Here we are providing a brief description of the Architecture of the gender predictions, the techniques which we applied to the image before predictions, the CNN architecture used to train the dataset, and finally how the predictions are done.

A. Architecture

Fig 1 depicts the architecture of the classifier which is a deep CNN to classify the detected image and then with the prediction of an image is developed. This system depends totally on human-computer interaction, where the computer tries to interpret what the user is trying to show with the knowledge it has gained from the dataset provided. The process simply starts with a user trying to predict an image as an input. Once the pre-processing of a region of interest is done the image in the form of NumPy array is passed to the CNN model for prediction. The output of the prediction is a NumPy array where the array consists of only one element that is the one having the highest weight. The output is stored and used as a digit or an operator for calculation.

Revised Manuscript received on August 01, 2020.

Revised Manuscript received on August 05, 2020.

Manuscript published on September 30, 2020.

* Correspondence Author

KurshidMadina, B.Tech in Information Technology from GITAM Institute of Technology, Visakhapatnam.

Saksham Mansotra, B.Tech in Information Technology from GITAM Institute of Technology, Visakhapatnam.

© The Authors. Published by Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP). This is an [open access](https://creativecommons.org/licenses/by-nc-nd/4.0/) article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/4.0/>)

B. Input Image Preprocessing

A person needs to make sure before starting the process that in order to predict accurately the image with the size of 64*64 needs to be placed as the model is trained on a dataset that has only 64*64 images.

When the image is sent for prediction it will predict according to the image given and results in the output. Every feature from the image will be extracted and will give the prediction. There are only 2 preprocessing techniques that are being applied on a frame that is resized and reshaped. The Region of interest is of size 300x300 but our CNN model takes only 100x100 images as input. So using resize function we transform the image to a 100x100 size. Then reshape is done to the resized NumPy array which makes sure the data is not changed.

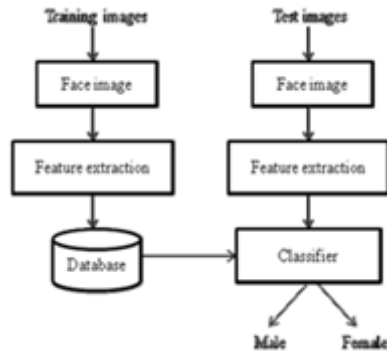


Fig 1. Gender classification system

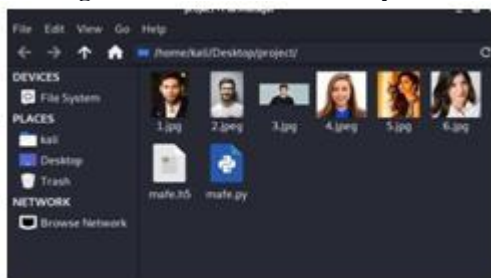


Fig.2 user showing images sent for predictions

C. CNN Architecture

Deep learning is entirely based on the connection of layers and I have done a lot of research in each and every layer and I will be mentioning each and every layer very clearly. CNN contains a total of 4 parts they are Convolution layer, the pooling layer, the flattening layer, and the fully connected layer also called the neural network layer. So coming to the first layer we have the Convolution layer[6]. A Convolution is basically a combined integration of two functions and it shows you how one function modifies the other or modifies the shape of the others. It is a quite popular function now.

On the left, we are going to look at the input image that's how we are going to look at images just 0's and 1's to simplify things. We can see a smiley face there then we got a feature detector. So, feature detectors are 3*3 matrix. There can be different feature detectors with 7*7 and 5*5 etc. Usually, we all see the 3*3 matrix only the reason is to make them 3*3 and so we are going to stick to the conventional way. A convolution operation is signified by an X in the circle. Well, we take this feature detector and put it on the image so it will cover the top left corner nine pixels and you basically multiply each a value by each value so respect to value so the top 0 by the top-left value and then the left value basically is in the position of a 1 1 by position about 1 1 position number or 0 1 or 02 and so on. So, it's element-wise multiplication on these matrices and then you

add up the result. So, in this case, nothing matches up so it's always either 0 by 0 or 0 by 1 so the result is 0. So, in the second one if you can see one of them matched up in the left so we will have a 1 and next 0 and so on. The image on the right is called a feature map also called a void feature. So when u apply a convolution operation operator to something it doesn't become convoluted it becomes convolved and it can also be called as activation map.

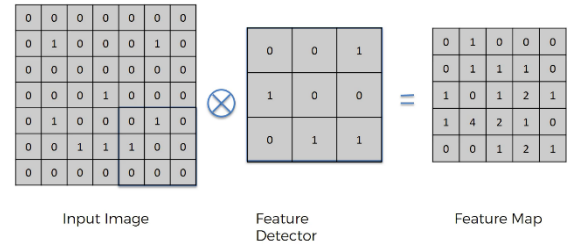


Fig 3. Feature extraction

This is an input image and we create multiple feature maps because we use different filters so that we preserve lots of information so we will be having multiple feature maps and for each feature map we can apply different filters it can be depended upon the feature map. So, this will help us obtain our first layer.

So after the convolution layer is obtained we built the ReLU layer[7]. We apply the ReLU layer on the top of the Convolution layer. It is known as a Rectified function. The reason why we apply rectified function is we want to increase non-linearity. The images would be having lots of non-linear elements and the transition between pixels is often non-linear because there are borders, different colors and there are different elements in the images and at the same time when we are applying a mathematical operation such as convolution and running this feature detection to create our feature maps it can go wrong as we might create something linear and therefore we need to break up the linearity.

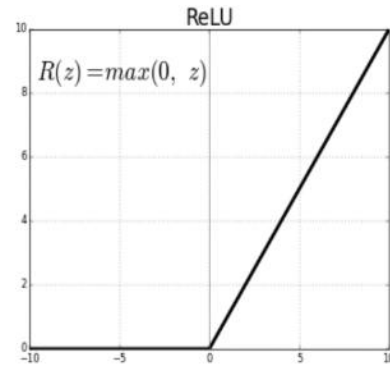


fig 4. Relu Activation function

So, the next layer will be max-pooling. Some images are perfect in position, some can be rotated and some can be squashed. We want the neural network to recognize every single image in whatever position they may be. We will not have every image at the same it will be different from each other and we want the neural network to recognize all of these images though they are different. It needs distinctive features to recognize the images and predict it though they are in different positions, directions, etc.

So, we have to make sure that our neural network has spatial invariance means that it doesn't care if the features are a bit tilted or a bit different texture or a bit closer or a bit further apart relative to each other. So, if the feature itself is a bit distorted our neural network has to have some level of flexibility to be able to find that feature and that is what pooling is all about.

So how pooling works is we take a box of 2*2 pixels, it doesn't have to be 2*2 it can be any size of the box. So, what happens here is we have to place it in the top left corner and it will select the maximum value in that box and then record only that value. Then we move the box towards the right and select the maximum value and so on. So, by pooling these features we are first getting rid of 75 percent of the information that is not a feature which is not important that we're looking for because we're really three pixels out of four so we are getting 25 percent and then also because we are taking the maximum of the pixels or the values that I have. We are therefore accounting for any distorting. By disregarding the unnecessary non-important formation we're helping with preventing overfitting.

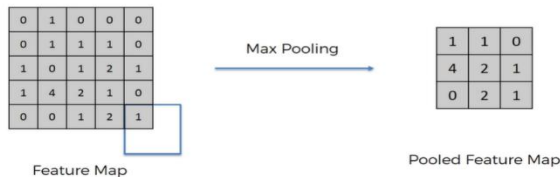


Fig 3. Feature extraction

This is our input image and then we apply convolution operation and we got the conclusion and now to each of those feature maps we've applied the pooling layer. So, we've got these two layers convolution and pooling. So, by this pooling what actually happens is suppose if an image is rotated the exact distinctive feature also gets rotated so to catch that distinctive feature we use max-pooling[8].

Next comes the Flattening, it is a very simple step. After the convolution operation and max-pooling, we will get a pooled feature map. We are going to take it and we're going to flatten it into a column. So basically just take the numbers row by row and put them into one long column and the reason for this is because we want to later input this into an artificial neural network for further processing.

To sum all these up we've got an input image, we apply a convolution layer and ReLU or rectified linear units function after the convolution and then we apply to pool and then we flatten everything into a long vector which will be our input layer for an artificial neural network and that's how flattening is done[9].

Next comes the full connection layer. The full connection layer contains an Artificial neural network. After the convolution, pooling, flattening we have to add the full connection which contains the Artificial neural network and this helps us in better predictions. The Artificial neural network normally has an input value and hidden layers in between and an output layer and all the layers will be fully connected then it will try for predictions like for example either its a dog or cat or does he have a tumor or not. So, what I'm doing here is I'm classifying the images using the Artificial neural network and trying to predict the perfect value[10]. In my experiment, I'm trying to classify male or female.

There will be an Artificial neural network that contains 5 inputs which are connected to 6 neurons which is a fully connected layer and then it is connected to 8 neurons and

then we have two outputs one for male and one for female. We normally have only one class for predicting a numerical type of problem or regression but when we are using classification you need an output for a class except for the exception is when you have only two classes. Normally for one class or two classes, we can use binary cross-entropy but when we have more than two classes we have to use categorical. The image gets processed and then convolved then pooled, flattened, and then through the artificial neural network, and then a prediction is made.

D. Prediction

To perform predictions, the user will be asked to insert some images into the list, the user has to insert some images. For choosing an image the user is supposed to choose male and female images with facial features. The list with images added will go for predictions and the predicted images will be printed in output as 0 if the image is female and 1 if the image is male.

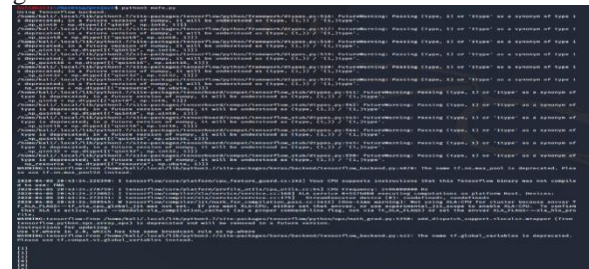


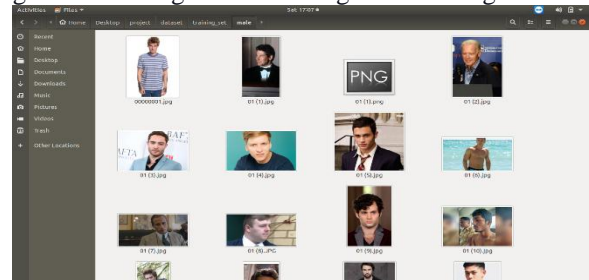
Fig 6. Showing the predicted values for images.

III. RESULTS AND DISCUSSION

In this section we'll be discussing the dataset we used to train the model, what results in we obtained after training which includes validation accuracy and loss and some experimental results.

A.Dataset

The concept of Convolution Neural Networks and Artificial Neural Networks were there from the 1980s but were not very much useful because of the lack of data. In earlier days registers were maintained to keep different all sorts of records and information. Very less digital data was available which is the key aspect for CNN to work. In this early 2000s data started to take digital form, everything started to get automated and computerized. The dataset that we have used to train our model is of male and female images. The 'kaggle' Gender Classification dataset of facial images have been used to evaluate the framework. The Gender classification dataset database has 2 classes of facial images. There are in total of 8000 training images and 2000 test images. Every class of facial images has around 4000 images for Training and 1000 images for Testing.



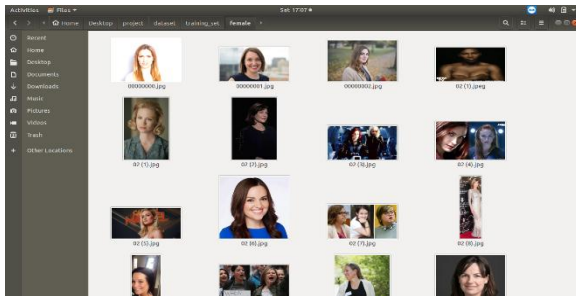


Fig 7. Sample 'gender predictions' images dataset

B.Experimental Results

During the process, the user will be asked to insert some images into the folder, as fig 8 (a) shows the user has to insert some images. For choosing an image the user is supposed to choose male and female images with facial features.

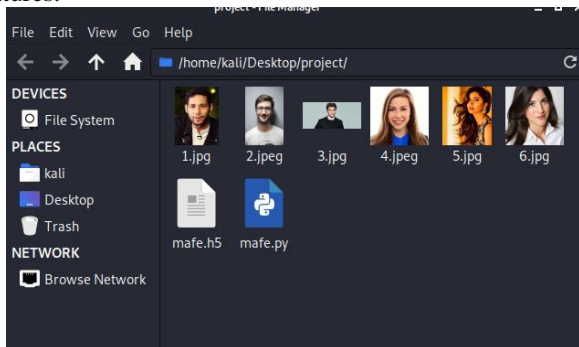
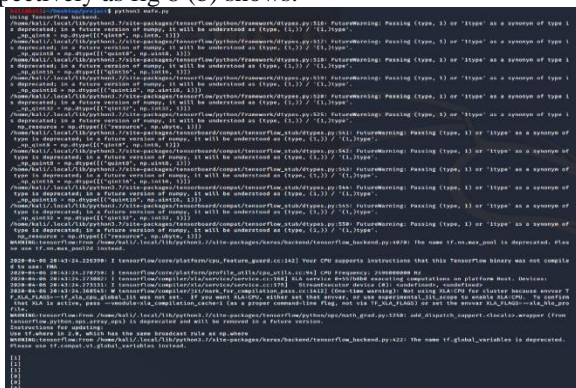


Fig 8 (a) Images for prediction.

In fig 8 (a) the user has inserted some of the images into the folder and it goes for prediction. If there is a female in the list it will be printed as 0 and males will be printed as 1 respectively as fig 8 (b) shows.



8 (b) Predicted values of the images

Table-1: validation accuracy and loss of different epochs.

	Validation Loss	Validation Accuracy
1	0.3216	0.7645
2	0.5832	0.7764
3	0.6821	0.7648
4	0.2629	0.7564
5	0.1057	0.7847
6	0.5519	0.759
7	0.2809	0.7651
8	0.8712	0.7601
9	0.6861	0.7906
10	0.9969	0.7953

IV. CONCLUSION:

This paper proposes an approach to use Convolution Neural Networks for classification of gender and use the predicted value to show which gender we belong to i.e male or female. Using the power of Convolution Neural Networks for classification we have applied to pre-process techniques on each image such as resize and reshape to fit each image to our Neural network. The training and testing dataset consist of 2 classes each with 8000 and 2000 images respectively. The validation accuracy obtained was 79%.The main thing everyone has to focus is on fitting. The model shouldn't go on overfitting and underfitting. We have to avoid false positives as much as possible. The Convolution NeuralNetwork consists of 2 Convolution layers attached with a pooling layer each, 1 flattening layer, 1 dense layer, and an output layer. Based on the predicted output of images the value will be 0 for females and 1 for males.

REFERENCES:

1. M. K. Bhuyan., et al. —A novel set of features for continuous hand gesture recognition, Journal on Multimodal User Interfaces., vol. 8, no. 4, pp. 333–343, 2014.
2. Kasthurirangan Gopalakrishnan, et al. – Deep Convolution Neural Networks with transfer learning for computer vision-based data-driven pavement distress detection, Journal on Elsevier, vol 157, pp 322-330,2017.
3. Deegan J Atha, et al. – Evaluation of deep learning approaches based on convolution neural networks for corrosion detection, Journal on sage, 2017.
4. Gil Levi, Taj Hassner. – Age and Gender Classification using Convolution Neural Networks. preceedings of IEEE Conference on Computer vision and pattern recognition (CVPR) Workshops, pp.34-42 ,2015.
5. Yann LeCun, et al. – Gradient-Based Learning Applied to Document Recognition. Proc. Of the IEEE, 1998.
6. Jianxin Wu – Introduction to Convolution Neural Networks, 2017.
7. C.-C. Jay Kuo – Understanding Convolution Neural Networks with a Mathematical Model, arXiv, 2016.
8. Dominik Scherer et al. – Evaluation of pooling Operations in Convolution Architectures for Object Recognition, ICANN, pp 92-101,2010.
9. JonghoonJin, et al. – Flattened Convolution Neural Networks for FeedForward Acceleration, arXiv, 2015.
10. Zixian Zeng, et al. – CNN Model Design of Gesture Recognition Based on Tensorflow Framework, IEEE, 2019.

AUTHORS PROFILE



Kurshid Madinah has completed his B.Tech in Information Technology from GITAM Institute of Technology, Visakhapatnam. His area of interest is towards Computer Vision.



Saksham Mansotra has completed his B.Tech in Information Technology from GITAM Institute of Technology, Visakhapatnam. His area of interest is towards Computer Vision.

