Detection and Classification of Human Gender into Binary (Male and Female) Using Convolutional Neural Network (CNN) Model

Abstract

This paper focuses on detecting the human gender using Convolutional Neural Network (CNN). Using CNN, a deep learning technique used as a feature extractor that takes input photos and gives values to various characteristics of the image and differentiates between them, the goal is to create and develop a real-time gender detection model.The model focuses on classifying human gender only into two different categories; male and female. The major reason why this work was carried out is to solve the problem of imposture. A CNN model was developed to extract facial features such as eyebrows, cheek bone, lip, nose shape and expressions to classify them into male and female gender, and also use demographic classification analysis to study and detect the facial expression. We implemented both machine learning algorithms and image processing techniques, and the Kaggle dataset showed encouraging results.

Keywords: Gender Detection, Convolutional Neural Network, Deep Learning, Artificial Intelligence, Gender

1.0 Introduction

Gender is one's identity in our society, a significant element used to identify a person in our society with reference to their biological difference. There are numerous applications for artificial intelligence gender recognition, including smart human-machine interface development, health, cosmetics, electronic commerce, and more. Convolutional Neural Network (CNN), a Deep Learning Algorithm, was employed in this study to categorize the gender of humans. CNN receives input images, applies a value to various visual components, and distinguishes between them. Facial recognition technology, which includes human gender detection, has drawn a lot of interest. This can be accomplished using a variety of static physical traits, such as the shape of the hand, body, fingernails, face, and eyebrows.

Facial recognition is one of the trendiest topics in facial technology right now. Many studies have been conducted to estimate age, gender, and detect emotions using Deep Learning techniques like Artificial Neural Networks (ANN) and CNN. suggested a CNN-based age identification system that achieves a 95% accuracy rate. We have demonstrated the CNN-based human gender classification in this work.

The major reason why this work was carried out is to solve the problem of imposture. Imposture is an instance of pretending to be someone else in order to deceive others.

The challenge of gender recognition from facial photographs is a research topic that is still being worked on. The criteria and performance are still insufficient, despite the fact that the researchers offered several solutions to the issue.

The aim of this work is to model a real-time gender detection system using CNN with the following objectives:

1. Design a convolutional neural network system to extract facial features of human.
2. Design a machine learning system that classifies human facial features into male and female gender.
3. To design a CNN using demographic classification analysis to study and detect facial expression and gender.

2.0 Review of Related Literature

Using facial image data, M. S. Fathollahi et al. [2] developed an automatic gender classification system utilizing two simultaneous CNN models. While VCNN uses vanilla CNN layers, CDCN, one of the two networks, uses the central difference convolution layer. Next, a dense layer for classification is created by concatenating the output of the two networks. The Casia WebFace dataset served as the system's training set. Two datasets were used for testing: the FEI dataset and the LFW (labeled faces in the wild) dataset. For the LFW dataset, the authors' technique yields an accuracy rate of 97.79%, while for the FEI dataset, it achieves a rate of 99.10%. For real-time emission and gender detection, A. Lahariya et al. [3] presented an ensemble CNN model by averaging the output of Mini-Xception [4] with a straightforward 4-layer vanilla CNN model. The suggested approach effectively achieves 95% accuracy for binary gender classification on the IMDB dataset and 68% accuracy for emotional classification into 7 categories (disgust, furious, fear, joyful, sad, surprise, and neutral) on the FER-2013 dataset.

Human gender estimate has been extensively studied in the fields of computer vision and machine learning. Using local binary pattern (LBP) and support vector machine (SVM) inside a polynomial kernel on the CAS-PEAL face database, Lian et al. [5] achieved an accuracy of 94.81%. This method states that if the block size for the LSP operator is chosen correctly—a truly hard task—good accuracy can be attained.

[6] classified people according to their gender using solely their facial traits (forehead, lips, nose, eyes, and brows). A limitation of this study is that complicated backgrounds have an impact on the feature extraction technique they employed.

[7] classified men and women using geometric-based futures in which each ethnic dataset contained 126 frontal photos and included augmented reality. In this instance, their accuracy was 80.3% and 86.6%, respectively.

[8] created a real-time Public Facial Recognition System (FRS) for Nigerian criminal investigation using HAAR cascades classifier technique, a prototype system to detect criminals in real time. The system achieved an 80% accuracy rate of collected photos stored in the database. This shows HAAR Cascade Classifier Technique may not be the best option for Gender Classification.

2010 saw the application of a texture-based local binary pattern for feature extraction and the naïve s Byes, ANN, and linear SVM classification algorithms. With just 100 facial photos taken from the Nottingham scan database, they obtained an impressive 63% accuracy rate.

[9] have experimented with black propagation, SVM, and LSP to classify gender from face images. They have used the 400-image ORL dataset and the 100-image Nottingham scan dataset in this study. Following deployment, they achieved 100% accuracy for the ORL dataset and 71% accuracy for the Nottingham scan database.

It is evident from the literature assessment as a whole that gender classification has to be improved. The primary drawback of the aforementioned gender categorization research projects is the separation of feature extraction and classification processes.

Here, prior knowledge is required to obtain an ideal pre-processing and feature extraction design. CNN is an example of a multi-layer neural network model that can optimize fitters by autonomous learning, independent of past information, showing that CNN can reach higher accuracy.

A superior proposal was made by [10]. They suggested using CNN with nine layers for real-time emotion recognition, which can classify seven different types of emotions with an accuracy of almost ninety.

utilizing the FER emotion recognition dataset, the emotion and gender detection system utilizing CNN proposed by [11] achieved 66% for emotion detection and 95% for gender detection.

[12] presented a computerized emotion identification system based on the LSP classifier. The system performed 92.22% on the JAFFE dataset and 94.39% on the CKT dataset.

[13] presented a method for age and gender identification that obtained 72.53% and 98.90%, respectively, in age and gender recognition. Using the FER-20133 dataset, Octavio [14] suggested a real-time emotion gender classification system using CNN that obtained 66% accuracy in emotion classification and 95% accuracy in gender classification when utilizing the IMDB dataset.

.

2.1 Application of Machine Learning in Facial Detection

The field of artificial intelligence known as "machine learning" is concerned with creating programs that can recognize patterns in data and forecast them [15]. To solve issues that are too complicated for traditional programming techniques to handle, machine learning algorithms learn from data.

Face recognition is a biological identification method that recognizes people based on the distinctive features of their faces. Convolutional neural networks (CNNs) with deep learning capabilities are the most widely used machine learning algorithms for facial recognition [15].

An artificial neural network architecture that works well for image categorization applications is CNN [16]. CNN gains the ability to identify features in photos and apply those features to categorize the images. For facial recognition, a CNN's depth is crucial because it enables the CNN to pick up more intricate facial traits. In identifying faces, the steps are;

1. **Face alignment and detection**: Finding faces in the fed images is the initial stage. A machine learning technique known as a Haar Cascade Classifier [8], which has been trained on both positive and negative pictures, can be used to accomplish this.
2. **Features measurement extraction**: Once faces have been aligned and detected, the next step is to extract features from them.
3. **Face recognition**: The last step is to match the extracted features with faces in a database.

3.0 Materials and Methods

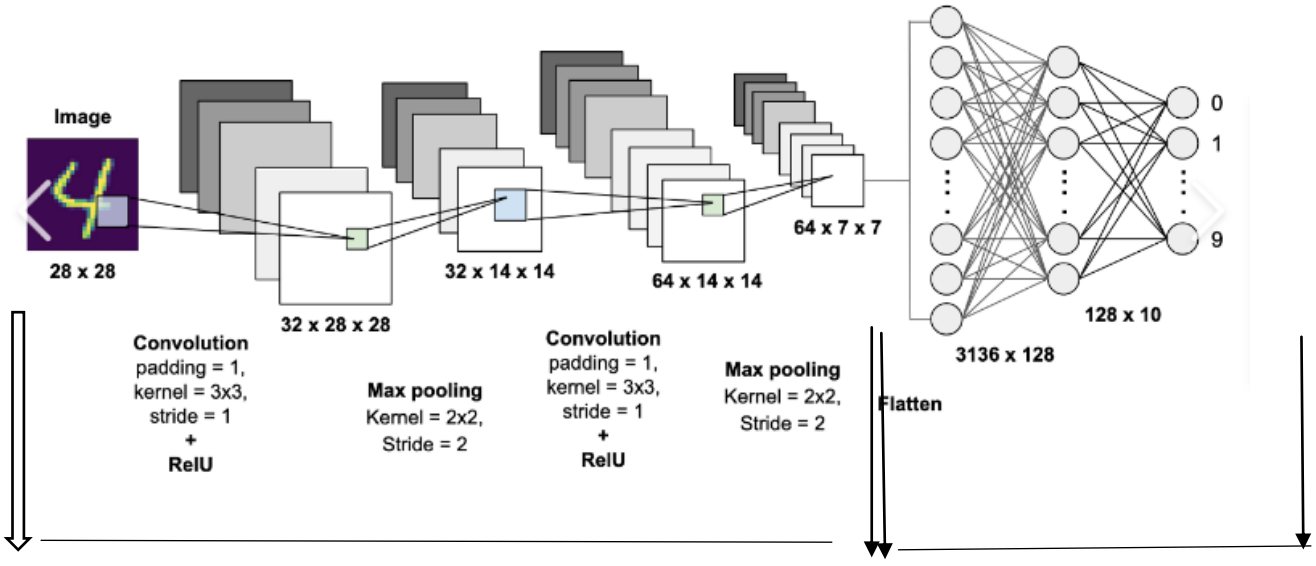
The methods used in this work are; Records view and Observation methods. Image samples used for data classification and feature extraction ware analyzed.The authors took time to observe the

end methods of which images were captured and classified and how the database saved the records.

The primary objective of this work is to recognize the gender with emotions from the human face images utilizing the set of facial features in real-time.

Facial extraction from face images is an important part of this method (as shown in figure 1), and it involves four stages. The stages include;

1. Image Pre-Processing: By removing noise and smoothing the pictures, the preprocessing procedure can enhance the appearance of the input image. Without sacrificing image specifics, it removes redundant images. To create an image with a consistent size and rotation, preprocessing also include filtering and normalizing the picture.
2. Face Detection: Feature extraction is defined as the process of extracting information from the backdrop of input photos with complicated backgrounds and varying lighting conditions in object detection [17]. Face feature extraction and segmentation from an uncontrolled background are involved.
3. Feature Extraction: It includes shapes, movement, color, the texture of the facial image. It extracts meaningful information of an image compared to the original image.
4. Feature Classification: The categorization step identifies faces in photos, categorizes them into classes, and aids in their proficient recognition. Feature categorization is a stage of selection that works with exchanges that connect them according to specific specifications and preserve important information.



Feature extraction Classification

Figure 1: Feature Extraction and Classification stages of Facial Extraction

Data Training

Data was gotten from Kaggle (https://www.kaggle.com/code/rajachanga/gender-classification-using-vgg16-cnn) , and trained. We summarize the model by using "Model.Summary()" method. The following are summary of the trained data set;

TOTAL PARAM's: 3,631,752

Trainable PARAM's: 3,621,752

Non-trainable params: 0. See more details in fig 2.

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d (Conv2D)              (None, 30, 30, 100)       1000
_________________________________________________________________
activation (Activation)      (None, 30, 30, 100)       0
_________________________________________________________________
max_pooling2d (MaxPooling2D) (None, 30, 30, 100)       0
_________________________________________________________________
conv2d_1 (Conv2D)            (None, 29, 29, 100)       40100
_________________________________________________________________
activation_1 (Activation)    (None, 29, 29, 100)       0
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 29, 29, 100)       0
_________________________________________________________________
conv2d_2 (Conv2D)            (None, 28, 28, 100)       40100
_________________________________________________________________
activation_2 (Activation)    (None, 28, 28, 100)       0
_________________________________________________________________
max_pooling2d_2 (MaxPooling2 (None, 14, 14, 100)       0
_________________________________________________________________
conv2d_3 (Conv2D)            (None, 13, 13, 400)       160400
_________________________________________________________________
activation_3 (Activation)    (None, 13, 13, 400)       0
_________________________________________________________________
max_pooling2d_3 (MaxPooling2 (None, 13, 13, 400)       0
_________________________________________________________________
flatten (Flatten)            (None, 67600)             0
_________________________________________________________________
dropout (Dropout)            (None, 67600)             0
_________________________________________________________________
dense (Dense)                (None, 50)                3380050
_________________________________________________________________
dense_1 (Dense)              (None, 2)                 102
=================================================================
Total params: 3,621,752
Trainable params: 3,621,752
Non-trainable params: 0
```

Figure 2: Details of the Trainable and Untrainable Parameters

The model has now been trained and is prepared to identify the sex of any arbitrary image found in the dataset. We present 15 test photos at random along with the expected labels and ground truth.

This work makes use of Keras, an open-source library for neural networks. By focusing on several variables including height shift, width shift, rotation range, rescale, range of shear, horizontal rip, and fill mode, it allows the model to execute random transformations and normalization operations on batches of image data. With the help of these characteristics, the system may automatically classify photos by rotating, translating, rescaling, zooming in or out of them, applying shearing transformation, ripping images horizontally, filling in recently formed pixels, etc.

Formulations

By combining a convolutional and a subsampling layer into one layer, the suggested CNN model has fewer layers. In this study, we use strides of 2 to replace successive convolutional and subsampling layers with a single convolutional layer (l). A feature map, $Y_j(x, y)$ in layer l can be described by the following equation:

$$Y_j^{(l)}(x, y) = f\left(\sum_{i=0}^{N}\sum_{u=0}^{K_x^{(l)}}\sum_{v=0}^{K_y^{(l)}} Y_i^{(l-1)}\left(S_x^{(l)}x + u, S_y^{(l)}y + v\right) w_{ji}^{(l)}(u, v) + \theta_j^{(l)}\right), \quad (1)$$

where $Y_i^{(l-1)}$ and $Y_j^{(l)}$ are the input and output feature maps respectively, $f()$ denotes the activation function, $w_{ji}^{(l)}$ is the convolutional kernel weight, $\theta_j^{(l)}$ is the bias, N represents the total number of input feature maps,

$S_x^{(l)}$ is the horizontal convolution step size, $S_y^{(l)}$ is the vertical convolution step size, and $K_x^{(l)}$ and $K_y^{(l)}$ are the width and height of convolutional kernels, respectively. The width $W^{(l)}$ and height $H^{(l)}$ of the output feature map with convolution step sizes of $S_x^{(l)}$ and $S_y^{(l)}$ can be computed as:

$$W^{(l)} = \frac{W^{(l-1)} - K_x^{(l)}}{S_x^{(l)}} + 1 \quad (2)$$

and

$$H^{(l)} = \frac{H^{(l-1)} - K_y^{(l)}}{S_y^{(l)}} + 1, \quad (3)$$

where $W^{(l-1)}$ and $H^{(l-1)}$ correspond to the width and height of input feature map.

An illustration of the aforesaid formulation can be found in Figure 3. The figure illustrates how a single 6 × 6 convolution operation with step size (strides) of 2 can substitute a 5 × 5 convolution followed by a 2 × 2 subsampling operation since they provide an output feature map that is precisely the same.

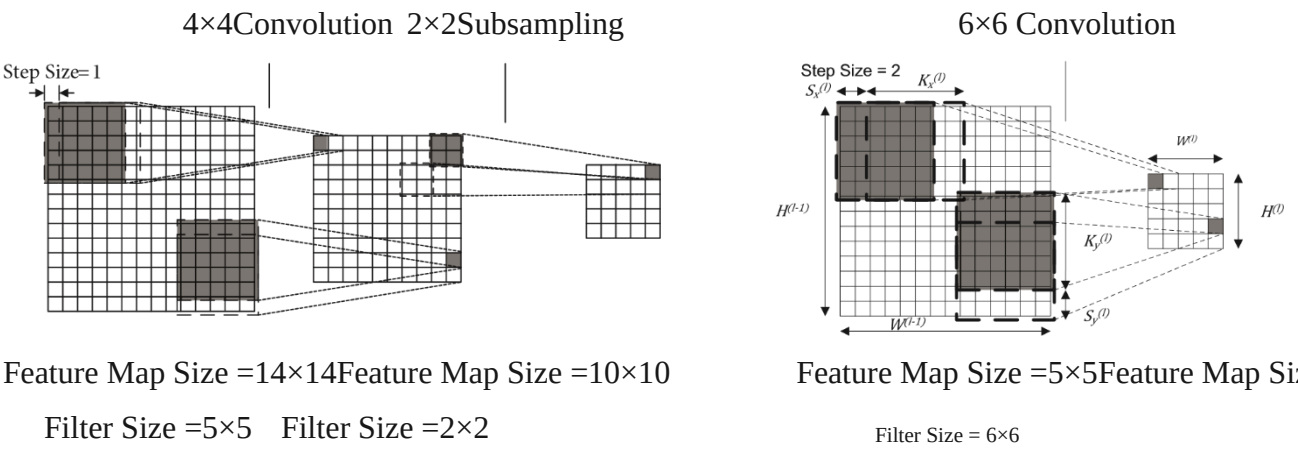


Figure 3: Feature map operations within the convolutional layer: The two approaches are as follows: (a) traditional method using convolution (with step size of 1) and subsampling; (b) fused convolution/subsampling method using convolutions (with step size of 2).

Take note that cross-correlation is used in place of convolution in the above expression. Convolution and cross-correlation are comparable operations in the processing of images, with the exception that convolution involves flipping the kernel weights upward and laterally. The following is the general formula for a 2D discrete convolution:

$$K_x \quad Ky$$

$$Y(x, y) = \sum \sum X(x - u, y - v)w(u, v) \tag{4}$$

$$u=0 \; v=0$$

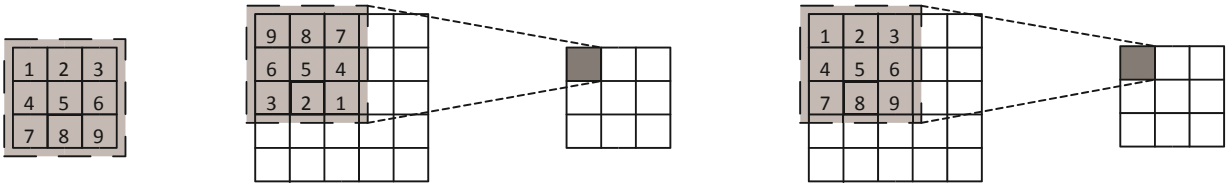
where w is the kernel, X and Y are the input and output images, and Kx and Ky stand for the convolutional kernel's width and height, respectively. Prior to doing the dot product, the kernel weights must be flipped in this process. In comparison, the following equation characterizes a 2D discrete cross-correlation (for image processing):

$$K_x \quad Ky$$

$$Y(x, y) = \sum \sum X(x + u, y + v)w(u, v). \tag{5}$$

$$u=0 \; v=0$$

The main difference between Eqs. (4) and (5) is that the kernel weights in Eq. (5) are not flipped. We use the diagrams in Figure 4 to demonstrate these processes; Figure 4a shows an example of a convolution kernel. The traditional method, which is shown in Figure 4b, uses the convolution kernel to convolve an overlapped input plane while flipping the kernel weights in both the horizontal and vertical directions. This results in a 2D discrete convolution. The same operation—minus inverting the kernel—is depicted in Figure 4c. Flipping has minimal impact on the convolution output since the convolution kernel's (weights) values are started at random in a convolutional layer.



(a) (b) (c)

Figure 4: The 2D discrete convolution operation: (a) example convolution kernel, (b) convolution with kernel weights flipped, (c) convolution with kernel weight not flipped.

In both feedforward and backward propagations, flipping operations increase the duration of computation. As a result, the change of using cross-correlation in place of convolution is beneficial, particularly when handling multiple convolutions that must be carried out on high kernel sizes and on huge datasets over a significant number of training iterations (epochs).

We use a scaled hyperbolic tangent function in the convolutional layers, which is represented by the following equation:

$$f(x) = A \tanh(Bx), \tag{6}$$

Where A denotes the amplitude of the function and B determines its slopes at the origin. The values of A and B are set at 1.7159 and 2/3 as suggested by [18].

For the output layer, a single perceptron is used to determine the class of a particular input pattern. The value at output neuron is given by the following equation:

$$Y_j^{(l)} = f\left(\sum_{i=1}^{N} Y_i^{(l-1)} w_{ji}^{(l)} + \theta_j^{(l)}\right), \tag{7}$$

where *f*() represents the activation function used for each neuron, *N* is the total number of input neuron(s), $Y_i(l^{-1})$ is the value of the input neuron, $w_{ji}^{(l)}$ is the synaptic weight connecting the input neuron(s) to the output neuron, and $\theta_j^{(l)}$ denotes the bias. Male gender is represented by the output value of +1, whereas −1 indicates that the pattern belongs to the female class.

4.0 Implementation

This project was implemented using the MxNet frameworks and the Python programming language. MxNet is an open-source deep learning framework that interacts with a variety of platforms, including mobile devices and cloud infrastructure, to enable the definition, training, and deployment of deep neural networks.

Getting Started

Kaggle's 2018 dataset named "Facial Age" was used. It's a 9,778 RGB images of faces in PNG format of size 200X200 pixels each. It covers a long age span, across 1 to 116 years. Image labels are embedded in file names, as per this nomenclature. So, we have age, gender and race, separated by underscores.

Since we were using a CNN-based deep learning approach for model development, we extracted gender markers encoded in these image file names and converted our photos to grayscale to reduce the computational cost of our model fitting. Our dataset was divided into training and test sets of 75% and 25%, respectively. Next, we trained our CNN model on this training set, and tested its performance on the test set.

Model Building

We set up our environment by importing needed libraries. These libraries include; pandas, numpy, cv2, matplotlib, tensorflow, Google Colab, keras, sklearn, etc. We then loaded our dataset, using a a "for loop" to read all images, convert from RGB to grayscale, and resize to 100x100. And then, we appended all image pixel data to this pixels numpy array and labels to gender array as shown in figure 5.

```
# Mount to Drive
from google.colab import drive
drive.mount('/content/drive')
%cd /content/drive/My Drive/Project5_AgeGenderEmotion_Detection/1.2_gender_input_output

# Load dataset
path = "./input/UTKFace"
pixels = []
age = []
gender = []
i=0
for img in os.listdir(path):
  i=i+1
  genders = img.split("_")[1]
  img = cv2.imread(str(path)+"/"+str(img))
  img = cv2.cvtColor(img,cv2.COLOR_BGR2GRAY)
  img=cv2.resize(img,(100,100))
  pixels.append(np.array(img))
  gender.append(np.array(genders))
pixels = np.array(pixels)
gender = np.array(gender,np.uint64)

# Split Dataset into 75:25 training & test
x_train,x_test,y_train,y_test = train_test_split(pixels,gender,random_state=100)
```

Figure 5: Code showing loading of dataset

The CNN architecture is then defined. In summary, we have:

• An input 2D MaxPooling layer combined with an input 2D Convolutional layer (including 32 filters).

• Three sets of two-dimensional convolutional layers (each with 64, 128, and 256 filters) combined with two-dimensional MaxPooling layers once more.

• One 128-node dense layer.

• An output dense layer consisting of two nodes, representing our labels for male and female.

The CNN architecture that was previously defined is assembled next. For model fitting, we employed model checkpoints to save our model as it keeps getting better over 30 epochs. Figure 7 illustrates how the model's performance was evaluated by plotting the lineplots for accuracy and loss using the code block in Figure 6.

```
# Checking the train and test loss and accuracy values from the neural network above.
train_loss = save.history['loss']
test_loss = save.history['val_loss']
train_accuracy = save.history['accuracy']
test_accuracy = save.history['val_accuracy']

# Plotting a line chart to visualize the loss and accuracy values by epochs.
fig, ax = plt.subplots(ncols=2, figsize=(15,7))
ax = ax.ravel()
ax[0].plot(train_loss, label='Train Loss', color='royalblue', marker='o', markersize=5)
ax[0].plot(test_loss, label='Test Loss', color = 'orangered', marker='o', markersize=5)
ax[0].set_xlabel('Epochs', fontsize=14)
ax[0].set_ylabel('Categorical Crossentropy', fontsize=14)
ax[0].legend(fontsize=14)
ax[0].tick_params(axis='both', labelsize=12)
ax[1].plot(train_accuracy, label='Train Accuracy', color='royalblue', marker='o', markersize=5)
ax[1].plot(test_accuracy, label='Test Accuracy', color='orangered', marker='o', markersize=5)
ax[1].set_xlabel('Epochs', fontsize=14)
ax[1].set_ylabel('Accuracy', fontsize=14)
ax[1].legend(fontsize=14)
ax[1].tick_params(axis='both', labelsize=12)
fig.suptitle(x=0.5, y=0.92, t="Lineplots showing loss and accuracy of CNN model by epochs", font
```

Figure 6: Codeblock for line plots

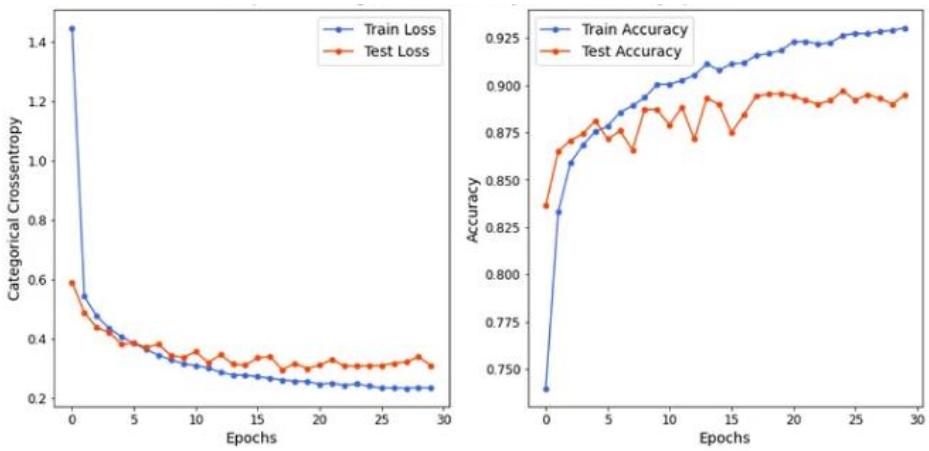


Figure 7: Lineplots showing loss and accuracy by epochs

Result and Discussion

A real-time gender detection system using CNN was developed. The system detects gender of the human being's image that is captured via the web camera of the system. Figure 8 shows a screenshot of the system.

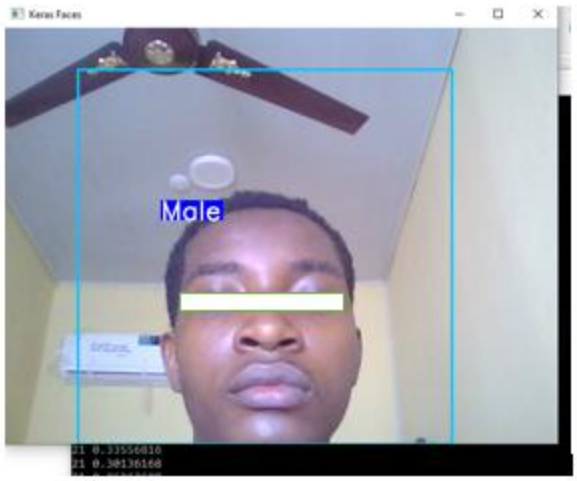


Figure 8:Real-time Gender detection system

5.0 Conclusion

The classification of human gender is a valuable tool for gathering data about and from individuals. The human face contains enough information to be useful in a variety of contexts. Gender classification is crucial for targeting the appropriate audience. We implemented a machine learning algorithm along with image processing techniques in this work, and the results look good for Kaggle data. To find out which optimizer produces the best output, several optimizers have been used. Following analysis of the data, a comparison between the efficacy of our model and our article has been provided, demonstrating where our system outperforms them.

Note that factors such as varying lighting conditions, facial expressions, and occlusions could affect the model's performance as they were not factored into the research, however, evaluation even with these factors, the model performed excellently.

References

[1] Uddin Md. Jashim, Dr. Paresh Chandra Barman, Khandaker TakdirAhmed S.M. Abdur Rahim , Abu Rumman Refat , Md Abdullah-AlImran6 "A Convolutional Neural Network for Real-time FaceDetection and Emotion & Gender Classification" IOSR Jo (2009)

[2] MS. Fathollahi, R. Heidari, “Gender classification from face images usingcentral difference convolutional networks,” Int. J. Multim. Inf. Retr., vol. 11,no. 4, pp. 695–703, 2022. [Online]. Available:https://doi.org/10.1007/s13735-022-00259-0

[3] A. Lahariya, V. Singh, US. Tiwary, “Real-time emotion and genderclassification using ensemble CNN,” CoRR, vol. abs/2111.07746, 2021.[Online]. Available: https://arxiv.org/abs/2111.07746

[4] O. Arriaga, M. Valdenegro-Toro, P. Pl¨oger, “Real-time convolutional neuralnetworks for emotion and gender classification,” CoRR, vol. abs/1710.07557,2017. [Online]. Available: http://arxiv.org/abs/1710. 07557

[5] Lian, H.-C., Lu, B.-L.: Multi-view gender classification using local binary patternsand support vector machines. In: Wang, J., Yi, Z., Zurada, J.M., Lu, B.-L., Yin,H. (eds.) ISNN 2006. LNCS, vol. 3972, pp. 202–209. Springer, Heidelberg (2006).https://doi.org/10.1007/11760023 30

[6] Li, P & Ma, X (2009) “Learning gender with support faces. IEEE Trans, vol,24 no 5

[7] Seaced , D (2009) “ A fast accurate unconstrained face detection. IEEE TRANS, vol 38,n02

[8] Onyemachi J N, Gift Adene, Belonwo T. S., Chinedu E. M., Adannaya U. G-A. (2024, March 11). *Digital Criminal Biometric Archives (DICA) and Public Facial Recognition System (FRS) for Nigerian criminal investigation using HAAR cascades classifier technique*. World Journal of Advanced Engineering Technology and Sciences. https://doi.org/10.30574/wjaets.2024.11.2.0077

[9] Sajja, T. (2009) “ Deep face recognition “. In proc-bmvc vol,1.2009.

[10] Rajesh, K & Gavi, K (2017) “Facial emotion analysis using deep CNN: IEEE No:339:443 vol 4 .2017

[11] Abdullah-Al-Imran MD “A Convolutional Neural Network for Realtime Face Detection and Emotion & Gender Classification’’ e-ISSN:2278-2834, p- ISSN: 2278-8735. Volume15, Issue 3, Ser. I (May -June2020), PP 37-46

[12] Happy SL and AurobindaRoutray“Automatic Facial ExpressionRecognition Using Features of Salient Facial Patches’’ DOI: 10.1109/TAFFC. 2014. 2386334 https://rb.gy/9m5dt2.

[13] Jang-Hee Yoo, So-Hee Park, and Yongjin Lee “Real-Time AgeandGender Estimation from Face Image” ISBN: 978-0-6480147-3-7.

[14] Octavio Arriaga1 and Matias Valdenegro-Toro and Paul G. Pl¨oger (2017) Real-time Convolutional Neural Networks for emotion and gender classification.”

[15] Yavanoglu, O., Aydos, M.: A review on cyber security datasets for machine learningalgorithms. In: 2017 IEEE International Conference on Big Data (Big Data), pp. 2186–2193(2017)(PDF) Machine Learning and Deep Learning Techniques for Cybersecurity: A Review.

[16] Khalil, T. (2020) “On convolutional neural network CNN, for real-time face detection & gender classification.

[17] Sajja, T & Voila, P (2009)" on comparison of feature extraction algorithms for gender classification from face images. Int, research technol, vol2,no5, 2009

[18] LeCun, Y., Bengio, Y., & Hinton, G. E. (2015, May 27). *Deep learning*. Nature. https://doi.org/10.1038/nature14539