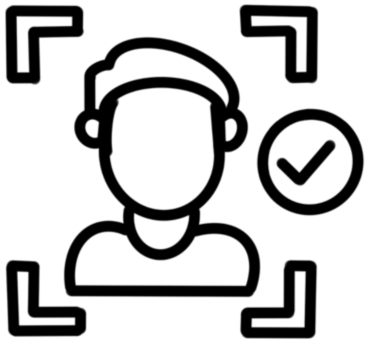
 [1]

Topic- Regression and Classification

Research Question- How can Regression and Classification supervised machine learning techniques for age and gender prediction respectively, be applied towards a facial recognition dataset(UTKFace)?

Subject: Computer Science

Word Count: 3984

Personal Code: jtk011

[1] Self-made on Procreate

Table of Contents

Intent

The advancement of computer vision is directly proportional to the development of Artificial intelligence (AI). In fact, Artificial Intelligence is a subfield of Computer Vision.[2] Development in computer vision results in advancement of Artificial Intelligence. Throughout out the years, the advancement in AI has resulted in machines having the ability to detect their surroundings. It is well known that the human brain can analyse and interpret complex data. Just like humans, modern computer vision models use the concept of neural networks to interpret and analyse complex data making it easier for processing and understanding many real life applications.

Worthiness

Exploring how CNNs fit into predicting demographic information is worthy of investigation as Convolutional Neural Networks (CNNs) are known to accurately predict age and gender by employing hierarchical feature extraction.[3] This technique enables the identification of subtle patterns within facial images, adding a layer of precision to the predictions. Features of Model depth, convolutional layers, and pooling operations all work together to improve the network's ability to identify complex age-related and gender-related stimuli.[4] Nonetheless, a wide range of intricate elements contribute to the difference in accuracy. The generalization of models is significantly influenced by the quality, size, and diversity of the data.[5] Complexities that affect the predicted outcomes are introduced by biases in the training data such as differences in the age distribution, and variations in the facial expressions.

---

[2] "What Is Computer Vision?" IBM, www.ibm.com/topics/computer-vision. Accessed 19 July. 2023
[3] Age and Gender Classification Using Convolutional Neural Networks ..., ieeexplore.ieee.org/document/7301352 Accessed 19 July. 2023
[4] Yamashita, Rikiya, et al. "Convolutional Neural Networks: An Overview and Application in Radiology - Insights into Imaging." SpringerOpen, Springer Berlin Heidelberg, 22 June 2018, insightsimaging.springeropen.com/articles/10.1007/s13244-018-0639-9 Accessed 21 July. 2023
[5] Lark, www.larksuite.com/en_us/topics/ai-glossary/training-data#. Accessed 29 Dec. 2023.

Attributes extracted from facial images play a vital role in numerous real life applications such as access control, video surveillance etc. Accurately finding one's age and gender can assist demographic data and could help in plotting various diagrams such as the population pyramid. If a machine could automatically determine how old the customer is from their facial detection in real time, it would be able to control the laws of underaged drinking, driving, entering a club, etc. Similarly, with the help of AI, the customer could get content generated based on people in the same age group's likings, customisable content could be showcased to the end-user based on their age and gender groups behaviour. Conventional machine learning techniques often include the manual definition of features, which may not always result in the best possible representations for the problems at hand.[6] Nonetheless, processed input may be assumed by deep neural networks for image identification, such as CNN, which then use a training process to determine the ideal network structure.

Age prediction is important for demographics, personalization, healthcare, and child protection. Traditional methods use hand-crafted features but may lack accuracy. CNNs, or Convolutional Neural Networks, are more effective for age prediction as they automatically learn features from facial data, offering improved accuracy and robustness. Thus, I thought that this research question would be worthy of investigation.

[6] Sarker, Iqbal H. "Machine Learning: Algorithms, Real-World Applications and Research Directions - SN Computer Science." SpringerLink, Springer Singapore, 22 Mar. 2021, link.springer.com/article/10.1007/s42979-021-00592-x. Accessed 11 Aug. 2023.

Scope

The scope of this extended essay is to implement supervised machine learning algorithms on a dataset called UTKFace for gender and age prediction using facial images. The scope also takes in account building and evaluating models for age and gender prediction from the facial images in the dataset. The experiment aims to assess the accuracy of supervised machine learning techniques in these tasks. Regression models will be used for age prediction, while classification algorithms will be used for gender prediction. This experimentation will include tasks such as data pre-processing where the data will be prepared and cleaned.[7] Moreover it will be explored thoroughly to find a pattern within the data. Afterwards, features will be extracted using either conventional approaches by convolutional neural networks. It is also important to see the composition of the dataset, exploring the different types of data it consists of, and how good the data quality is. I will explore the accuracy by changing the mix of ages, genders, and facial expressions in the dataset and see how well the model can predict age and gender. I will try to figure out the connection between the dataset's features, the design of the Convolutional Neural Networks (CNN), and how accurate the predictions are.

Background Information

Supervised learning is a process which involves training a model by using labelled data to improve the model's performance. During this process, the algorithm determines the connection between input features and its respective output labels. On the other hand, unsupervised learning deals with unlabelled data. The goal here is to find patterns or

---

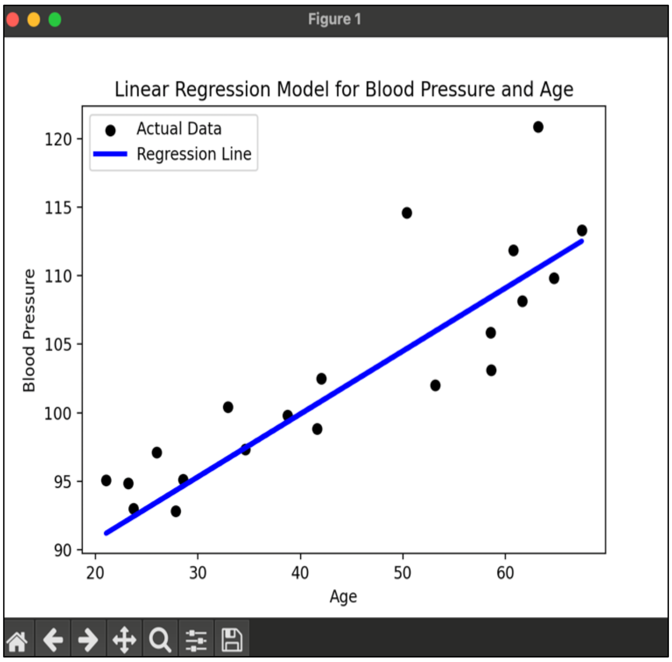
[7] Author links open overlay panelB. Abirami, et al. “Gender and Age Prediction from Real Time Facial Images Using CNN.” Materials Today: Proceedings, Elsevier, 24 Sept. 2020, www.sciencedirect.com/science/article/abs/pii/S2214785320362222. Accessed 28 July. 2023.

relationships in the data without having predefined output labels.[8] Reinforcement learning is all about training agents to make a series of decisions by interacting with an environment. The model gets feedback in the form of rewards or penalties based on its actions, helping it learn the best behaviour gradually.[9]

Regression analysis falls under supervised learning. Regression analysis uses a set of records generated by a function f(x). An adequate size of dataset including function parameter (x values) and corresponding output variables (y values) being would together be used to train a function for future predictability. The regression line is the best-fit line for this model. This function can then be applied to predict Y from an unknown X. To obtain the value of Y in a regression given X as independent characteristics, a function that predicts continuous Y is needed. Here, X is referred to as an independent variable and is also known as Y's predictor, while Y is referred to as the dependent or target variable. Regression may make use of a wide variety of functions or modules.[10] The most basic kind of function is a linear function. X might be one characteristic or a collection of traits that together reflect the issue. Classification, like regression, is also a type of supervised learning process. However, its purpose is to categorize or label data into distinct classes or groups. Instead of predicting a continuous value, classification algorithms aim to assign data points to predefined categories or classes based on their features.[11] A classic example for classification could be the selection of mathematics

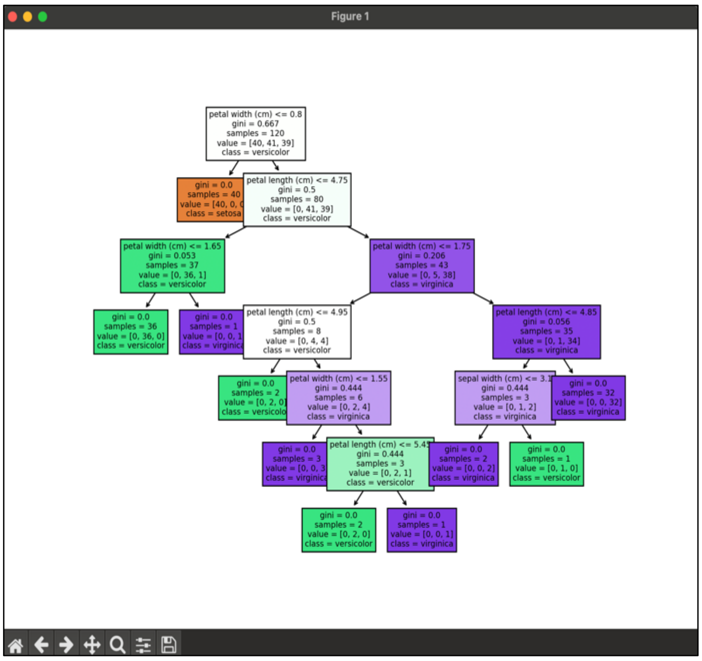
[8] "Supervised and Unsupervised Learning." GeeksforGeeks, GeeksforGeeks, 4 Dec. 2023, www.geeksforgeeks.org/supervised-unsupervised-learning/. Accessed 9 Dec. 2023.
[9] Hashemi-Pour, Cameron, and Joseph M. Carew. "What Is Reinforcement Learning?: Definition from TechTarget." Enterprise AI, TechTarget, 16 Aug. 2023, www.techtarget.com/searchenterpriseai/definition/reinforcement-learning#:~:text=Reinforcement%20learning%20is%20a%20machine,learn%20through%20trial%20and%20error. Accessed 11 Nov. 2023.
[10] "Regression in Machine Learning." GeeksforGeeks, GeeksforGeeks, 24 Jan. 2024, www.geeksforgeeks.org/regression-classification-supervised-machine-learning/#:~:text=Regression%20are%20used%20to%20predict,learning%20tasks%20in%20machine%20learning. Accessed 29 Jan. 2024.
[11] "Getting Started with Classification." GeeksforGeeks, GeeksforGeeks, 24 Jan. 2024, www.geeksforgeeks.org/getting-started-with-classification/. Accessed 26 Jan. 2024.

between Analysis and approaches and Applications and interpretations. Below are 2 models, 1 of regression and 1 of classification which I had made for better understanding.



Graph 1: Self-made on python. Code in Appendix A1.

The model explores the relationship between age and blood pressure in a population using a simple linear regression approach. Synthetic data is used for demonstration, with age as the independent variable and blood pressure as the dependent variable. The model is trained and evaluated to understand the correlation between age and blood pressure in the population.



Tree 1: Self-made on python. Code in Appendix A2.

The code uses the Iris dataset to train a Decision Tree Classifier, a machine learning model for classification tasks. After splitting the data into training and testing sets, the model is trained, and its accuracy is evaluated. The decision tree is then visualized using `matplotlib` to provide insights into the model's decision-making process.

Below is a table which I formulated to understand the difference between regression and classification.[12]

[12] "Regression vs Classification in Machine Learning - Javatpoint." *Www.Javatpoint.Com*, www.javatpoint.com/regression-vs-classification-in-machine-learning. Accessed 1 Jan. 2024.

Feature	Regression	Classification
Objective	Predict based on a continuous outcome	Assign an input to a discrete category or class.
Output (Nature of output)	Continuous values (Real numbers)	Discrete categories or classes (Categories or classes)
Loss Function	Mean Squared Error (MSE), Mean Absolute Error.	Cross-entropy, hinge loss.
Evaluation Metrics	Mean Squared Error (MSE), R-squared.	Accuracy, precision, recall, F1 score.
Output Interpretation	Predicted value represents a quantity.	Predicted class or probability of belonging to a class.

Table 1: *Stating the difference between regression and classification.*

Neural Networks (NN)

Neural networks, or NNs, are powerful machine learning models composed of interconnected nodes arranged in layers.[13] These nodes, process information through a series of matrix operations during feed-forward. The connections between nodes have weight values that determine the influence of one node's output on another. The ability to adjust these weights through a learning process enables Neural Networks to discern patterns for specific classifications. Each node includes a bias term, affecting its tendency to activate. The weighted sum of inputs, adjusted by these weights and biases, undergoes an activation function. When the sum surpasses a threshold, the activation function triggers, allowing the network to make decisions.[14] A simple design of a neural network is as follows[15]:

[13] "What Are Neural Networks?" IBM, www.ibm.com/topics/neural-networks#:~:text=Their%20name%20and%20structure%20are,layers%2C%20and%20an%20output%20layer. Accessed 8 Nov. 2023.
[14] "The Role of Weights and Bias in Neural Networks." GeeksforGeeks, GeeksforGeeks, 11 Oct. 2023, www.geeksforgeeks.org/the-role-of-weights-and-bias-in-neural-networks/. Accessed 18 Nov. 2023.
[15] Aaqil RahmanAaqil Rahman 1333 bronze badges, and MitikuMitiku 5. "Error with Checking Target: Expected Dense_49 to Have 4 Dimensions, but Got Array with Shape (2250, 3)." Stack Overflow, 1 Sept. 1964, stackoverflow.com/questions/53148901/error-with-checking-target-expected-dense-49-to-have-4-dimensions-but-got-arra. Accessed 27 Nov. 2023.

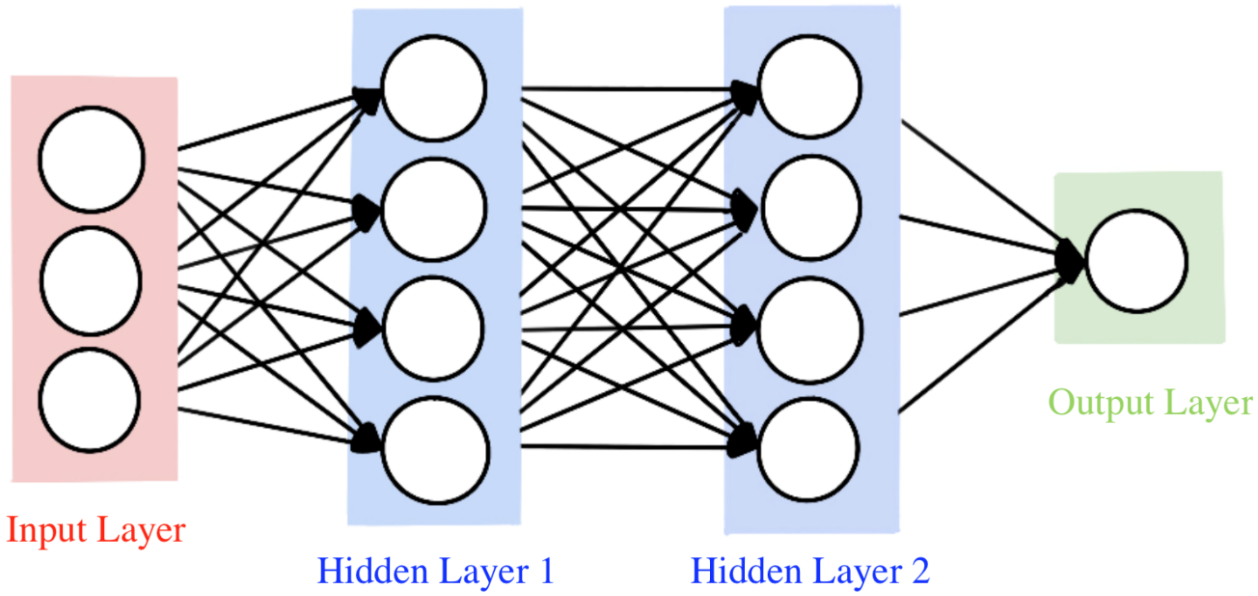


Image 1: *A self-made diagram on Procreate inspired from Stack Overflow on my knowledge of a Neural Network.*

The output layer, representing different classes or categories, activates the node corresponding to the correct class when a data vector is fed into the network.[16] During the training phase, if errors occur, the network adjusts its parameters using the negative gradient of an error function. In Convolutional Neural Networks (CNNs), the input layer processes data, often images. Neurons represent pixels or small image regions, with values indicating colour intensities. Hidden layers, the "brain" of the network, include convolutional layers for detecting features, pooling layers for efficiency, and fully connected layers for pattern interpretation.[17] The output layer serves as the network's "mouth," making final predictions. In image-related tasks, it might output probabilities for different categories. CNNs excel in tasks like image classification, leveraging their ability to learn hierarchical features.

[16] Ognjanovski, Gavril. "Everything You Need to Know about Neural Networks and Backpropagation - Machine Learning Made Easy..." Medium, Towards Data Science, 7 June 2020, towardsdatascience.com/everything-you-need-to-know-about-neural-networks-and-backpropagation-machine-learning-made-easy-e5285bc2be3a. Accessed 16 Oct. 2023.
[17] Mishra, Mayank. "Convolutional Neural Networks, Explained." Medium, Towards Data Science, 2 Sept. 2020, towardsdatascience.com/convolutional-neural-networks-explained-9cc5188c4939. Accessed 1 Jan. 2024.

Convolutional Neural Networks (CNN)

Convolutional neural networks (CNNs) are not significantly different from regular neural networks (NNs). In fact, CNN is a subset of NN, it's a type of NN. Their main difference between the two lies in the pre-processing of data through specialized layers before entering the standard fully-connected layers of NNs.[18]

When working with images, we could consider each pixel as a separate node in the input layer. However, this method quickly becomes demanding on computational resources as the number of input nodes grows. As more nodes are added, matrix operations become more computationally demanding due to the increased weights and biases. To tackle this issue, researchers found that using convolutions and sub-sampling in specialized layers allows neural networks to efficiently manage multidimensional data, like images, without compromising its performance.[19]

The convolutional and sub-sampling layers make the data more simpler while preserving its important and crucial features which simplifies the process of managing intricate details within the data. There are many layers in CNNs that use kernels or filters (Pre-selected matrices which are used to analyse the incoming image matrix. Through matrix multiplication, these matrices generate results which give information about the different characteristics which the image holds) to scan pictures.[20] Both kernels and filters have the job of moving through the image, one pixel at a time and to focus on a pixel at a time. They are required to extract the smallest

[18] "Convolutional Layer." Convolutional Layer - an Overview | ScienceDirect Topics, www.sciencedirect.com/topics/computer-science/convolutional-layer. Accessed 8 Sept. 2023.

[19] Fawzi, Alhussein, et al. "Discovering Faster Matrix Multiplication Algorithms with Reinforcement Learning." Nature News, Nature Publishing Group, 5 Oct. 2022, www.nature.com/articles/s41586-022-05172-4. Accessed 5 Jan. 2024.

[20] Dertat, Arden. "Applied Deep Learning - Part 4: Convolutional Neural Networks." Medium, Towards Data Science, 13 Nov. 2017, towardsdatascience.com/applied-deep-learning-part-4-convolutional-neural-networks-584bc134c1e2. Accessed 1 August. 2023.

details of each pixel which they focus on and generate a pattern. Once the pattern is generated, the CNN model improves its accuracy as it can make a connection with the test image to the pattern which was generated. [21] A simple design of a neural network is as follows:[22]

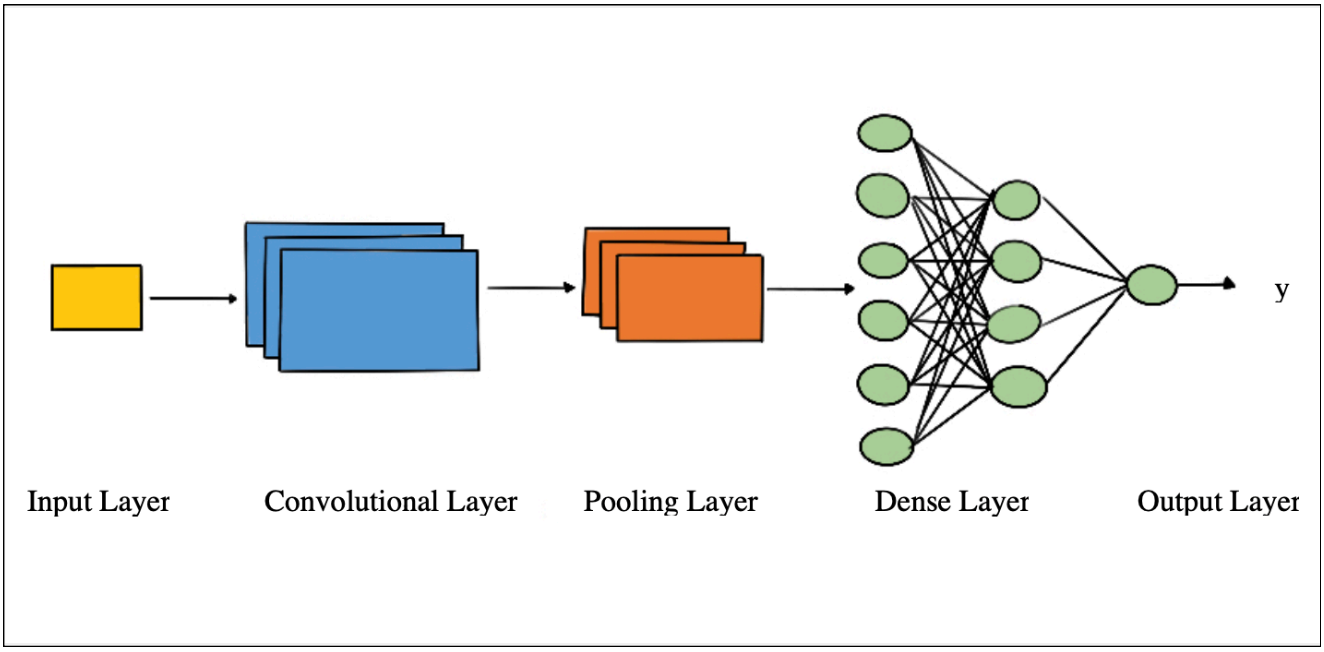


Image 2: *A self-made diagram on Procreate inspired from Stack Overflow on my knowledge of a Convolutional Neural Network.*

CNN's and Facial Recognition for Age and Gender Prediction

The objective and purpose of a CNN is too identify a pattern in an image which enables them to recognise and differentiate between various classes, objects and categories. CNNs are highly proficient in learning from images. By training a CNN with lots of different facial images, it can easily differentiate them and contribute in figuring out a pattern. The use of CNN will help us and make it easier for age and gender as they are known for excelling at recognising the smallest details in faces. The process starts with finding basic techniques which build up the

[21] Brownlee, Jason. "How Do Convolutional Layers Work in Deep Learning Neural Networks?" MachineLearningMastery.Com, 16 Apr. 2020, machinelearningmastery.com/convolutional-layers-for-deep-learning-neural-networks/. Accessed 1 Dec. 2023.
[22] Goud, Yeswanth. "Convolutional Neural Network (CNN) Using Tensorflow." LinkedIn, 19 Feb. 2023, www.linkedin.com/pulse/convolutional-neural-network-cnn-using-tensorflow-yeswanth-goud/?trk=public_post_main-feed-card_feed-article-content. Accessed 17 Nov. 2023.

face like edges and textures. Then, it moves on to more complex features like eyes, nose, and mouth.[23] To make CNNs work better, we can use hierarchical feature extraction. This helps them deal with changes in lighting, different poses, and various facial expressions.[24] People use CNNs in many ways, like in security systems, verifying users, and analysing emotions. They're a powerful tool for recognizing faces and are super useful in many applications.

Recent advancements in CNNs have made improvements in precision and effectiveness of facial recognition. These systems independently extract crucial visual features from images, utilizing innovations such as Siamese neural networks for verification and attention mechanisms that prioritize important facial areas.[25] For instance, if an individual intends to go clubbing or purchase alcohol with a counterfeit identification document, this facial recognition system could prevent the promotion of alcohol to underage individuals. Precise facial recognition is advantageous as it improves security and convenience. It provides a secure way to access devices, buildings, and accounts, while also offering convenient features such as unlocking smartphones and streamlining identity verification processes.

Experimental Setup and Methodology

Software

TensorFlow & Keras

TensorFlow is a machine learning platform, made by Google. It's great for building and training various types of smart computer models, especially neural networks as it implements various

[23] Le, Thai Hoang. "Applying Artificial Neural Networks for Face Recognition." Advances in Artificial Neural Systems, Hindawi, 3 Nov. 2011, www.hindawi.com/journals/aans/2011/673016/. Accessed 17 Nov. 2023.
[24] Author links open overlay panelJie Shao, et al. "Three Convolutional Neural Network Models for Facial Expression Recognition in the Wild." Neurocomputing, Elsevier, 10 May 2019, www.sciencedirect.com/science/article/abs/pii/S0925231219306137. Accessed 21 Nov. 2023.
[25] Recent Advances in Deep Learning Techniques for Face Recognition, arxiv.org/pdf/2103.10492.pdf. Accessed 21 Nov. 2023.

libraries, tools and resources in doing so I have decided to make my Machine Learning (ML) model with the usage of TensorFlow.[26] Keras is an Application Programming interface used in the TensorFlow platform.[27] This implements deep learning and will thus allow me to tune hyperparameters in my experiment to be able to arrive at the optimum value to achieve accuracy in gender and age prediction in my chosen dataset[28]. TensorFlow can help in sorting the selected dataset by changing the image size, making their colours contrast better, more visible and easier for the model to process, and creating more images to help your model learn better.[29]

The environment Used

Kaggle is a platform that focuses on data science and artificial intelligence. It allows users to not only upload and explore their datasets but also dive into datasets that others have shared. Additionally, data scientists can share pieces of code related to these datasets and engage in discussions with other professionals in the field to exchange insights and knowledge.[30]

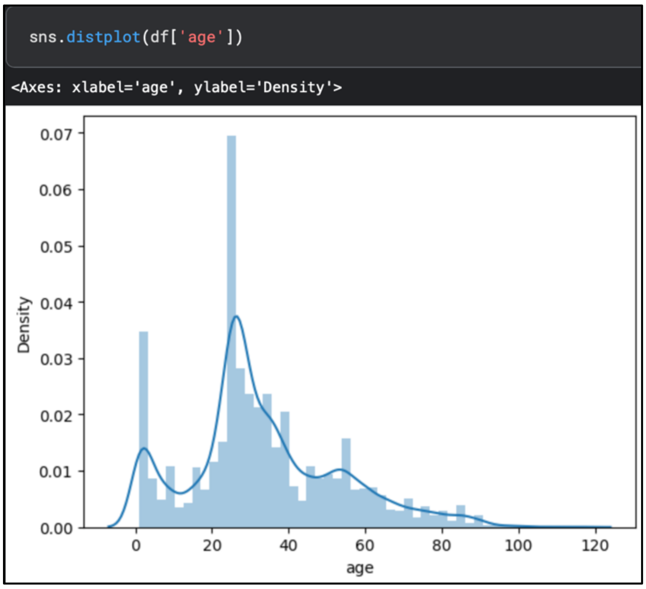
The Dataset used

The dataset was pre-existing dataset freely available on Kaggle.[31] The name of this data set is UTKFace. This dataset is a comprehensive collection of facial images that spans a wide range of ages, from new-borns to individuals as old as 116 years. It includes more than 23,000 facial

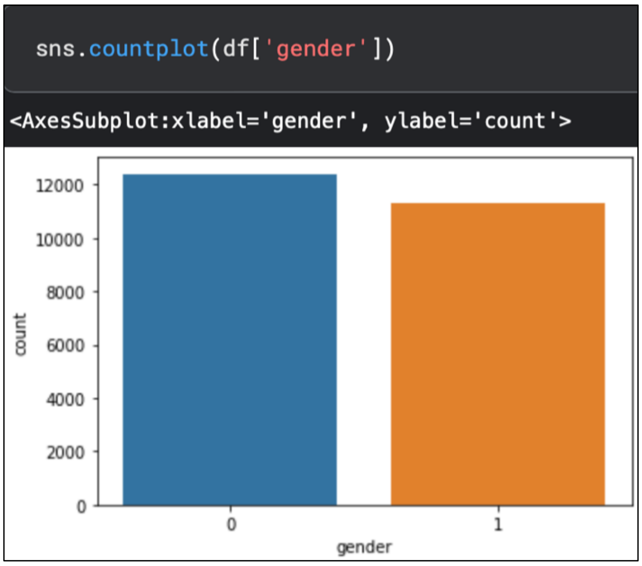
---

[26] Banoula, Mayank. “What Is Tensorflow? Deep Learning Libraries and Program Elements Explained.” Simplilearn.Com, Simplilearn, 16 Feb. 2023, www.simplilearn.com/tutorials/deep-learning-tutorial/what-is-tensorflow. Accessed 19 Dec. 2023.
[27] Ibid
[28] Ibid
[29] “Introduction to the Keras Tuner: Tensorflow Core.” TensorFlow, www.tensorflow.org/tutorials/keras/keras_tuner. Accessed 19 Dec. 2023.
[30] Uslu, Çağlar. “What Is Kaggle?” DataCamp, DataCamp, 16 Mar. 2022, www.datacamp.com/blog/what-is-kaggle. Accessed 22 Nov. 2023.
[31] Subedi, Sanjaya. “Utkface.” Kaggle, 16 Aug. 2018, www.kaggle.com/datasets/jangedoo/utkface-new/data. Accessed 22 Nov. 2023.

images, each with detailed annotations for age, gender, and ethnicity.[32] The images exhibit a diverse range of factors, including variations in pose, facial expressions, lighting conditions, occlusion, resolution, and more. This dataset is versatile and can be applied to various tasks such as facial detection, predicting age, modelling age progression or regression, and accurately locating facial landmarks.[33] Since I don't have the computational resources to be able to identify the ethnicity as well, each image was converted into 8-bit images - the images got converted into black and white so that it takes up less resources. These file names of the images are already sorted out based on their age, gender and ethnicity but we will not be focusing on ethnicity as it just another example of classification and we are already looking at gender for classification.



This density age graph shred's light on the variety of age's I have in my dataset and exhibits the wide range, from newborn to as old as 116.



This bar chart represents the how many people of different genders are present in my dataset.

[32] Ibid
[33] Ibid

Experimental Procedure

1. Importing specific models for the neural network: tensorflow, keras.preprocessing.image, keras.models, keras.layers, and keras.utils. We need to import models to get a hold of the various libraries which will help to create the neural network. Tensorflow as mentioned above is a machine learning platform, made by Google and Keras is an Application Programming interface used by TensorFlow.
2. Loading the data into the network, and creating three lists are for image paths, age labels, and gender labels. I will then load the UTKFace Dataset.
3. Converting the dataset into a DataFrame with 'image', 'age', and 'gender' columns. This will help in image classification.
4. Creating a gender dictionary to map gender labels to strings for easier interpretation.
5. Performing exploratory data analysis, displaying the age distribution, class distribution, and a set of images.
6. Pre-processing the data for feature extraction, and a function is defined to extract all the features from the images.
7. Images are resized to 128x128 and converted to grayscale to save memory.
8. Normalize the features and convert them into a NumPy array for the neural network.
9. Convert the gender and age labels to NumPy arrays.
10. Create a convolutional neural network with two outputs: one for classification (gender) and one for regression (age).
11. Train the data using the fit method, and then graph the progress.
12. Plot both the accuracy and loss graphs for both training and validation sets to see the accuracy and validation difference.
13. Analyse and evaluate the result of classification using confusion matrix and regression using mean squared error.

14. Compare the original and predicted gender and age that are displayed

Complete code has been attached in Appendix B.

Experimental Results

Data: 30th Epoch Performance

I made this table by gathering the performance metrics from the 30th training epoch for each cross-validation fold. This data table is the average data table of 3 trials. (Appendix C1, C2, C3).

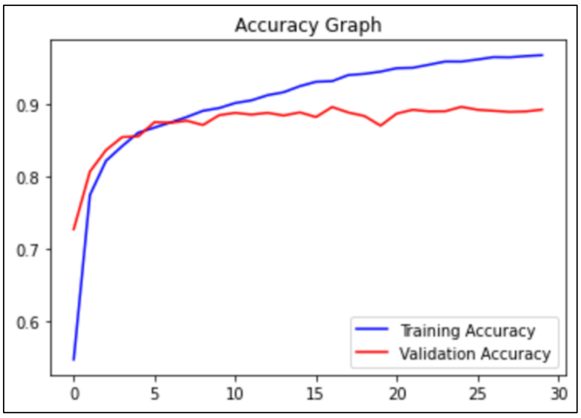
Age and Gender Prediction:

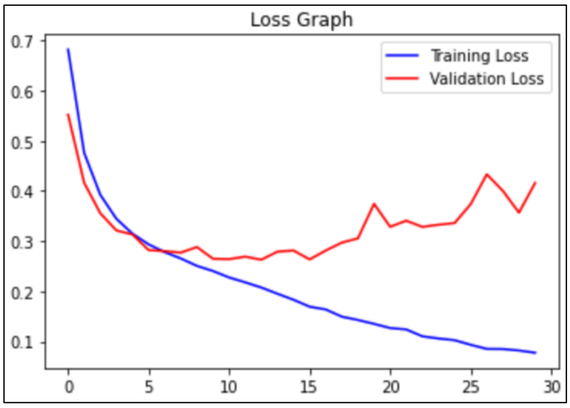
Test No.	Loss:	Accuracy:	Age Precision:	Gender Precision	Recall:
1	3.8026	0.97	Predicted Age: 2 Actual Age: 3	Predicted Gender:0 Actual Gender:1	1.0
2	3.8365	0.98	Predicted Age: 14 Actual Age: 15	Predicted Gender:0 Actual Gender:0	0.99
3	3.8112	0.94	Predicted Age: 27 Actual Age: 28	Predicted Gender:0 Actual Gender:0	0.99
4	3.7968	0.96	Predicted Age: 42 Actual Age: 40	Predicted Gender: 1 Actual Gender: 1	0.98
5	3.8722	0.96	Predicted Age: 64 Actual Age: 61	Predicted Gender: 1 Actual Gender: 1	0.99
6	3.7962	0.97	Predicted Age: 78 Actual Age: 78	Predicted Gender: 0 Actual Gender: 0	1.0
7	3.8095	0.97	Predicted Age: 92 Actual Age: 93	Predicted Gender: 1 Actual Gender: 1	0.98

Data Table 1- *Average of 3 trials taken*

Result Graph

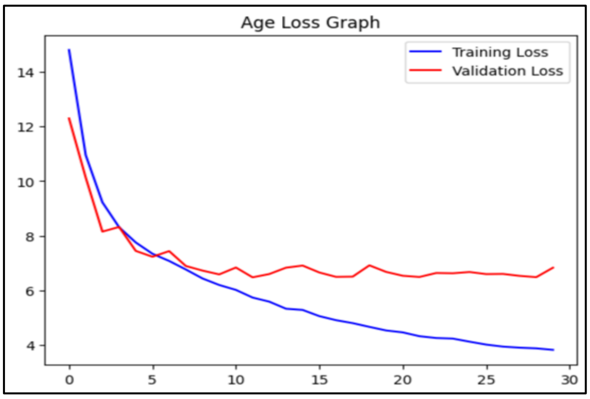
Results for Gender

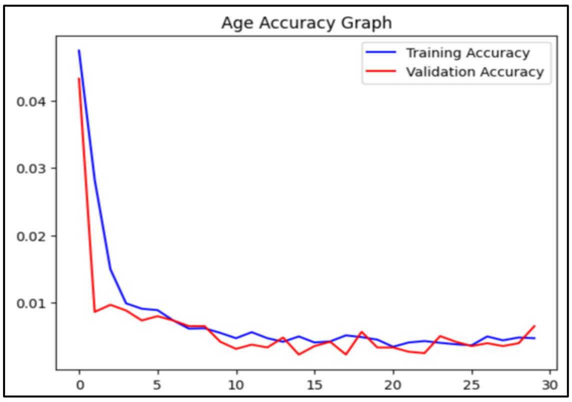




Results Graphs for gender

Results for Age





Results Graphs for age

Evaluation and Conclusion

Evaluation

Two important measures are used to evaluate the model's performance during training: validation loss and validation accuracy. The proportion of properly categorized instances in the validation set is measured as validation accuracy, which offers insight into the model's classification precision on unobserved data. The ratio of accurately predicted samples to the total number of validation samples is used to compute it. On the other hand, the validation loss represents the value of the loss function calculated on the validation set.

While training a CNN, one of the key objectives is to minimize the training loss. It is also important to consider the training loss as a factor as to why there is a chance of error in the result as some data was not completely loaded and trained.[34] It is essential to keep an eye on both validation loss and accuracy when evaluating the learning curve of the CNN. This helps identify any indications of overfitting or underfitting and ensures optimal performance in real-world scenarios.

Overfitting happens when a model overly fixates on training data, capturing irrelevant details or unusual patterns, causing subpar performance with new, unseen data. On the flip side, underfitting happens when a model is too simplistic, failing to comprehend underlying patterns in the training data, resulting in poor performance on both the training and new data. The dataset was further reduced to 5000 images. The benefit of taking a reduced dataset is that I can compare it with the results I obtained from the original dataset. This encourages the model to learn more generalizable patterns. By reducing the dataset, we can also examine if it is compatible by less computationally strong systems and it also reduces redundancies in the data. The average result after 3 trials was as follows (All 3 trials can be found in Appendix D1, D2 and D3):

[34] Irina, et al. "Improving Validation Loss and Accuracy for CNN." Data Science Stack Exchange, 1 July 1965, datascience.stackexchange.com/questions/55963/improving-validation-loss-and-accuracy-for-cnn. Accessed 5 Jan. 2024.

Test No.	Loss:	Accuracy:	Age Precision:	Gender Precision	Recall:
1	4.4038	0.78	Predicted Age: 5 Actual Age: 3	Predicted Gender:0 Actual Gender:1	0.80
2	4.7394	0.75	Predicted Age: 13 Actual Age: 15	Predicted Gender:1 Actual Gender:0	0.76
3	4.3175	0.79	Predicted Age: 33 Actual Age: 28	Predicted Gender:0 Actual Gender:0	0.82
4	4.7956	0.74	Predicted Age: 42 Actual Age: 40	Predicted Gender: 1 Actual Gender: 1	0.75
5	4.5767	0.76	Predicted Age: 56 Actual Age: 61	Predicted Gender: 1 Actual Gender: 1	0.77
6	4.9962	0.73	Predicted Age: 73 Actual Age: 78	Predicted Gender: 0 Actual Gender: 0	0.76
7	4.4494	0.79	Predicted Age: 98 Actual Age: 93	Predicted Gender: 1 Actual Gender: 1	0.82

Data Table 2- *Average of 3 trials taken of reduced dataset*

Analysis of Regression

Mean Squared Error (MSE) is a common metric used to evaluate the performance of regression models. It measures the average squared difference between the predicted values and the actual values in a dataset. I will be using Mean squared Error to evaluate my regression model of age prediction.

Mathematically, it is expressed as:[35]

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

[35] Frost, Jim. "Mean Squared Error (MSE)." Statistics By Jim, 28 May 2023, statisticsbyjim.com/regression/mean-squared-error-mse/. Accessed 9 Jan. 2024.

Where:

- n is the number of data points
- y_i is the actual target value for the i-th data point
- $\hat{y}_i$ is the predicted value for the i-th data point.[36]

Advantages of using MSE[37]:

- Calculates Prognosis Accuracy: The Mean Squared Error (MSE) indicates how well the model's predictions agree with the observed data. Better accuracy is indicated by a lower MSE.
- Emphasizes Larger mistakes: By squaring the mistakes in the MSE, the larger errors are given more weight. In certain circumstances, particularly when greater mistakes are more significant, this may be advantageous.
- Mathematical Convenience: Squaring errors reduce the complexity of the equations, which facilitates their manipulation and differentiation during optimization procedures.[38]

Working out the MSE for the average original result:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

[36] Ibid

[37] M, Padhma. "A Comprehensive Introduction to Evaluating Regression Models." Analytics Vidhya, 30 Nov. 2023, www.analyticsvidhya.com/blog/2021/10/evaluation-metric-for-regression-models/. Accessed 15 Jan. 2024.

[38] Ibid

$$MSE = \frac{1}{7}\sum_{i=1}^{7}(y_i - \hat{y}_i)^2$$

$$MSE = \frac{1}{7}\sum_{i=1}^{7}(y_i - \hat{y}_i)^2$$

$$(3-2)^2, (15-14)^2, (28-27)^2, (42-40)^2, (64-61)^2, (78-78)^2, (93-92)^2$$

$$(1)^2 + (1)^2 + (1)^2 + (2)^2 + (3)^2 + (0)^2 + (1)^2$$

$$= 15$$

$$\frac{1}{7} \times 15 = \frac{15}{7}$$

Working out the MSE for the average the reduced data result:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

$$MSE = \frac{1}{7}\sum_{i=1}^{7}(y_i - \hat{y}_i)^2$$

$$MSE = \frac{1}{7}\sum_{i=1}^{7}(y_i - \hat{y}_i)^2$$

$$(3-5)^2, (15-13)^2, (28-33)^2, (40-42)^2, (61-56)^2, (78-73)^2, (93-98)^2$$

$$(2)^2 + (-2)^2 + (-5)^2 + (-2)^2 + (5)^2 + (5)^2 + (-5)^2$$

$$= 112$$

$$\frac{1}{7} \times 122 = 16$$

My target variable is age, and age my dataset has a wide range of age, from as young as 0 years old to as old as 116 years old people. Having a MSE of 16 indicates that my model could be considered as reasonably accurate model. [39]

Analysis of Classification

Confusion matrix evaluates the performance of classification models by tabulating true positive, true negative, false positive, and false negative predictions. It provides a detailed snapshot of a model's ability to correctly classify instances.[40] I will be using Confusion Matrix to evaluate my regression model of age prediction.

Advantages of using a binary confusion matrix:

- Quantitative analysis- A confusion matrix can prove a quantitative matrix which could help in evaluating the model's performance in terms of accuracy, recall, precision.[41]
- Threshold Adjustment- A confusion matrix can help to identify how small changes in the values of the confusion matrix for gender changes affect the model's performance. [42]
- Simplicity- A confusion matrix is straightforward and easily comprehendible. It breaks down predictions into 4 categories as represented in the diagram above. TP- True Positive, TN- True Negative, FP- False Positive, FN- False Negative.[43]

[39] Ibid

[40] "What Is a Confusion Matrix?: Machine Learning Glossary: Encord." Encord, encord.com/glossary/confusion-matrix/#:~:text=A%20confusion%20matrix%20is%20a,effectiveness%20of%20the%20model%27s%20predictions. Accessed 15 Jan. 2024.

[41] Bhandari, Aniruddha. "Understanding & Interpreting Confusion Matrix in Machine Learning (Updated 2024)." *Analytics Vidhya*, 11 Jan. 2024, www.analyticsvidhya.com/blog/2020/04/confusion-matrix-machine-learning/. Accessed 15 Jan. 2024.

[42] Ibid

[43] Ibid

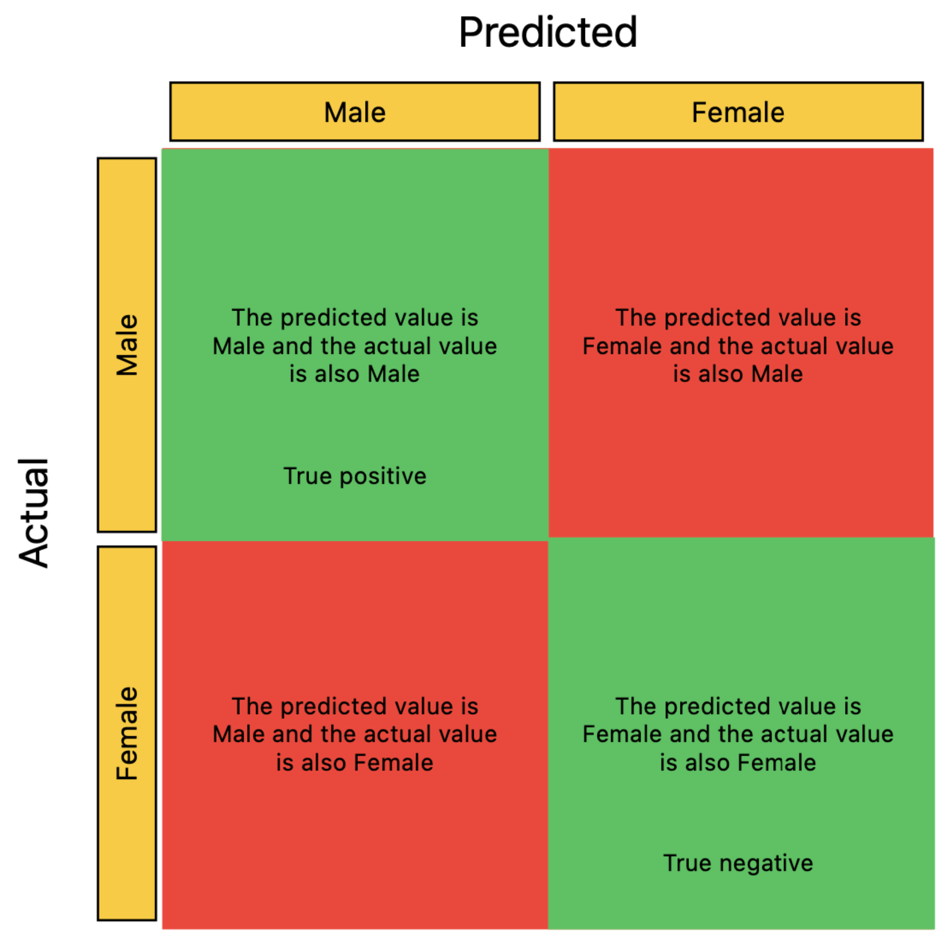


Diagram Self-made on Freeform

Diagram 1: *This shows a representation of my interpretation of a binary confusion matrix.*

Working out the Confusion for the average original data result:

	Predicted		
Actual		Male	Female
	Male	3	0
	Female	1	3

Confusion Matrix 1- *This tests for gender prediction*

To evaluate this matrix I will find out its accuracy, precision, recall and F1 score. To find the values, I will be using these formulas:

$Precision = \frac{TP}{TP+FP}$, where TP is the total true positives and FP is the false positive[44]

$Recall = \frac{TP}{TP+FN}$, where TP is the total true positives and FP is the false negative[45]

$$F1\ Score = 2 \times \frac{Precision\ \times\ Recall}{Precision + Recall}$$ [46]

Recall is the percentage of true positives among all real positives. True positives are instances that are accurately recognized as positive, false negatives are cases that are wrongly identified as negative, and false positives are cases that are incorrectly labelled as positive.[47]

Class	n(Truth)	n(classified)	Accuracy	Precision	Recall	F1 Score
Male	4	3	85.71%	1.0	0.75	0.86
Female	3	4	85.71%	0.75	1.0	0.86

Data Table 3- *This table has the results for all calculations made.*

[44] Ibid
[45] Ibid
[46] Ibid
[47] "Precision and Recall: How to Evaluate Your Classification Model." *Built In*, builtin.com/data-science/precision-and-recall#. Accessed 18 Jan. 2024.

A **F1 Score of 0.86** could be considered as **very good** especially for binary classification. F1 score is called the **harmonic mean** of precision and recall. A value of 0.86 shows that there is a **fair and balanced compromise** in between precision and recall which shows that the model is preforming well in correctly identifying the various genders.

Working out the Confusion Matrix for the average the reduced data result:

	Predicted		
Actual		Male	Female
	Male	2	1
	Female	1	3

Confusion Matrix 1- *This tests for gender prediction Of the reduced dataset*

Class	n(Truth)	n(classified)	Accuracy	Precision	Recall	F1 Score
Male	3	3	71.43%	0.67	0.67	0.67
Female	4	4	71.43%	0.75	0.75	0.75

Data Table 4- *This table has the results for all calculations made of the reduced dataset.*

A F1 Score of **0.67 and 0.75** could be considered relativity **not so good**. Reducing the dataset has made it **difficult for accurately predicting the gender**. There has been a drastic **drop of**

14.28% in the accuracy. There has been **drastic falls** in the precision, recall and F1 scores as well. This indicates that a larger dataset with more training will provide more assuring and better results.

As mentioned in my research MSE measures the **average squared difference between the predicted values and the actual values** in a dataset. An MSE of $\frac{15}{7}$ is a **relatively very good** considering was just over **26000** and there are more than **7 million** face. I also mentioned that a confusion matrix provides a detailed snapshot of a model's ability to correctly classify instances. With an **F1 score of 0.86**, we can say that the model can is also very accurate as there are training data which might confuse the model. **A woman with short hair or vice versa might also contradict the model**. I also mentioned that reducing the dataset helps the model to learn **more generalizable patterns** in my research but I turned out to **see the opposite**.

Conclusion

In this Extended Essay, I was set out to explore how machine learning techniques could be applied towards a dataset and I created a model using Neural networks. I used Regression and classification supervised machine learning techniques and applied to the UTKFace dataset for age and gender prediction respectively in facial recognition tasks. For age prediction, regression algorithms were used to train on facial features extracted from the dataset to predict the age of individuals accurately. Features like wrinkles, skin texture, and facial structure were also utilized to infer age. Similarly, for gender prediction, classification algorithms was employed. These algorithms learn patterns from facial features such as jawline, eyebrow shape, and lip thickness to classify individuals into male or female categories. Both supervised machine learning techniques gave positive results with the original dataset. All the **accuracies were above 94%** for age predictions and the **mean squared error of** $\frac{15}{7}$ which is considered to be really good. Similarly for gender prediction, the results were put through a **binary confusion matrix** and the **F1 score was 0.86** which is a really good score but the reduced dataset had an F1 score of 0.67 and 0.75 which reduced the accuracy.

Limitations and potential improvements

Because just three trials were employed in this study and their findings were averaged, the resulting data was greatly distorted by anomalies that were either extraordinarily big or extremely minor. To combat this, additional trials can be carried out, and their median could be used rather than their mean to reduce the impact of outliers and guarantee a more accurate result.

Further Scope and Research

For Further Scope of research, I could considering changing architecture parameter factors such as Number of layers, Number of filters, Filter size, Pooling size and Stride.

- Number of layers: Influences the depth and complexity of the network, potentially affecting model capacity and ability to capture hierarchical features.
- Number of filters: Determines the richness of feature representation at each layer, impacting the network's ability to learn discriminative patterns.
- Filter size: Dictates the receptive field of each convolutional operation, affecting the scale of features extracted and the level of detail captured.
- Pooling size: Controls the down sampling of feature maps, affecting spatial information retention and the network's robustness to spatial translations.

- Stride: Defines the step size of the convolutional operation, influencing the spatial resolution of feature maps and the amount of overlap between receptive fields, impacting feature extraction efficiency.

Moreover there are different learning parameters which I can explore such as the learning rate, batch size and epochs. Making these the independent variable would explain which learning parameters would make my model the most effective. While researching, I came across something known as Hyperparameter Turning, where all the optimization algorithms, activation functions, and initialization methods for weights are performed. In my research as well, for predicting age and gender through supervised learning adjusting hyperparameters ensures the model works effectively and accurately. By systematically trying out various hyperparameter settings, we pinpoint the best combination. This leads to faster convergence during training and better adaptability to new data. Methods like grid search, random search, or more sophisticated techniques such as Bayesian optimization or evolutionary algorithms are used to explore these hyperparameters efficiently. This process boosts the model's performance on the specific dataset, ensuring it performs optimally for age and gender prediction in facial recognition tasks. Researching more into that area of knowledge would definitely benefit my research.

Bibliography

Aaqil RahmanAaqil Rahman 1333 bronze badges, and MitikuMitiku 5. “Error with Checking Target: Expected Dense_49 to Have 4 Dimensions, but Got Array with Shape (2250, 3).” Stack Overflow, 1 Sept. 1964, stackoverflow.com/questions/53148901/error-with-checking-target-expected-dense-49-to-have-4-dimensions-but-got-arra. Accessed 27 Nov. 2023.

Age and Gender Classification Using Convolutional Neural Networks ..., ieeexplore.ieee.org/document/7301352. Accessed 19 July. 2023

Author links open overlay panelB. Abirami, et al. “Gender and Age Prediction from Real Time Facial Images Using CNN.” Materials Today: Proceedings, Elsevier, 24 Sept. 2020, www.sciencedirect.com/science/article/abs/pii/S2214785320362222. Accessed 28 July. 2023.

Author links open overlay panelJie Shao, et al. “Three Convolutional Neural Network Models for Facial Expression Recognition in the Wild.” Neurocomputing, Elsevier, 10 May 2019, www.sciencedirect.com/science/article/abs/pii/S0925231219306137. Accessed 21 Nov. 2023.

Banoula, Mayank. “What Is Tensorflow? Deep Learning Libraries and Program Elements Explained.” Simplilearn.Com, Simplilearn, 16 Feb. 2023, www.simplilearn.com/tutorials/deep-learning-tutorial/what-is-tensorflow. Accessed 19 Dec. 2023.

Bhandari, Aniruddha. “Understanding & Interpreting Confusion Matrix in Machine Learning (Updated 2024).” *Analytics Vidhya*, 11 Jan. 2024, www.analyticsvidhya.com/blog/2020/04/confusion-matrix-machine-learning/. Accessed 15 Jan. 2024.

Brownlee, Jason. “How Do Convolutional Layers Work in Deep Learning Neural Networks?” MachineLearningMastery.Com, 16 Apr. 2020, machinelearningmastery.com/convolutional-layers-for-deep-learning-neural-networks/. Accessed 1 Dec. 2023.

“Convolutional Layer.” Convolutional Layer - an Overview | ScienceDirect Topics, www.sciencedirect.com/topics/computer-science/convolutional-layer. Accessed 8 Sept. 2023.

Dertat, Arden. “Applied Deep Learning - Part 4: Convolutional Neural Networks.” Medium, Towards Data Science, 13 Nov. 2017, towardsdatascience.com/applied-deep-learning-part-4-convolutional-neural-networks-584bc134c1e2. Accessed 1 August. 2023.

Fawzi, Alhussein, et al. “Discovering Faster Matrix Multiplication Algorithms with Reinforcement Learning.” Nature News, Nature Publishing Group, 5 Oct. 2022, www.nature.com/articles/s41586-022-05172-4. Accessed 5 Jan. 2024.

Frost, Jim. “Mean Squared Error (MSE).” Statistics By Jim, 28 May 2023, statisticsbyjim.com/regression/mean-squared-error-mse/. Accessed 9 Jan. 2024.

“Getting Started with Classification.” GeeksforGeeks, GeeksforGeeks, 24 Jan. 2024, www.geeksforgeeks.org/getting-started-with-classification/. Accessed 26 Jan. 2024.

Goud, Yeswanth. “Convolutional Neural Network (CNN) Using Tensorflow.” LinkedIn, 19 Feb. 2023, www.linkedin.com/pulse/convolutional-neural-network-cnn-using-tensorflow-yeswanth-goud/?trk=public_post_main-feed-card_feed-article-content. Accessed 17 Nov. 2023.

Hashemi-Pour, Cameron, and Joseph M. Carew. “What Is Reinforcement Learning?: Definition from TechTarget.” Enterprise AI, TechTarget, 16 Aug. 2023, www.techtarget.com/searchenterpriseai/definition/reinforcement-learning#:~:text=Reinforcement%20learning%20is%20a%20machine,learn%20through%20trial%20and%20error. Accessed 11 Nov. 2023.

“Introduction to the Keras Tuner: Tensorflow Core.” TensorFlow, www.tensorflow.org/tutorials/keras/keras_tuner. Accessed 19 Dec. 2023.

Irina, et al. “Improving Validation Loss and Accuracy for CNN.” Data Science Stack Exchange, 1 July 1965, datascience.stackexchange.com/questions/55963/improving-validation-loss-and-accuracy-for-cnn. Accessed 5 Jan. 2024.

Lark, www.larksuite.com/en_us/topics/ai-glossary/training-data#. Accessed 29 Dec. 2023.

Le, Thai Hoang. “Applying Artificial Neural Networks for Face Recognition.” Advances in Artificial Neural Systems, Hindawi, 3 Nov. 2011, www.hindawi.com/journals/aans/2011/673016/. Accessed 17 Nov. 2023.

Mishra, Mayank. “Convolutional Neural Networks, Explained.” Medium, Towards Data Science, 2 Sept. 2020, towardsdatascience.com/convolutional-neural-networks-explained-9cc5188c4939. Accessed 1 Jan. 2024.

M, Padhma. “A Comprehensive Introduction to Evaluating Regression Models.” Analytics Vidhya, 30 Nov. 2023, www.analyticsvidhya.com/blog/2021/10/evaluation-metric-for-regression-models/. Accessed 15 Jan. 2024.

Ognjanovski, Gavril. “Everything You Need to Know about Neural Networks and Backpropagation - Machine Learning Made Easy...” Medium, Towards Data Science, 7 June 2020, towardsdatascience.com/everything-you-need-to-know-about-neural-networks-and-backpropagation-machine-learning-made-easy-e5285bc2be3a. Accessed 16 Oct. 2023.

“Precision and Recall: How to Evaluate Your Classification Model.” *Built In*, builtin.com/data-science/precision-and-recall#. Accessed 18 Jan. 2024.

Recent Advances in Deep Learning Techniques for Face Recognition, arxiv.org/pdf/2103.10492.pdf. Accessed 21 Nov. 2023.

“Regression in Machine Learning.” GeeksforGeeks, GeeksforGeeks, 24 Jan. 2024, www.geeksforgeeks.org/regression-classification-supervised-machine-learning/#:~:text=Regression%20are%20used%20to%20predict,learning%20tasks%20in%20machine%20learning. Accessed 29 Jan. 2024.

“Regression vs Classification in Machine Learning - Javatpoint.” *Www.Javatpoint.Com*, www.javatpoint.com/regression-vs-classification-in-machine-learning. Accessed 1 Jan. 2024

Sarker, Iqbal H. “Machine Learning: Algorithms, Real-World Applications and Research Directions - SN Computer Science.” SpringerLink, Springer Singapore, 22 Mar. 2021, link.springer.com/article/10.1007/s42979-021-00592-x. Accessed 11 Aug. 2023.

Subedi, Sanjaya. “Utkface.” Kaggle, 16 Aug. 2018, www.kaggle.com/datasets/jangedoo/utkface-new/data. Accessed 22 Dec. 2023.

“Supervised and Unsupervised Learning.” GeeksforGeeks, GeeksforGeeks, 4 Dec. 2023, www.geeksforgeeks.org/supervised-unsupervised-learning/. Accessed 9 Dec. 2023.

“The Role of Weights and Bias in Neural Networks.” GeeksforGeeks, GeeksforGeeks, 11 Oct. 2023, www.geeksforgeeks.org/the-role-of-weights-and-bias-in-neural-networks/. Accessed 18 Nov. 2023.

Uslu, Çağlar. “What Is Kaggle?” DataCamp, DataCamp, 16 Mar. 2022, www.datacamp.com/blog/what-is-kaggle. Accessed 22 Dec. 2023.

“What Are Neural Networks?” IBM, www.ibm.com/topics/neural-networks#:~:text=Their%20name%20and%20structure%20are,layers%2C%20and%20an%20output%20layer. Accessed 8 Nov. 2023.

“What Is Computer Vision?” IBM, www.ibm.com/topics/computer-vision. Accessed 19 July. 2023.

“What Is a Confusion Matrix?: Machine Learning Glossary: Encord.” Encord, encord.com/glossary/confusion-matrix/#:~:text=A%20confusion%20matrix%20is%20a,effectiveness%20of%20the%20model%27s%20predictions. Accessed 15 Jan. 2024.

Yamashita, Rikiya, et al. “Convolutional Neural Networks: An Overview and Application in Radiology - Insights into Imaging.” SpringerOpen, Springer Berlin Heidelberg, 22 June 2018, insightsimaging.springeropen.com/articles/10.1007/s13244-018-0639-9 Accessed 21 July. 2023

Appendix

Appendix A1- Code used for Regression Model.

```
# Import necessary libraries
import numpy as np
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error
import matplotlib.pyplot as plt

# Generate synthetic data
np.random.seed(42)
age = np.random.uniform(20, 70, 100)
blood_pressure = 80 + 0.5 * age + np.random.normal(0, 5, 100)

# Create a DataFrame
data = pd.DataFrame({'Age': age, 'Blood_Pressure': blood_pressure})

# Split the data into training and testing sets
X = data[['Age']]
y = data['Blood_Pressure']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Create a linear regression model
model = LinearRegression()

# Train the model on the training data
model.fit(X_train, y_train)

# Make predictions on the test set
y_pred = model.predict(X_test)

# Evaluate the model performance
mse = mean_squared_error(y_test, y_pred)
print(f'Mean Squared Error: {mse}')
```

```
# Plot the data points and the regression line
plt.scatter(X_test, y_test, color='black', label='Actual Data')
plt.plot(X_test, y_pred, color='blue', linewidth=3, label='Regression Line')
plt.xlabel('Age')
plt.ylabel('Blood Pressure')
plt.title('Linear Regression Model for Blood Pressure and Age')
plt.legend()
plt.show()
```

Appendix A2- Code used for Classification Tree Model.

```
# Import necessary libraries
from sklearn.datasets import load_iris
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier
from sklearn import metrics
import matplotlib.pyplot as plt
from sklearn.tree import plot_tree

# Load the iris dataset
iris = load_iris()
X = iris.data
y = iris.target

# Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Create a decision tree classifier
clf = DecisionTreeClassifier()

# Train the classifier on the training data
clf.fit(X_train, y_train)

# Make predictions on the testing data
y_pred = clf.predict(X_test)

# Evaluate the model
```

accuracy = metrics.accuracy_score(y_test, y_pred)

print(f"Accuracy: {accuracy:.2f}")

Plot the decision tree

plt.figure(figsize=(12, 8))

plot_tree(clf, filled=True, feature_names=iris.feature_names, class_names=iris.target_names)

plt.show()

Appendix B- Code used for my Model

Import Modules

```
import pandas as pd
import numpy as np
import os
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
from tqdm.notebook import tqdm
warnings.filterwarnings('ignore')
%matplotlib inline

import tensorflow as tf
from keras.preprocessing.image import load_img
from keras.models import Sequential, Model
from keras.layers import Dense, Conv2D, Dropout, Flatten, MaxPooling2D, Input
```

Load the Dataset

```
BASE_DIR = '../input/utkface-new/UTKFace/'
```

```
# labels - age, gender, ethnicity
image_paths = []
age_labels = []
gender_labels = []

for filename in tqdm(os.listdir(BASE_DIR)):
    image_path = os.path.join(BASE_DIR, filename)
    temp = filename.split('_')
    age = int(temp[0])
    gender = int(temp[1])
    image_paths.append(image_path)
    age_labels.append(age)
    gender_labels.append(gender)
```

```
# convert to dataframe
df = pd.DataFrame()
df['image'], df['age'], df['gender'] = image_paths, age_labels, gender_labels
df.head()
```

```
# map labels for gender
gender_dict = {0:'Male', 1:'Female'}
```

Exploratory Data Analysis

```
from PIL import Image
img = Image.open(df['image'][0])
plt.axis('off')
plt.imshow(img);
```

```
sns.distplot(df['age'])
```

```
sns.countplot(df['gender'])
```

```
# to display grid of images
plt.figure(figsize=(20, 20))
files = df.iloc[0:25]

for index, file, age, gender in files.itertuples():
    plt.subplot(5, 5, index+1)
    img = load_img(file)
    img = np.array(img)
    plt.imshow(img)
    plt.title(f"Age: {age} Gender: {gender_dict[gender]}")
    plt.axis('off')
```

Feature Extraction

```
def extract_features(images):
    features = []
    for image in tqdm(images):
        img = load_img(image, grayscale=True)
        img = img.resize((128, 128), Image.ANTIALIAS)
        img = np.array(img)
        features.append(img)

    features = np.array(features)
    # ignore this step if using RGB
    features = features.reshape(len(features), 128, 128, 1)
    return features
```

```
X = extract_features(df['image'])
```

```
X.shape
```

```
# normalize the images
X = X/255.0
```

```
y_gender = np.array(df['gender'])
y_age = np.array(df['age'])
```

```
input_shape = (128, 128, 1)
```

Model Creation

```
inputs = Input((input_shape))
# convolutional layers
conv_1 = Conv2D(32, kernel_size=(3, 3), activation='relu') (inputs)
maxp_1 = MaxPooling2D(pool_size=(2, 2)) (conv_1)
conv_2 = Conv2D(64, kernel_size=(3, 3), activation='relu') (maxp_1)
maxp_2 = MaxPooling2D(pool_size=(2, 2)) (conv_2)
conv_3 = Conv2D(128, kernel_size=(3, 3), activation='relu') (maxp_2)
maxp_3 = MaxPooling2D(pool_size=(2, 2)) (conv_3)
conv_4 = Conv2D(256, kernel_size=(3, 3), activation='relu') (maxp_3)
maxp_4 = MaxPooling2D(pool_size=(2, 2)) (conv_4)

flatten = Flatten() (maxp_4)

# fully connected layers
dense_1 = Dense(256, activation='relu') (flatten)
dense_2 = Dense(256, activation='relu') (flatten)

dropout_1 = Dropout(0.3) (dense_1)
dropout_2 = Dropout(0.3) (dense_2)

output_1 = Dense(1, activation='sigmoid', name='gender_out') (dropout_1)
output_2 = Dense(1, activation='relu', name='age_out') (dropout_2)

model = Model(inputs=[inputs], outputs=[output_1, output_2])

model.compile(loss=['binary_crossentropy', 'mae'], optimizer='adam', metrics=['accuracy'])
```

```
# plot the model
from tensorflow.keras.utils import plot_model
plot_model(model)
```

```
# train model
history = model.fit(x=X, y=[y_gender, y_age], batch_size=32, epochs=30, validation_split=0.2)
```

Plot the Results

```
# plot results for gender
acc = history.history['gender_out_accuracy']
val_acc = history.history['val_gender_out_accuracy']
epochs = range(len(acc))

plt.plot(epochs, acc, 'b', label='Training Accuracy')
plt.plot(epochs, val_acc, 'r', label='Validation Accuracy')
plt.title('Accuracy Graph')
plt.legend()
plt.figure()

loss = history.history['gender_out_loss']
val_loss = history.history['val_gender_out_loss']

plt.plot(epochs, loss, 'b', label='Training Loss')
plt.plot(epochs, val_loss, 'r', label='Validation Loss')
plt.title('Loss Graph')
plt.legend()
plt.show()
```

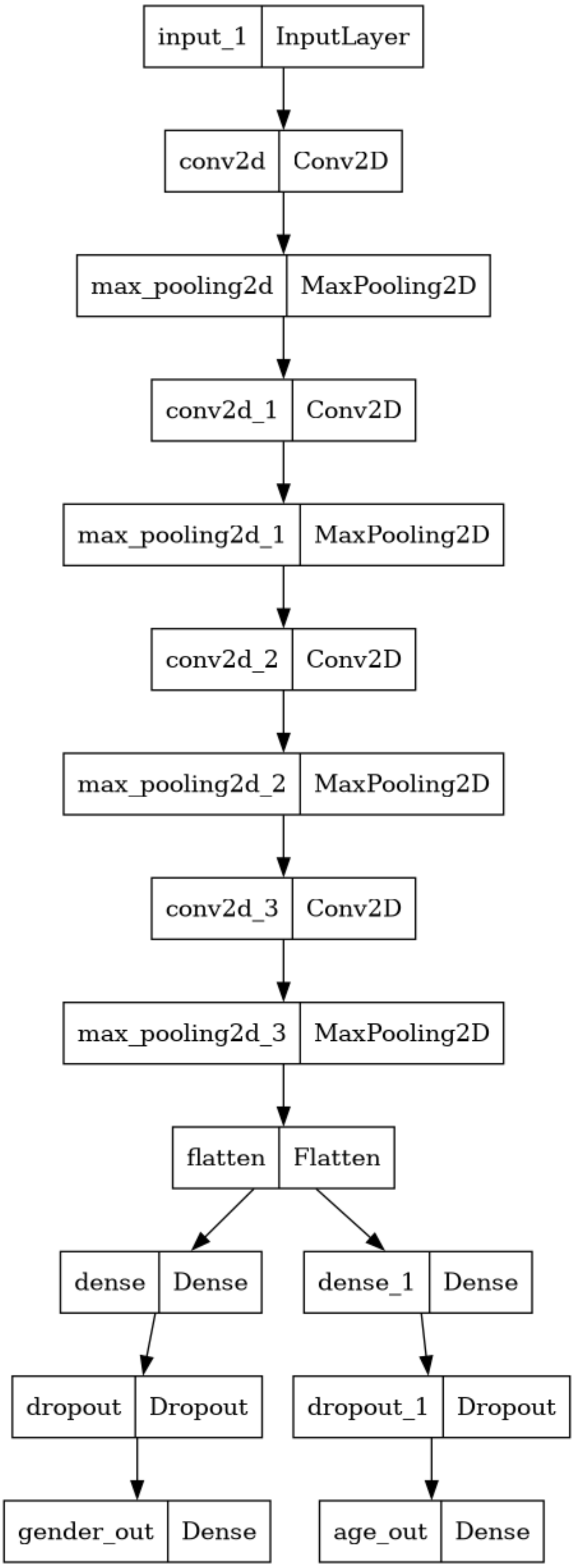
```
# plot results for age
loss = history.history['age_out_loss']
val_loss = history.history['val_age_out_loss']
epochs = range(len(loss))

plt.plot(epochs, loss, 'b', label='Training Loss')
plt.plot(epochs, val_loss, 'r', label='Validation Loss')
plt.title('Age Loss Graph')
plt.legend()
plt.show()

# plot results for age accuracy
acc = history.history['age_out_accuracy']
val_acc = history.history['val_age_out_accuracy']
epochs = range(len(acc))

plt.plot(epochs, acc, 'b', label='Training Accuracy')
plt.plot(epochs, val_acc, 'r', label='Validation Accuracy')
plt.title('Age Accuracy Graph')
plt.legend()
plt.show()
```

Prediction with Test Data

```
image_index = 100
print("Original Gender:", gender_dict[y_gender[image_index]], "Original Age:", y_age[image_index])
# predict from model
pred = model.predict(X[image_index].reshape(1, 128, 128, 1))
pred_gender = gender_dict[round(pred[0][0][0])]
pred_age = round(pred[1][0][0])
print("Predicted Gender:", pred_gender, "Predicted Age:", pred_age)
plt.axis('off')
plt.imshow(X[image_index].reshape(128, 128), cmap='gray');
```

```
Original Gender: Female Original Age: 3
Predicted Gender: Female Predicted Age: 1
```

MODEL USED BY MY CNN



Appendix C- Data table with original dataset.

Appendix C1

Test No.	Loss:	Accuracy:	Age Precision:	Gender Precision	Recall:
1	3.8022	1.00	Predicted Age: 3 Actual Age: 3	Predicted Gender:0 Actual Gender:1	1.0
2	3.8362	0.94	Predicted Age: 12 Actual Age: 15	Predicted Gender:0 Actual Gender:0	0.97
3	3.8110	0.95	Predicted Age: 26 Actual Age: 28	Predicted Gender:0 Actual Gender:0	0.98
4	3.7963	0.93	Predicted Age: 45 Actual Age: 40	Predicted Gender: 1 Actual Gender: 1	0.96
5	3.8722	0.98	Predicted Age: 62 Actual Age: 61	Predicted Gender: 1 Actual Gender: 1	0.99
6	3.7962	1.00	Predicted Age: 78 Actual Age: 78	Predicted Gender: 0 Actual Gender: 0	1.0
7	3.8090	0.95	Predicted Age: 92 Actual Age: 97	Predicted Gender: 1 Actual Gender: 1	0.94

Appendix C2

Test No.	Loss:	Accuracy:	Age Precision:	Gender Precision	Recall:
1	3.8028	0.96	Predicted Age: 2 Actual Age: 3	Predicted Gender:0 Actual Gender:0	1.0
2	3.8363	0.96	Predicted Age: 16 Actual Age: 15	Predicted Gender:0 Actual Gender:1	0.99
3	3.8110	0.96	Predicted Age: 27 Actual Age: 28	Predicted Gender:0 Actual Gender:1	0.98
4	3.7968	0.98	Predicted Age: 38 Actual Age: 40	Predicted Gender: 1 Actual Gender: 1	0.98
5	3.8720	0.93	Predicted Age: 66 Actual Age: 61	Predicted Gender: 1 Actual Gender: 1	0.93
6	3.7962	1.00	Predicted Age: 78 Actual Age: 78	Predicted Gender: 0 Actual Gender: 0	1.0
7	3.8095	0.97	Predicted Age: 92 Actual Age: 88	Predicted Gender: 1 Actual Gender: 1	0.97

Appendix C3

Test No.	Loss:	Accuracy:	Age Precision:	Gender Precision	Recall:
1	3.8032	0.97	Predicted Age: 4 Actual Age: 3	Predicted Gender:0 Actual Gender:0	1.0
2	3.8368	0.98	Predicted Age: 15 Actual Age: 15	Predicted Gender:0 Actual Gender:0	0.99
3	3.8116	0.94	Predicted Age: 27 Actual Age: 28	Predicted Gender:0 Actual Gender:1	0.99
4	3.7973	0.96	Predicted Age: 42 Actual Age: 40	Predicted Gender: 1 Actual Gender: 1	0.98
5	3.8724	0.96	Predicted Age: 65 Actual Age: 61	Predicted Gender: 1 Actual Gender: 1	0.99
6	3.7962	1.00	Predicted Age: 78 Actual Age: 78	Predicted Gender: 0 Actual Gender: 0	1.0
7	3.8100	0.97	Predicted Age: 92 Actual Age: 90	Predicted Gender: 1 Actual Gender: 1	0.98

Appendix D- Data table with reduced dataset.

Appendix D1

Test No.	Loss:	Accuracy:	Age Precision:	Gender Precision	Recall:
1	4.4026	0.75	Predicted Age: 2 Actual Age: 3	Predicted Gender:0 Actual Gender:1	0.77
2	4.7380	0.73	Predicted Age: 15 Actual Age: 15	Predicted Gender:0 Actual Gender:0	0.75
3	4.3933	0.81	Predicted Age: 30 Actual Age: 28	Predicted Gender:0 Actual Gender:0	0.82
4	4.7900	0.77	Predicted Age: 41 Actual Age: 40	Predicted Gender: 1 Actual Gender: 1	0.79
5	4.5766	0.78	Predicted Age: 53 Actual Age: 61	Predicted Gender: 1 Actual Gender: 1	0.80
6	4.9964	0.76	Predicted Age: 73 Actual Age: 78	Predicted Gender: 0 Actual Gender: 0	0.77
7	4.4480	0.82	Predicted Age: 94 Actual Age: 93	Predicted Gender: 1 Actual Gender: 1	0.84

Appendix D2

Test No.	Loss:	Accuracy:	Age Precision:	Gender Precision	Recall:
1	4.4066	0.76	Predicted Age: 5 Actual Age: 3	Predicted Gender:0 Actual Gender:1	0.77
2	4.7399	0.75	Predicted Age: 16 Actual Age: 15	Predicted Gender:1 Actual Gender:0	0.76
3	4.2190	0.80	Predicted Age: 35 Actual Age: 28	Predicted Gender:1 Actual Gender:0	0.81
4	4.7800	0.73	Predicted Age: 41 Actual Age: 40	Predicted Gender: 1 Actual Gender: 1	0.77
5	4.5788	0.77	Predicted Age: 54 Actual Age: 61	Predicted Gender: 1 Actual Gender: 1	0.72
6	4.9885	0.71	Predicted Age: 69 Actual Age: 78	Predicted Gender: 0 Actual Gender: 0	0.72
7	4.4489	0.79	Predicted Age: 98 Actual Age: 93	Predicted Gender: 1 Actual Gender: 1	0.80

Appendix D3

Test No.	Loss:	Accuracy:	Age Precision:	Gender Precision	Recall:
1	4.4020	0.82	Predicted Age: 4 Actual Age: 3	Predicted Gender:1 Actual Gender:1	0.84
2	4.9397	0.77	Predicted Age: 11 Actual Age: 15	Predicted Gender:1 Actual Gender:0	0.79
3	4.2290	0.77	Predicted Age: 35 Actual Age: 28	Predicted Gender:1 Actual Gender:0	0.76
4	4.7960	0.72	Predicted Age: 44 Actual Age: 40	Predicted Gender: 1 Actual Gender: 1	0.75
5	4.5780	0.74	Predicted Age: 59 Actual Age: 61	Predicted Gender: 1 Actual Gender: 1	0.77
6	4.9999	0.71	Predicted Age: 77 Actual Age: 78	Predicted Gender: 0 Actual Gender: 0	0.74
7	4.4499	0.74	Predicted Age: 98 Actual Age: 93	Predicted Gender: 1 Actual Gender: 1	0.76