Detection of Gender, Age and Ethnicity Through Facial Images

Introduction

Facial recognition is a technology that is crucial for many industries, including security, marketing, healthcare, and entertainment, as it enables efficient identification, verification, and tracking of humans by machines (Li et al., 2022). It has numerous applications such as surveillance, access control, personalised marketing experiences, disease diagnosis, patient monitoring, social robots, and virtual assistants (Srivastava et al., 2022; Katsanis et al., 2021; Baltanas et al., 2021).

Facial recognition technology can detect demographic factors such as age and gender. Accurate detection of these factors is important because it helps to avoid bias and discrimination in AI systems. AI is capable of increasing productivity, simplifying processes, and reducing costs. Nonetheless, because of its inherent biases and errors, it can institutionalise discrimination and so perpetuate structural inequalities (Packin and Lev-Aretz, 2018). While many agree that AI is the future, it is essential to ensure that it is not a future that serves a few. The main research question this paper covers is: “Do factors such as ethnicity and skin colour affect the accuracy of a model’s age and gender prediction?” It is expected that the model will perform accurately across the board, especially because we are building the models from scratch.

Literature Review

With significant accuracy levels on numerous datasets, convolutional neural networks (CNNs) have emerged as the most advanced method for the problem. However, concerns have been raised about the potential biases in these models towards certain ethnicities and skin tones. In this literature review, we will discuss previous works on the impact of ethnicity and skin colour on CNN models for gender and age estimation using facial images.

Studies have examined ethnicity and skin colour's effect on CNN models for gender and age estimation. Mohamad et al. (2022) explored skin colour's impact on gender classification using a CNN model trained on the CelebA dataset. The MobileNet model was used in another study to classify face images based on skin colour, with two transfer learning methods applied. These methods achieved competitive classification accuracy with proper training strategies.

Using the UTKFace dataset, Srivastava et al. (2022) examined the effect of ethnicity on CNN models for gender classification. They observed that white people had higher accuracy whereas East Asian and Indian people had poorer accuracy. The impact of race on accuracy was significant, with lighter-skinned people scoring better. They observed that bias may result from facial features and expressions.This highlights the need for more representative datasets.

Rothe et al. (2015) investigated how ethnicity and gender affect the accuracy of CNN models for age estimation using the IMDB-WIKI dataset. They found that the model had higher accuracy for some ethnic groups, such as White and Hispanic individuals, than others, such as Asian and Black individuals. Gender also had an impact, with females estimated younger than males, and Asians younger than other ethnicities. The study highlights the need to consider these factors when developing CNN models for age estimation.

Using the UTKFace and APPA-REAL datasets, Puc et al. (2020) looked at how well CNN-based age estimation models performed across various gender and racial subgroups. They used WideResNet and FaceNet, two commercially available age estimation algorithms, to classify face pictures according to race, gender, and combinations of race and gender. The study discovered considerable performance gaps between the groups, with men consistently outperforming women in terms of age estimation precision. However, in various test datasets, race had variable implications on the models.

Facial recognition technology has been found to exhibit bias towards certain demographics, particularly people of colour (Klare et al., 2012). However, most studies have evaluated the accuracy of facial recognition systems rather than their performance for gender and age estimation, and datasets used have lacked diversity in ethnicity and skin colour. This may not accurately capture the extent of biases in CNN models for gender and age estimation. Additionally, some studies have used datasets that may not represent the population's diversity, leading to potential biases. This comparative study aims to determine which model performs better when applied to images with darker-skinned people.

Objectives

We investigate the impact of ethnicity and skin colour on CNN models for gender and age estimation from facial images. Prior research has indicated the potential for bias in facial recognition technology towards people of colour (Buolamwini and Gebru, 2018). Thus, we will assess the models' accuracy across different ethnic groups and skin colours using two approaches. The first involves constructing and training three CNN models on the UTKFace dataset, while the second uses transfer learning from the VGG16 model. Our study will contribute to understanding biases in CNN models and inform the development of more accurate and unbiased models for gender and age estimation in facial images.

Methodology

Convolutional Neural Network

Convolutional neural networks (CNNs) are a category of neural networks that are especially effective at tasks involving object and image detection. The structure of the human visual cortex, which is in charge of processing visual data, serves as an inspiration for CNNs. Information moves through various layers in a CNN, each serving a distinct function. By applying a series of convolved filters, convolutional layers identify features in the input image. The spatial dimensionality of the convolutional layer output is decreased via pooling layers. Fully linked layers, which come last, accomplish classification by mapping the features that were discovered by the prior layers to the classes of output.

Convolutional Neural Networks (CNNs) are widely used in computer vision applications for image classification and object detection. The key building block of a CNN is the convolutional layer, which applies filters to the input image to extract important features. After the convolutional layer, a non-linear activation function such as Rectified Linear Unit (ReLU) is used to introduce non-linearity. The pooling layer follows the convolutional layer to reduce the spatial dimensions while retaining essential features. This helps to reduce the computational complexity and overfitting of the model. The fully connected layer is responsible for classifying the input image into different categories by applying a set of weights that produce a probability distribution over the classes. Batch Normalisation is a technique used to normalise the inputs of each layer, which improves the network's performance. This technique normalises the activations of each layer to have zero mean and unit variance, leading to faster training and better generalisation.performance of the network.

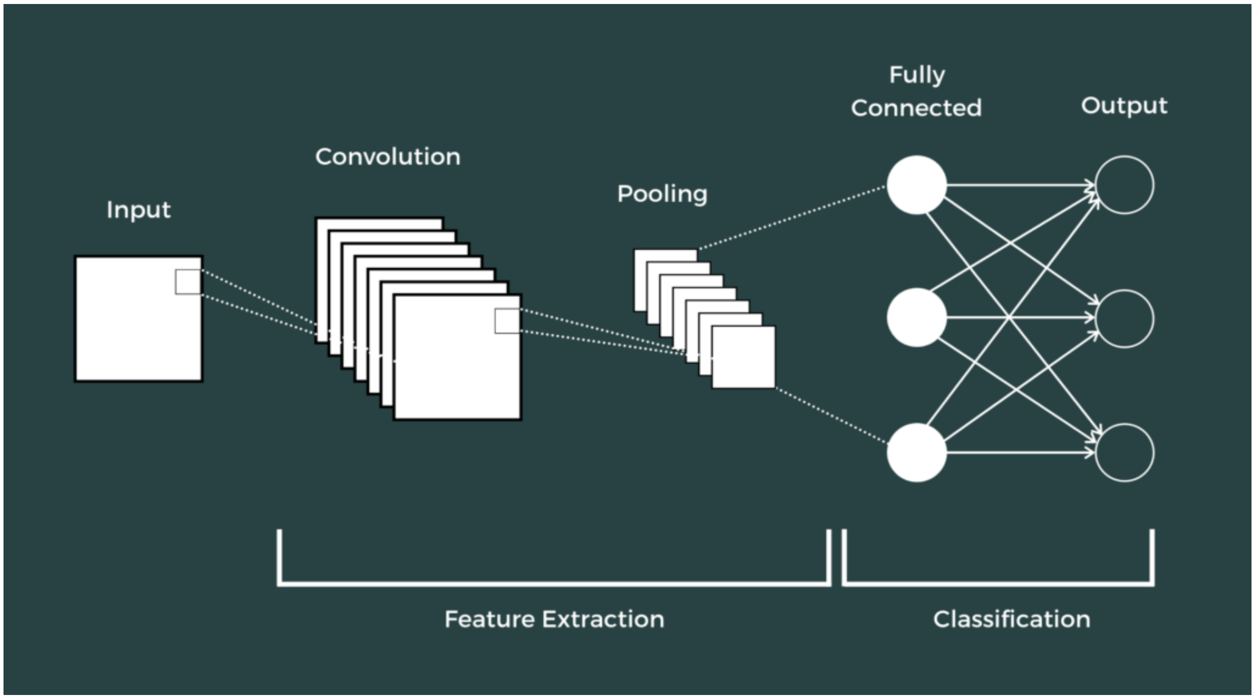


Figure I: Building a Convolutional Neural Network (The Click Reader, n.d)

Transfer Learning Using VGG16

Transfer learning is a machine learning technique that enables a pre-trained model on one task to be adapted for use on another, related task. VGG16 is a widely adopted Convolutional Neural Network (CNN) architecture developed in 2014 by Simonyan and Zisserman at the Visual Geometric Group of Oxford University. This architecture has become a popular choice for transfer learning in various computer vision applications, including image classification, object detection, and segmentation. The VGG16 model comprises 13 convolutional layers and three fully connected layers that are organised into five blocks, with each block containing two or three convolutional layers, followed by a max-pooling layer. The fully connected layers are succeeded by a softmax layer, which outputs the probability distribution over the available classes. The VGG16 network reduces the dimensions of feature maps via small convolutional filters and max-pooling layers, resulting in a high-dimensional representation of the input image that can be classified using a softmax output layer. Batch normalisation is often applied to improve the performance of the network.

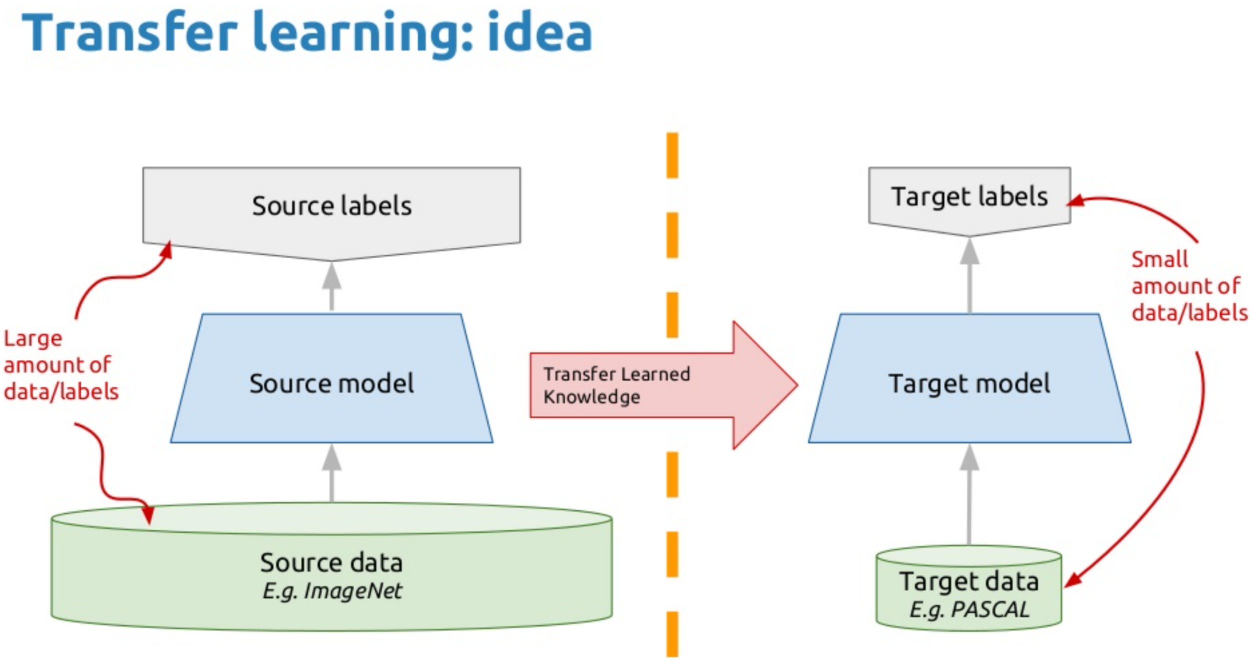


Figure II: Architecture of Transfer Learning (Basu, 2019)

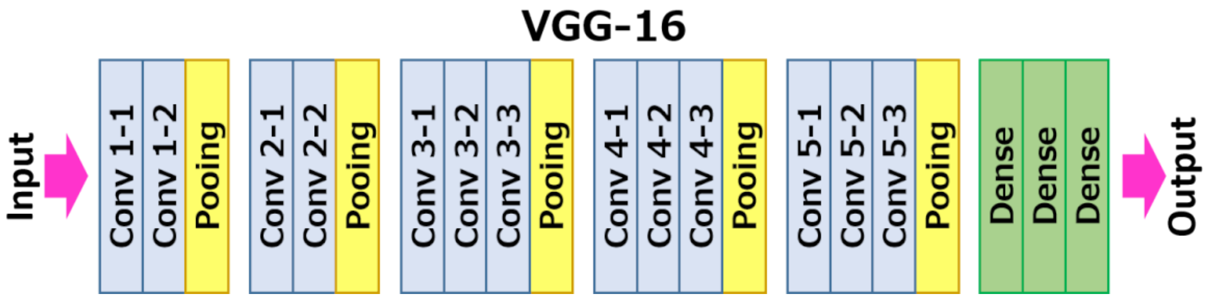


Figure III: Architecture of VGG16 CNN (Popular Networks, 2018)

Experiments

Dataset

This study makes use of the UTKFace dataset, which is obtainable on Kaggle, and is characterised by an extended age range. The dataset contains two versions with varying colour channels. The first version comprises grayscale images, while the latter contains RGB images. The dataset encompasses more than 20,000 facial images annotated with age, gender, and ethnicity. The age feature is an integer ranging from 0 to 116, while the gender attribute is represented by 0 (male) or 1 (female). The ethnicity property is encoded with integers from 0 to 4, denoting White, Black, Asian, Indian, and Others (like Hispanic, Latino, and Middle Eastern).

In preparing the dataset, it was partitioned into 70% for training, 30% for testing, and 10% for validation purposes.

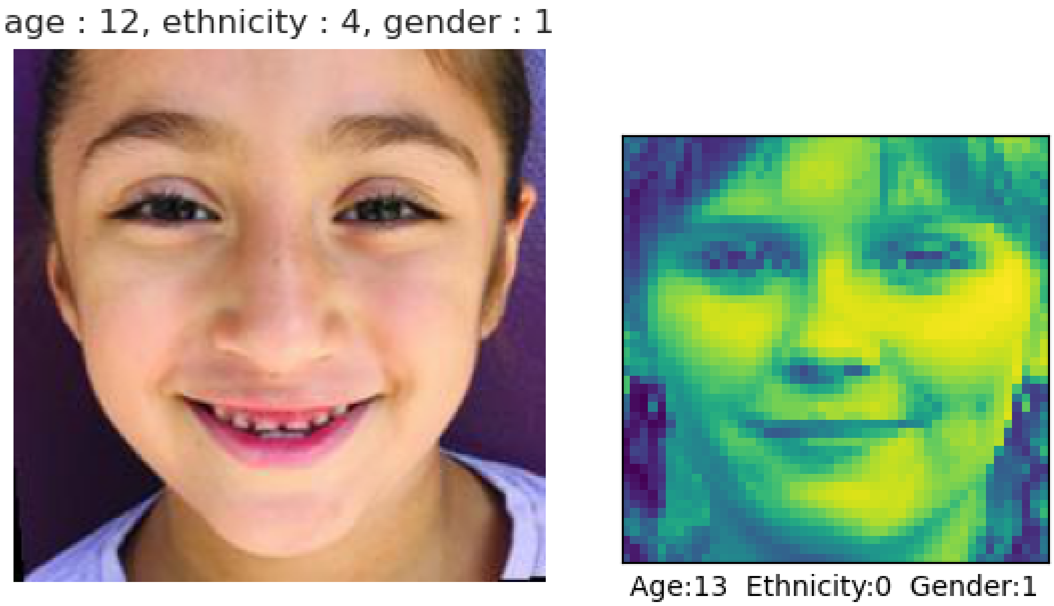

age : 12, ethnicity : 4, gender : 1
Age:13 Ethnicity:0 Gender:1

Figure IV: Sample images from the dataset

Data Pre-Processing

Data preprocessing is a crucial stage when analysing data. This involves cleaning, preprocessing, and normalising the data. This stage includes tasks such as feature extraction, feature scaling, data transformation, data cleaning, data augmentation, and data splitting. The UTKFace dataset was already preprocessed and labelled by age, gender and ethnicity. For the custom approach, the greyscale images of the dataset were used to train the models to expedite convergence through the reduction of channels. The pixels were also normalised using the lambda function. Normalisation transforms the data into a more consistent and comparable range that can help improve the performance of machine learning algorithms or data analysis results. In the transfer approach, the RGB dataset was used because the method requires three channels. Data augmentation was implemented using TensorFlow to create duplicate images in variations to train the dataset better. The first data generator, *train_image_generator,* applies various image augmentation techniques, including rescaling, shearing, zooming, rotation, and shifting to the training images. The second data generator, *test_image_generator*, only rescales the test images. The rescaling in both generators normalises the pixel values of the images to a range between 0 and 1.

Data Exploration

The distributions of the age, gender and ethnicity features were examined and visualised to enhance understanding of the dataset. Furthermore, it was observed that there was an imbalance in the dataset. To solve this issue, the data was oversampled. This will be covered later on in the report.

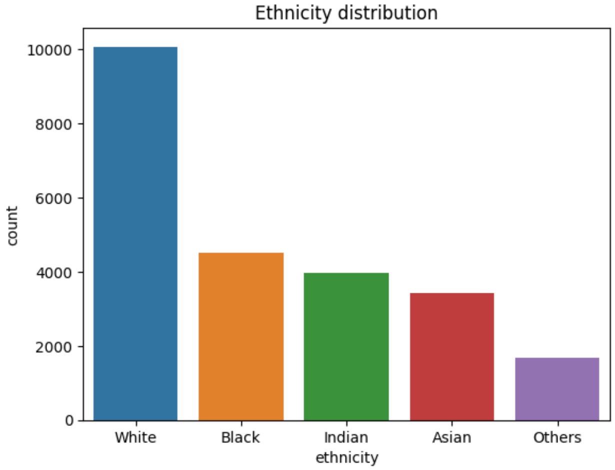


Figure V: Ethnicity distribution of dataset

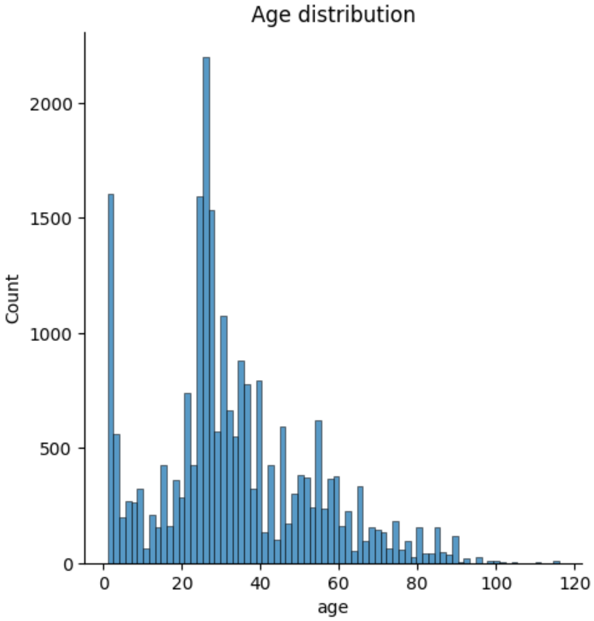


Figure VI: Age distribution of dataset

Model Training

We created a function that takes in the Keras model, train generator, validation generator, number of epochs, and a name as input arguments. It then defines three callbacks for monitoring and saving the best model during training, including:

ReduceLROnPlateau: This reduces the learning rate by a factor of 0.1 when the validation loss does not improve for 6 epochs.
ModelCheckpoint: This saves the weights of the best model based on validation loss during training.
EarlyStopping: this stops the training when validation loss does not improve for 15 epochs and restores the weights of the best model.

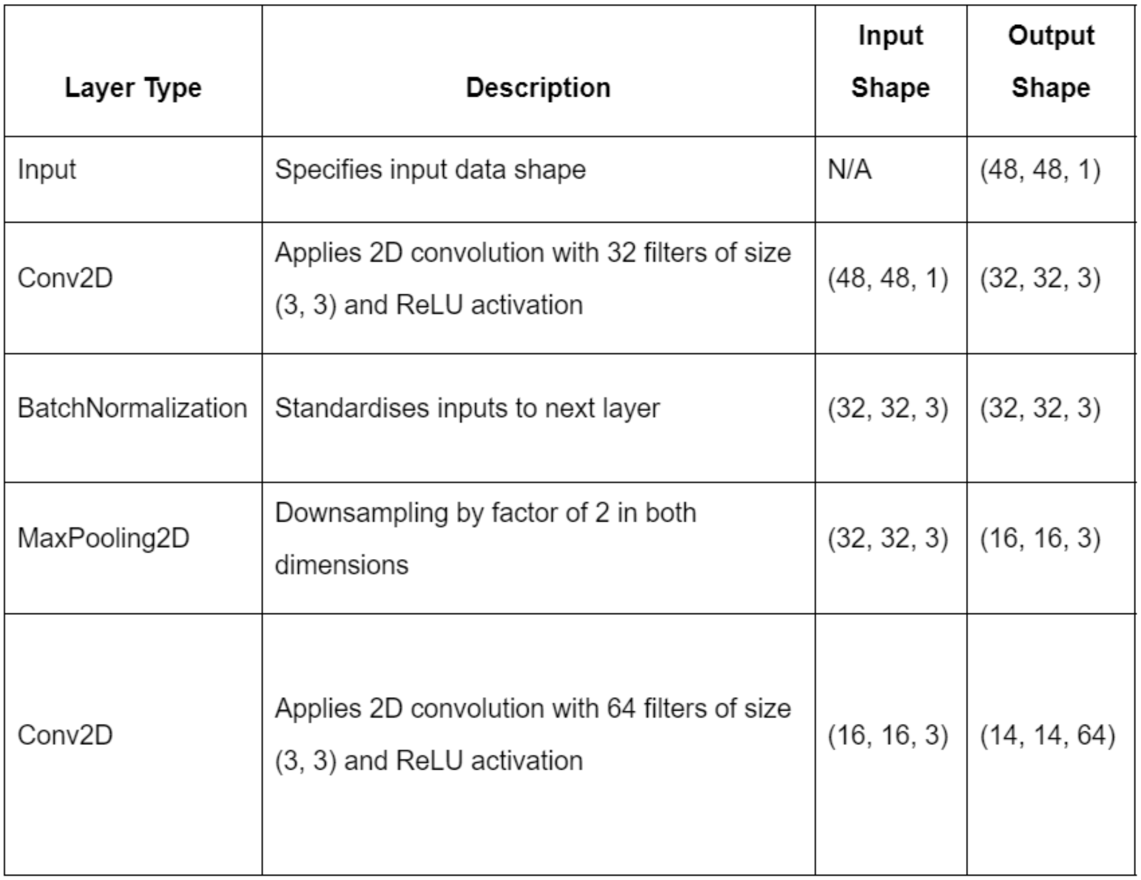
The Custom Approach

Three CNN models were built for each attribute. This separation allows the models to be optimised based on the features they will learn separately from the single dataset. Gender and ethnicity were considered binary and multiclass classification problems. While age was considered a regression problem because a numeric value was being created. The same data was trained to predict three different targets. Each feature was isolated to make predictions for each target.

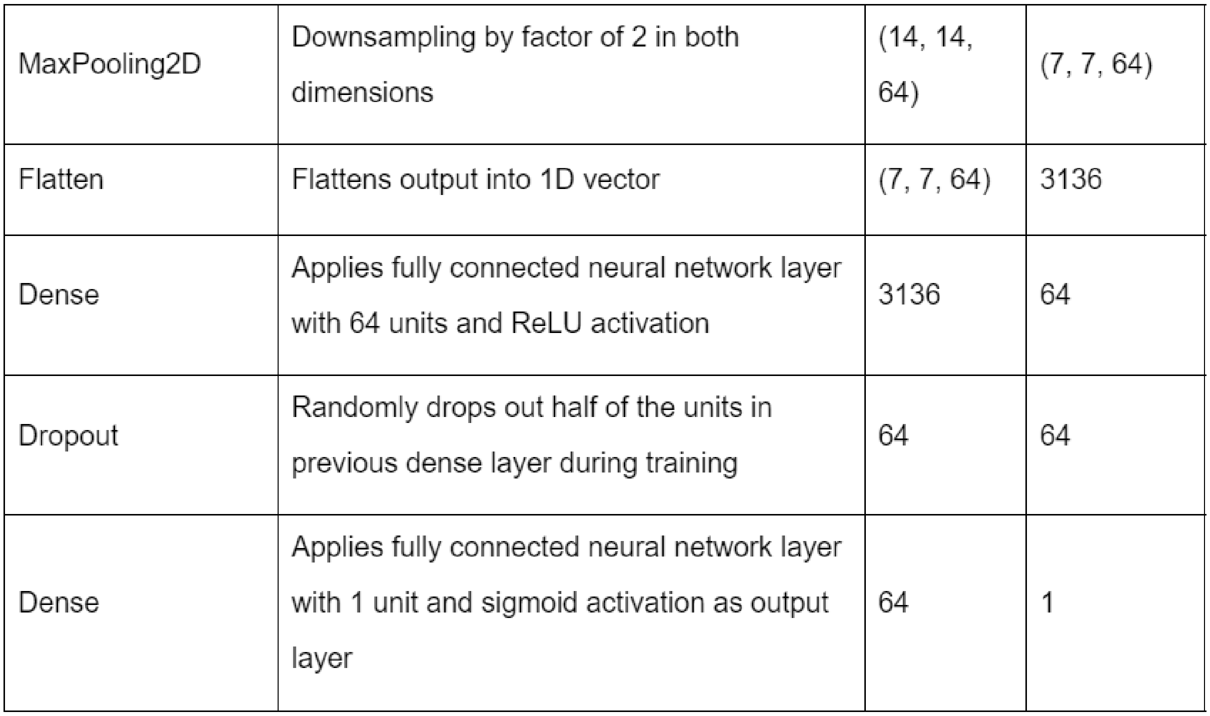
i. Gender

The table below summarises the different layers used in the neural network model architecture for predicting the gender of an individual from images. Each layer is described by its type, its function, and the input and output shapes of the layer. The input shape represents the dimensions of the data input to the layer, and the output shape represents the dimensions of the data output from the layer.

Table I: Summary of the layers in the custom CNN gender model



Layer Type	Description	Input Shape	Output Shape
Input	Specifies input data shape	N/A	(48, 48, 1)
Conv2D	Applies 2D convolution with 32 filters of size (3, 3) and ReLU activation	(48, 48, 1)	(32, 32, 3)
BatchNormalization	Standardises inputs to next layer	(32, 32, 3)	(32, 32, 3)
MaxPooling2D	Downsampling by factor of 2 in both dimensions	(32, 32, 3)	(16, 16, 3)
Conv2D	Applies 2D convolution with 64 filters of size (3, 3) and ReLU activation	(16, 16, 3)	(14, 14, 64)



MaxPooling2D	Downsampling by factor of 2 in both dimensions	(14, 14, 64)	(7, 7, 64)
Flatten	Flattens output into 1D vector	(7, 7, 64)	3136
Dense	Applies fully connected neural network layer with 64 units and ReLU activation	3136	64
Dropout	Randomly drops out half of the units in previous dense layer during training	64	64
Dense	Applies fully connected neural network layer with 1 unit and sigmoid activation as output layer	64	1

The model is then compiled using stochastic gradient descent (SGD) as the optimiser, binary cross-entropy as the loss function, and accuracy as the evaluation metric. Once the model is compiled, it is trained using the *fit()* method by passing the training data and labels.

We then define a custom Keras callback class called '*KerasCallbackCriterion*' that stops the training of the deep learning model if the validation loss drops below a certain threshold. This class inherits from the '*tf.keras.callbacks.Callback*' class, a base class for defining custom callback functions during training. Within the *on_epoch_end()* method, the validation loss is accessed from the logs dictionary using the *get()* method, and if it is less than 0.2700, training is stopped by setting the *stop_training* attribute of the model to True.

ii. Ethnicity

The model is defined using the Keras Sequential API, which allows for the easy creation of a linear stack of layers. It is designed with an input layer that takes in 48x48 grayscale images, followed by several convolutional layers with max pooling to extract and downsample image features. The output layer consists of five neurons representing the possible ethnicity classes. The table serves as a reference for understanding the architecture of the neural network model. It comprises the following architecture:

Table II: Summary of the layers in the custom CNN ethnicity model

Layer Type	Description	Input Shape	Output Shape
Input	Takes tensor of shape (48, 48, 1) representing a 48x48 grayscale image	N/A	(48, 48, 1)
Conv2D	Applies 2D convolution with 32 filters of size (3, 3) and ReLU activation	(48, 48, 1)	(46, 46, 32)
MaxPooling2D	Downsamples output of previous layer by a factor of 2 in both dimensions	(46, 46, 32)	(23, 23, 32)
Conv2D	Applies 2D convolution with 64 filters of size (3, 3) and ReLU activation	(23, 23, 32)	(21, 21, 64)
MaxPooling2D	Downsamples output of previous layer by a factor of 2 in both dimensions	(21, 21, 64)	(10, 10, 64)
Flatten	Converts output tensor of previous layer into a 1D tensor	(10, 10, 64)	6400
Dense	Applies fully connected neural network layer with 64 neurons and ReLU activation	6400	64
Dropout	Randomly drops out half of the units in previous dense layer during training	64	64
Dense	Applies fully connected neural network layer with 5 neurons as output layer	64	5

The *compile()* method is used to configure the model for training. It takes three arguments: optimiser, loss function and metrics. The optimiser is set to '*rmsprop*', a gradient descent optimisation algorithm that uses a moving average of the squared gradients to adjust the

learning rate. It helps to mitigate the vanishing gradient problem and helps to converge faster to the global minimum of the loss function.The loss function is used for multi-class classification problems where integers represent the classes. The model's output is not normalised and represents logits, which are unnormalised log probabilities. This is useful when using the softmax activation function, which is used implicitly in this case, as it can lead to more numerically stable training. The metric used for evaluation during training is accuracy, which measures the proportion of correctly classified samples in the training data.
Similar to the gender model, we define a custom Keras callback class that stops the training of the deep learning model if the validation accuracy reaches 80%.

Oversampling

Oversampling addresses class imbalance by creating synthetic examples of the minority class, improving model performance. SMOTE generates new examples by interpolating between existing minority class instances, providing the model with additional samples to learn from. To balance the dataset, we apply the SMOTE algorithm with a random state of 37 to X and y. The resulting resampled data is split into training and testing sets using a stratify parameter to ensure the proportion of the minority class is kept the same in both sets.

iii. Age

The model's architecture comprises a series of convolutional and pooling layers to extract features from input images. The input layer processes a grayscale image of 48x48, and the layers employ the ReLU activation function to reduce output spatial size and avoid overfitting. The final pooling layer output is flattened and fed through a fully connected dense layer that contains 64 neurons. A Dropout layer is then applied to randomly remove 50% of activations to counter overfitting. The final output layer has a single neuron with a ReLU activation function to predict age in a regression problem.

Table III: Summary of the layers in the custom CNN model

Layer	Number of Filters	Filter Size	Activation Function	Description

Input	-	-	-	Takes 48x48 grayscale image
Convolutional	32	3x3	ReLU	-
Batch Normalisation	-	-	-	Normalises activations of previous layer
MaxPooling2D	-	2x2	-	Reduces spatial size of output
Convolutional	64	3x3	ReLU	-
MaxPooling2D	-	2x2	-	Reduces spatial size of output
Convolutional	128	3x3	ReLU	-
MaxPooling2D	-	2x2	-	Reduces spatial size of output
Flatten	-	-	-	-
Dense	64	-	ReLU	-

Dropout	-	-	-	Randomly drops 50% of activations
Dense	1	-	ReLU	Predicts age as a regression problem

The model utilises the Adam optimiser, which adjusts the learning rate for each parameter during training. The mean squared error (MSE) is adopted as the loss function, measuring the variance between the predicted age and the true age. The mean absolute error (MAE) is the evaluation metric, calculating the average absolute deviation between the predicted age and the true age. Similar to previous models, a callback class is defined to observe the validation loss during training and terminate the training process when the validation loss reaches a threshold of 110.

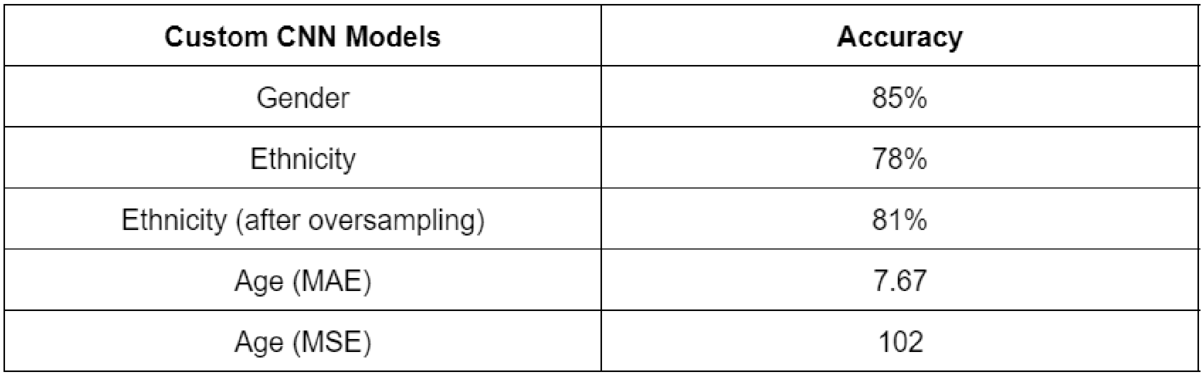
The Transfer Approach

A convolutional neural network (CNN) was constructed using transfer learning from the pretrained VGG16 model in TensorFlow's Keras API. Initially, the VGG16 model was loaded and configured to use the "imagenet" weights, which is a pre-trained model trained on millions of images for image classification. The layers in the VGG16 model were set to be non-trainable to preserve the pre-trained weights. Subsequently, the output of the pre-trained VGG16 model was reshaped to a 6x6 grid of 512 feature maps using the reshape layer and flattened using the flatten layer. Two fully connected layers, each with 256 and 64 neurons, respectively, were added. To prevent overfitting of the model to the training data, a dropout regularisation with 0.2 probability was applied to the first fully connected layer. Finally, a dense layer with two neurons and a softmax activation function was added, which outputs the predicted probabilities of the input image belonging to the two classes. The overall architecture was then combined and defined as a Keras "model" object with the pre-trained VGG16 model as the input layer and the final prediction layer as the output.

Evaluation and Results

To confirm the generalisation of the models to new data, a validation dataset was used to evaluate their performance. The gender and ethnicity models were evaluated based on accuracy scores, while the age models were assessed using mean absolute error and mean squared error, as it was treated as a regression problem. The validation process also included precision, recall, and F1 scores for further verification. Tables were created to summarise the evaluation results.

Table IV: Summary of custom CNN models' evaluation



Custom CNN Models	Accuracy
Gender	85%
Ethnicity	78%
Ethnicity (after oversampling)	81%
Age (MAE)	7.67
Age (MSE)	102

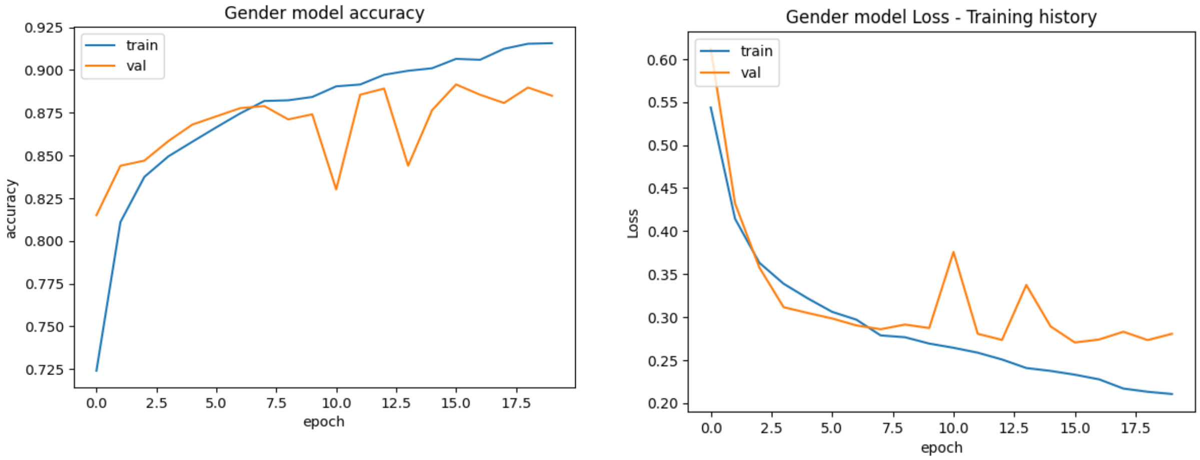


Figure VII: Graph showing custom CNN for gender accuracy and loss

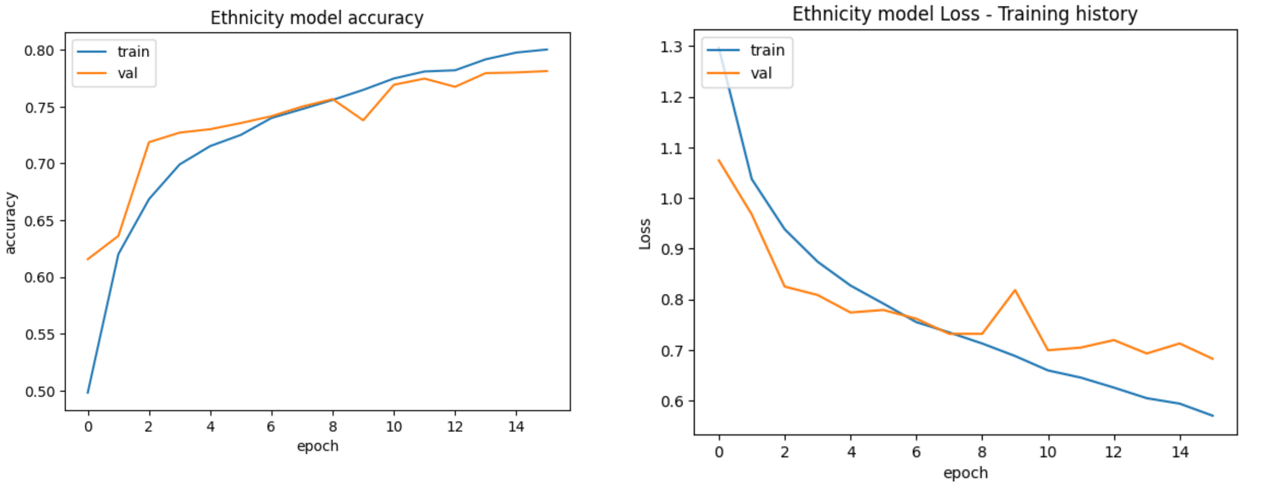


Figure VIII: Graph showing custom CNN for ethnicity accuracy and loss

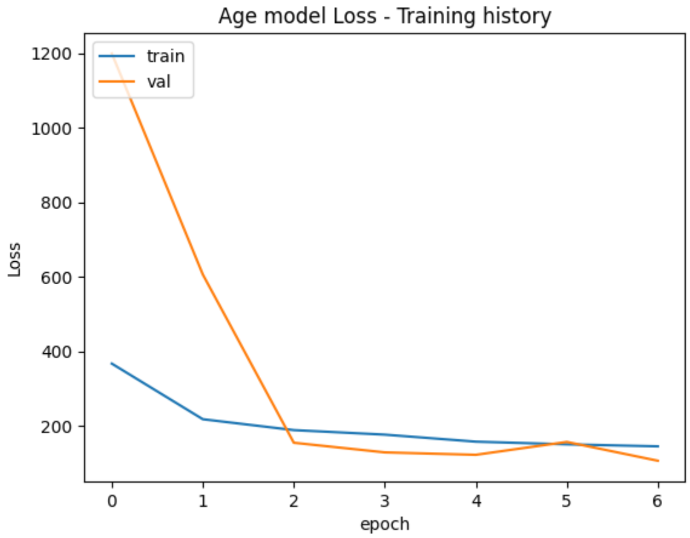


Figure IX: Graph showing custom CNN for age loss

Table V: Summary of transfer learning CNN models' evaluation

Transfer Models	Accuracy
Gender	88%
Age: MSE	141
Ethnicity	73%

The transfer models were tested on image samples of people belonging to different genders and ethnicities. These are the results:

Table VI: Summary of transfer learning CNN models' evaluation on specific sample data

	Male			**Female**		
	White	**Black**	**Asian**	**White**	**Black**	**Asian**
Gender	88%	89%	79%	90%	88%	88%
Age (MSE)	94	84	89	181	108	181

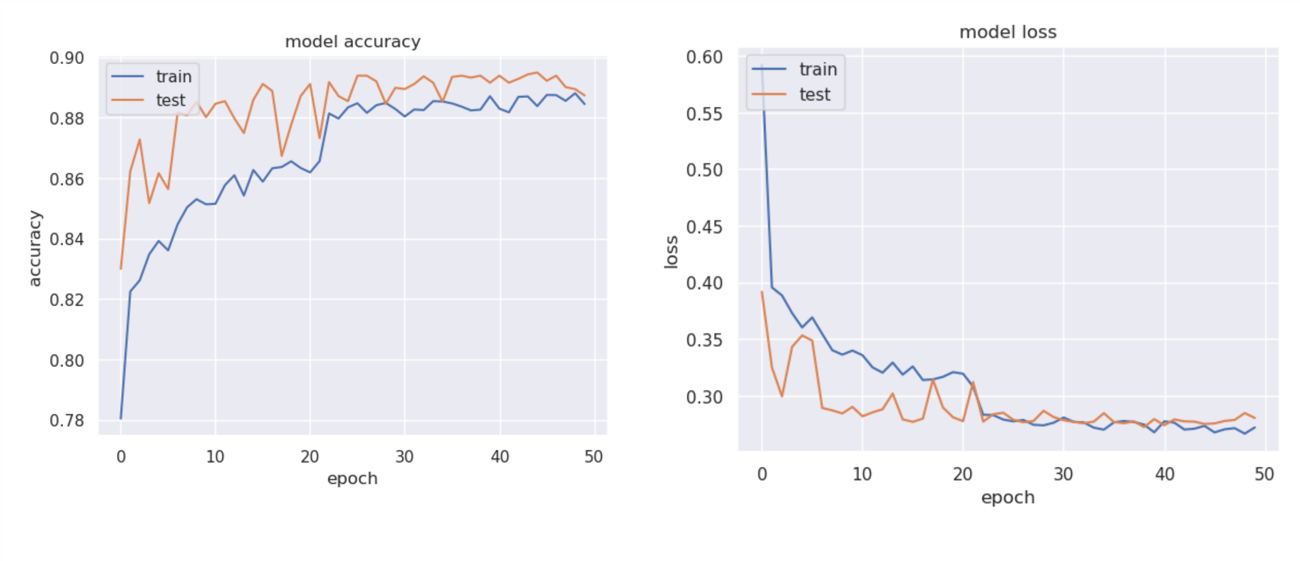
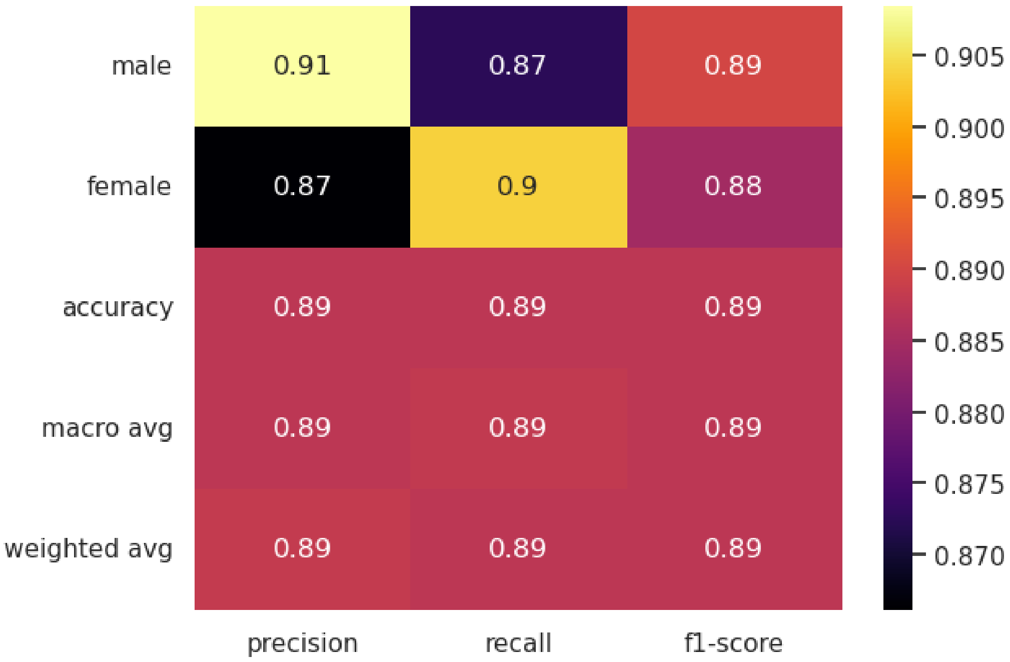


Figure X: Graph showing transfer learning CNN for gender accuracy and loss



	precision	recall	f1-score
male	0.91	0.87	0.89
female	0.87	0.9	0.88
accuracy	0.89	0.89	0.89
macro avg	0.89	0.89	0.89
weighted avg	0.89	0.89	0.89

Figure XI: Classification report evaluating transfer learning CNN for gender model

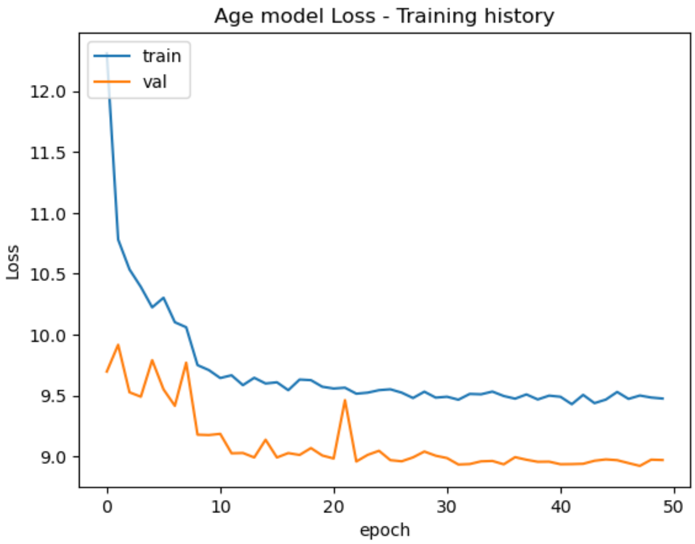


Figure XII: Graph showing transfer learning CNN for age loss

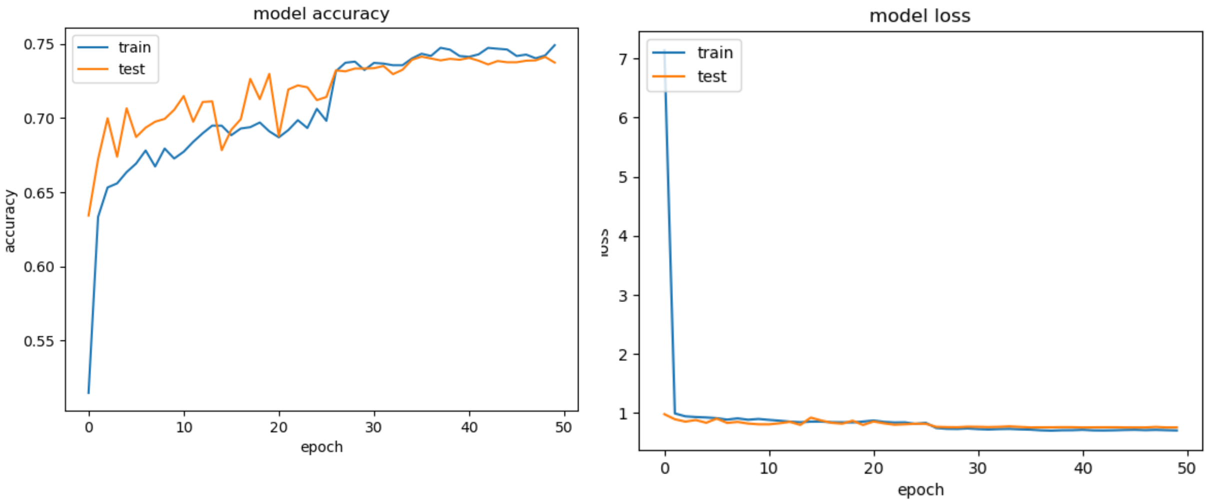


Figure XIII: Graph showing transfer learning CNN for ethnicity accuracy and loss

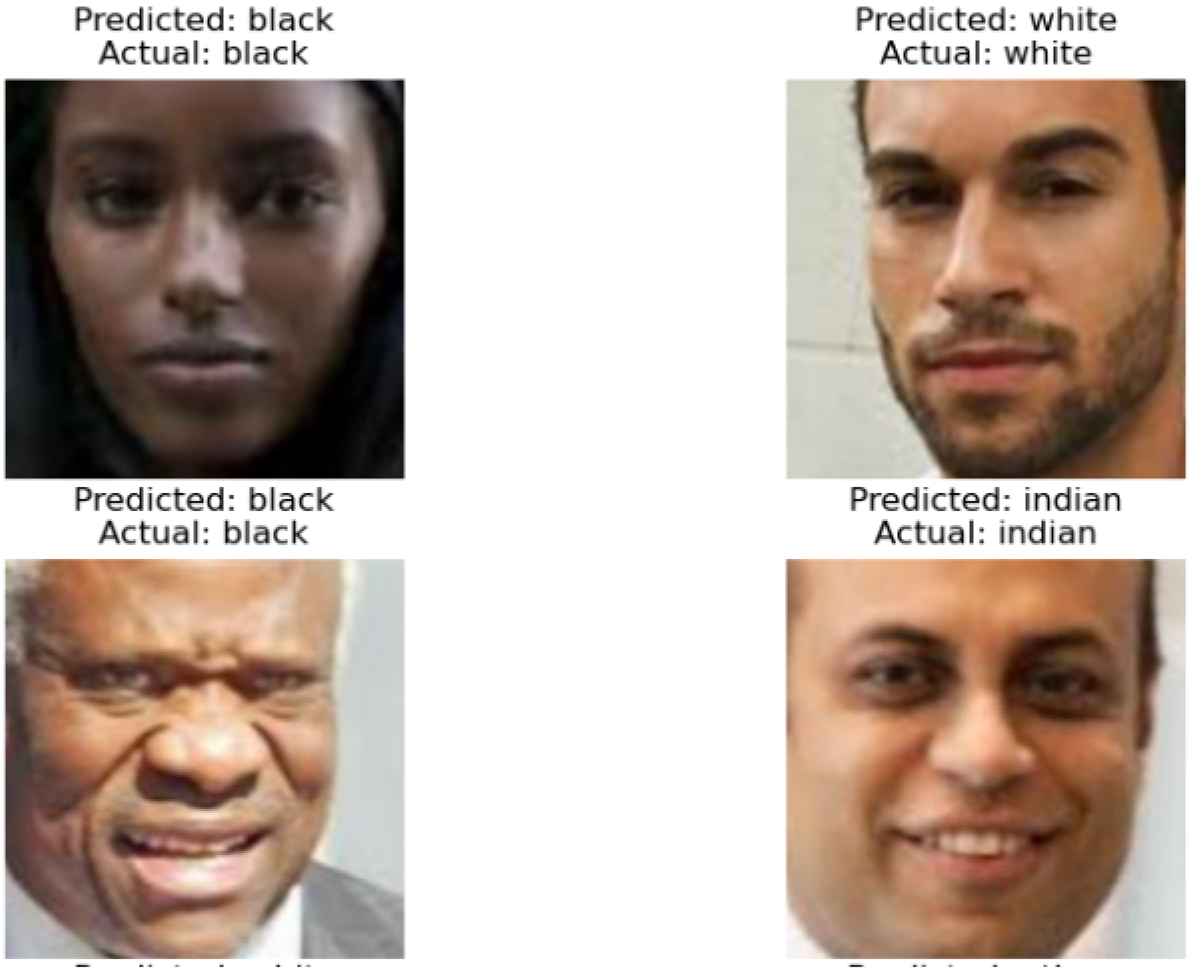


Figure XIV: Sample predictions of transfer learning CNN for ethnicity model

Conclusion and Future Works

The transfer model is effective in accurately predicting gender without regard to race or colour, as demonstrated by the minimal deviation in test accuracies obtained for images of all three races and colours. The model for males obtained accuracies of 88%, 88%, and 79%, while that for females reached 90%, 88%, and 88%. Nonetheless, the presence of poorly annotated data can result in erroneous outcomes, such as mislabeling a white male as Asian and vice versa, which poses a challenge to model training. The task of age prediction for individuals under 10 years old is notably problematic due to the similarity of facial features between boys and girls. Likewise, for those over 90 years old, there may be a lack of certain features, further complicating accurate age prediction. However, some features such as jewellery and facial hair may provide useful distinguishing factors in identifying gender in specific images.

In the ethnicity model, skin tone is used as the primary feature to predict an individual's ethnicity, although this approach may not always be accurate and may produce erroneous outcomes, especially when combined with poorly annotated datasets. The model also relies on detecting pixel values of 0 and 255, which correspond to black and white, respectively. This can lead to confusion in instances where there may be a white person against a dark-coloured background.

In contrast, the age model demonstrates a gender bias, with mean absolute error (MAE) values ranging from 80-95 for males and 108-180 for females. This could be due to more noticeable facial changes such as baldness and moustaches in males.

Efforts were made to assess the models using datasets that contained more individuals with dark skin tones, but the available datasets online were limited to facial images captured in low lighting conditions. Future studies using more complex systems and larger training datasets could potentially improve the accuracy of the models beyond the current reported results. Additionally, the gender model's accuracy scores were relatively consistent in both approaches, and increasing the number of epochs may further verify these findings.

References

Baltanas, S.-F., Ruiz-Sarmiento, J.-R. & Gonzalez-Jimenez, J. (2021) Improving the Head Pose Variation Problem in Face Recognition for Mobile Robots. *Sensors*, 21(2), 659. Available online: https://doi.org/10.3390/s21020659.

Basu, V. (2019) TransferLearning and Unet to segment rocks on moon Kaggle.com. Available online: https://www.kaggle.com/code/basu369victor/transferlearning-and-unet-to-segment-rocks-on-moon [Accessed 20/April/2023].

Building A Convolutional Neural Network - The Click Reader (n.d.) TheClickReader. Available online: https://www.theclickreader.com/building-a-convolutional-neural-network/ [Accessed 24/April/2023].

Buolamwini, J. & Gebru, T. (2018) Gender Shades: Intersectional Accuracy Disparities in Commercial Gender Classification – *MIT Media Lab MIT Media Lab*. Available online: https://www.media.mit.edu/publications/gender-shades-intersectional-accuracy-disparities-in-commercial-gender-classification/ [Accessed 3/March/2023].

Devesh Kumar Srivastava, Gupta, E., Snigdha Shrivastav & Sharma, R. (2023) Detection of Age and Gender from Facial Images Using CNN. *Lecture Notes in Networks and Systems*, 481–491. Available online: https://doi.org/10.1007/978-981-19-6088-8_42.

Haseena, S., Saroja, S., Madavan, R., Karthick, A., Pant, B. & Kifetew, M. (2022) Prediction of the Age and Gender Based on Human Face Images Based on Deep Learning Algorithm. *Computational and Mathematical Methods in Medicine*, 2022, e1413597. Available online: https://doi.org/10.1155/2022/1413597.

Katsanis, S.H., Claes, P., Doerr, M., Cook-Deegan, R., Tenenbaum, J.D., Evans, B.J., Lee, M.K., Anderton, J., Weinberg, S.M. & Wagner, J.K. (2021) A survey of U.S. public perspectives on facial recognition technology and facial imaging data practices in health and research contexts. *ProQuest* [Preprint]. Available online: https://doi.org/10.1371/journal.pone.0257923.

Klare, B.F., Burge, M.J., Klontz, J.C., Vorder Bruegge, R.W. & Jain, A.K. (2012) Face Recognition Performance: Role of Demographic Information. *IEEE Transactions on Information Forensics and Security*, 7(6), 1789–1801. Available online: https://doi.org/10.1109/tifs.2012.2214212.

Levi, G. & Hassner, T. (n.d.) *Age and gender classification using convolutional neural networks*. Available online: https://www.cv-foundation.org/openaccess/content_cvpr_workshops_2015/W08/papers/Levi_Age_and_Gender_2015_CVPR_paper.pdf.

Li, W., Li, J. & Zhou, J. (2022) Deblurring method of face recognition AI technology based on deep learning. *Advances in Multimedia*. Edited by Q. Li, 2022, 1–9. Available online: https://doi.org/10.1155/2022/9146711.

Mohamad, N.M., Mustapha, M.F. & Ab Hamid, S.H. (2022) *Improving gender classification based on skin color using CNN transfer learning IEEE Xplore*. Available online: https://doi.org/10.1109/AiDAS56890.2022.9918791.

Packin, N.G. & Lev-Aretz, Y. (2018) Learning algorithms and discrimination. *Research Handbook on the Law of Artificial Intelligence*, 88–113. Available online: https://www.elgaronline.com/display/edcoll/9781786439048/9781786439048.00014.xml [Accessed 18/April/2023].

Puc, A., Grm, K. & Štruc, V. (2021) *Analysis of Race and Gender Bias in Deep Age Estimation Models*. Available online: https://www.eurasip.org/Proceedings/Eusipco/Eusipco2020/pdfs/0000830.pdf [Accessed 2/April/2023].

Rothe, R., Timofte, R. & Gool, L.V. (2015) *DEX: Deep EXpectation of Apparent Age from a Single Image IEEE Xplore*. Available online: https://doi.org/10.1109/ICCVW.2015.41.

Sheoran, V., Joshi, S. & Bhayani, T.R. (2021) Age and gender prediction using deep CNNs and transfer learning. *Communications in Computer and Information Science* [Preprint]. Available online: https://doi.org/10.1007/978-981-16-1092-9_25.

Simonyan, K. & Zisserman, A. (2014) *Very Deep Convolutional Networks for Large-Scale Image Recognition*. Available online: https://arxiv.org/abs/1409.1556 [Accessed 2/April/2023].

Srivastava, G., and Bag, S. (2022). Modern day marketing concepts based on face recognition and neuro-marketing: A review and future research directions*. Benchmarking: An International Journal, (ahead-of-print).* DOI: https://doi.org/10.1108/BIJ-09-2022-0588

VGG16 - Convolutional Network for Classification and Detection (2018) *Neurohive.io*. Available online: https://neurohive.io/en/popular-networks/vgg16/ [Accessed 20/April/2023].