

AGE AND GENDER CLASSIFICATION APPLICATION

Lee Wang Lin (1171200748), Lee Xi Jie (1161204459), Cheok Jia Heng (1171201466), Oi Zhen Fan (1181300513)

Multimedia University

ABSTRACT

The fundamental goals of pattern recognition and automated classification research are to construct intelligent systems that can effectively learn and identify things. Since the growth of social platforms and social media, automatic age and gender categorization has been important to a growing number of applications. The purpose of this research is to employ deep-convolutional neural networks to create an age and gender classification application. To demonstrate this point, we implement the application using pre-trained deep learning models such as VGG16 and ResNet50. Finally, we proposed its own classification application to determine gender and the age group based on the human face.

1. INTRODUCTION

The human face communicates significant information about individual characteristics via facial features such as identification, gender, expression, estimated age, and so on. This has inspired researchers to obtain information from facial images. Age estimation methods depend largely on characteristics obtained from face images to convey age-related visual information, with the key face features being utilised in electronic customer relationship management, surveillance monitoring, and other applications (Dong, Yuan and Liu, Yinan and Lian, Shiguo, 2016).

The facial features are important in our lives, but the capacity to forecast them accurately and consistently from a face picture is still far from meeting the needs of commercial applications. This is because with numerous internal and external impact elements like one's health status, lifespan, and severe weather circumstances, it is difficult to regulate the progression of human ageing. Other than the weather conditions, facial features are also affected by the environment and genes, which are difficult to model.

Studies by [2], Convolutional Neural Network (CNN) techniques have recently been used in the present age estimate study, resulting in improved age estimation accuracy performance. Deep learning and CNN, as learning-based feature representation techniques, were forced to learn discriminative feature representation directly from raw pixels and acquire linear feature filters required to project face

pictures into another feature space. CNN architectures may extract facial characteristics from a face picture in a unique and robust manner.

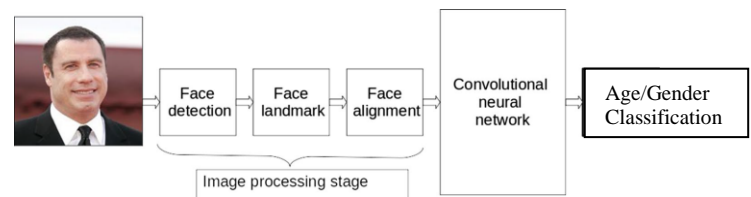


Figure 1.1 CNN Architecture

Figure 1.1 above shows the CNN architecture which the processes include face detection, image processing, features extractions and the classification.

Obtaining the facial features required in an age classification model has proven to be very difficult using the traditional approaches used by the majority of current systems. Many of those techniques are hand-crafted, which requires extensive previous information to design; they cannot be depended on to correctly estimate a person's age. Therefore, the team has decided to introduce a deep learning model for age classification.

In this project, there are two types of characteristics we aim to classify, which are gender and age group. From our human point of view, it is absolute we can know that the left sided are the male; meanwhile the right sided is a female. Other than gender classification, age group detection is also one of the goals for this project and the level of age group is shown in table 2 below;



Male	Female
	

Table 1.1 Gender detection





	Child
	Adolescence
	Adult
	Senior Adult

Table 1.2 Age group detection

One of the motivations in this project is to create an application that can help the human to classify age and gender, with this application, it can help to reduce the manpower of manually detecting the characteristic from human faces. To illustrate on this point, this application is basically using the image-based approaches which extract the features from the images and learn from it.

2. BACKGROUND STUDIES

2.1 DEEP LEARNING MODEL

Convolution Neural Network (CNN) is specialised in processing data that has a 'grid' or 'matrix' like structure such as an image. The main advantages of a CNN is the ability to extract features and patterns from the data without any human supervision, which makes it widely popular in the field of image recognition and computer vision. A CNN typically consists of three layers, a convolutional layer, a pooling layer and a fully connected layer, where the first two layer represent feature extraction and the final layer represents the classification part.

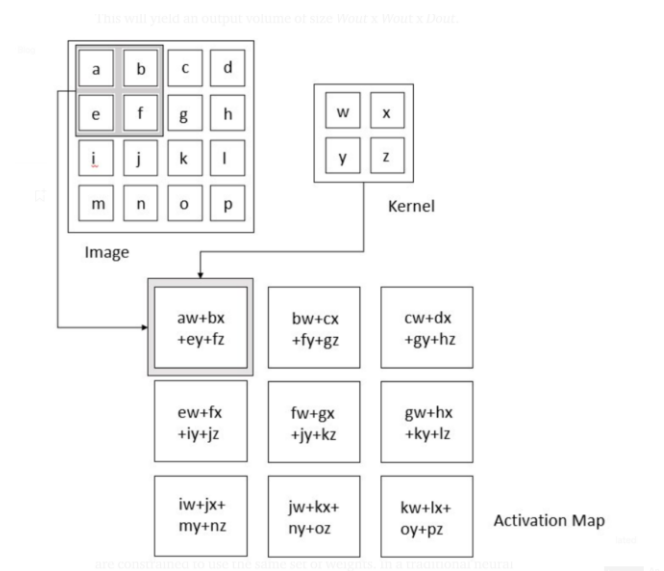


Figure 2.1 Convolutional operation

In the first layer which is the convolutional layer, the layer would receive an input, in this case an image for explanation because this is the most widely used input for CNN. The image input is processed by a kernel function determined by the user to extract the feature in the selected area of matrices, condensing the information (from 4 pixels to 1 in the figure shown above), which reduces the memory requirement for computational processing and improves the statistical efficiency of the model.

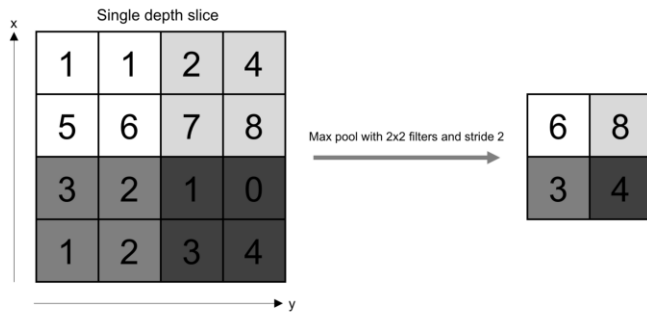


Figure 2.2 Pooling layer

In the second layer which is called the pooling layer, it accepts the processed data after the convolutional layer as input. In the example shown in the figure 2.2 above is a max pooling, which replaces the output of the network at certain positions with the highest value which helps in reducing the spatial size and decreases the computational weight. In ideal conditions, the output after this layer will be able to extract important features and patterns of the image and decrease the noises in the image which will serve better in image recognition or computer vision where features would be captured and verified through the training of these inputs. After the final layer for convolutional and pooling layer, the feature map would be converted and flattened into a 1D array.

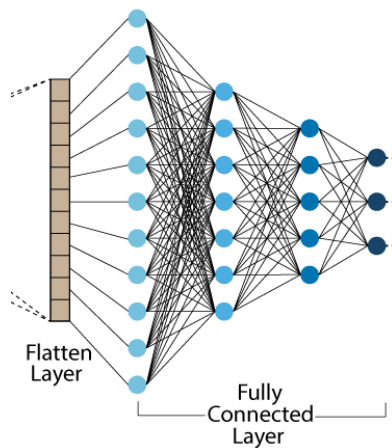


Figure 2.3 Fully connected layer

The fully connected layer is similar to a regular neural network with an output layer which consists of 3 neural units in the example shown in Figure 2.3. The 1D array from the last pooling layer is accepted as input and the classification and machine learning process happens in this layer. Many parameters were given the tools to be customised such as 'Dropout' is a technique that could be used to reduce overfitting of the training data. This layer works the same with a regular neural network model in which the user will be able to modify the output value, activation function, learning rate etc.

2.2 FACE DETECTION

Studies by [4], the authors introduced that use of Haar Cascades to detect the human face, which is a machine learning approach where a cascade function is trained with a set of input data. Figure 2.4 below shows the result from the face detection model.

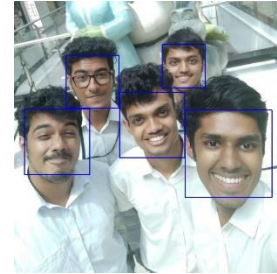


Figure 2.4 Result of face detection [4]

2.3 GENDER DETECTION

According to [3], the researchers claim gender detection is one of the vital processes in face recognition. In this paper, the researchers show the results of classifying the gender using CNN based deep learning architectures using Tensorflow's deep learning framework. The models that are used by the researchers in their experiment are ImageNet and a benchmarking on numerous models such as VGG16, ResNet50 and MobileNet.

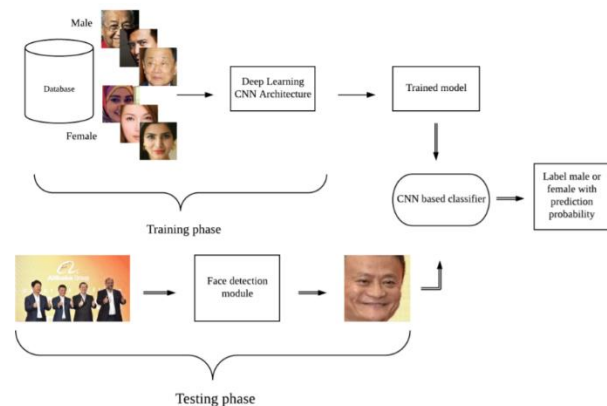


Figure 2.5 Proposed methods by [3]

Models	Input image size	Parameters	No of epochs	Training accuracy	Loss
VGG16	224x224	138,357,544	100	100%	1.7074e-6
Resnet50	224x224	25,636,712	100	99.9%	2.4288e-3
MobileNet	224x224	4,253,864	100	99.8%	7.4571e-3

Table 2.1 Accuracy of the trainset based on different models [3]

Based on the Table 2.1 above, the researchers obtained the 100% accuracy for the VGG16 models with 100 epochs and batch size setting 16. Meanwhile in the Figure 2.6 and Figure 2.7 shows the accuracy and the loss occurred during the training process.

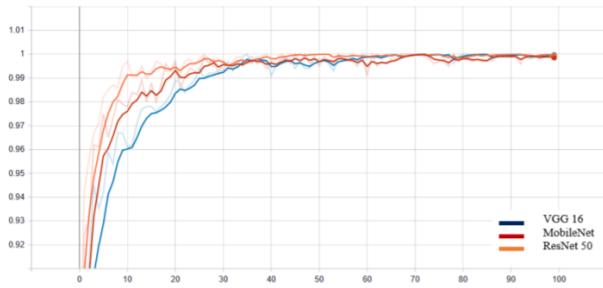


Figure 2.6 Accuracy obtained [3]

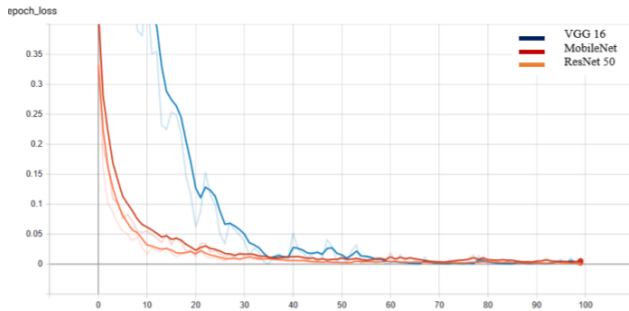


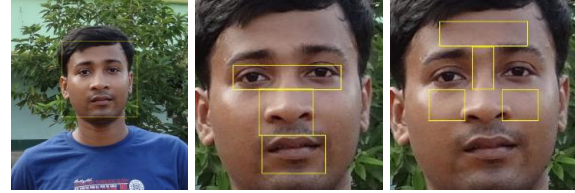
Figure 2.7 Loss occurred [3]

2.4 AGE DETECTION

A clustered using fuzzy c-means method is applied in [1]. This paper implements a method for detecting facial wrinkles, clustering adults into young and old. The researchers use aging function in aging function to perform automatic age estimation. Extract facial information such as eyes, nose and chin area from 3-Dimensions sensors. Then use Bayesian classifiers to classify the photos based on age. In addition, this paper also uses the feature extraction method. Then extract the inverted triangle area composed of two eyes and mouth needed to perform in the K-Means clustering algorithm. Finally Age estimation was done using wrinkle

features in the face image using fuzzy c-means clustering with the accuracy up to 87.5%

Overall in this paper, a preprocessing, eyes, nose, lips, forehead, mid eyebrow and eyelid region is extracted and the samples of the figure are shown below.



In feature extraction, wrinkle information will be extracted from the mid eyebrow and eyelid region by using canny edge detection. In clustering, wrinkle information will be used in fuzzy C-Means clustering algorithms for training purposes. Finally, age can be estimated through membership data generated by fuzzy C-Means clustering algorithms.

No. of Tested Image	No. of Clusters	Marginal Error	No. of Correct Estimation	Accuracy
120	4	± 2 Years	69	57.5 %
		± 3 Years	79	65.8 %
		± 4 Years	90	75 %
		± 5 Years	93	77.5 %
	7	± 2 Years	83	69.2 %
		± 3 Years	93	77.5 %
		± 4 Years	97	80.8 %
		± 5 Years	105	87.5 %
	10	± 2 Years	97	80.8 %
		± 3 Years	101	84.2 %
		± 4 Years	105	87.5 %
		± 5 Years	105	87.5 %

Table 2.2 Result of age estimation [1]

Table 2.2 shows the result of an age estimation. The proposed technique is that based on the wrinkle information from the face and obtain a accuracy of 87.5%.

3.APPROACH

3.1 THE DATASET

The dataset of the gender are retrieved from Kaggle and called as Gender Classification 200k Images | CelebA [5].

From this dataset, we had only taken partial data to conduct this experiment. After cutting down the numbers, there are a total of three folders in this dataset which are called Train, Validate, Test and the subfolders Female and Male. In the training dataset there are 26862 and 27853 numbers of images belonging to male and female respectively; Meanwhile in the validation dataset, there are 6174 and 6089 images belonging to male and female respectively. Other than that, there are 8459 and 10426 numbers of images belong to male and female images in the test dataset

Other than gender dataset, we have obtained the dataset for age at Kaggle also called as UTKFace. In this UTKFace dataset, it contains a total 23700 images for the age from 0 to 116. Therefore we have decided to divide the age from this dataset into categories Child (0-9), Adolescence (10-19), Adult (20-59) and Senior Adult (above 60). After binning the dataset, we realize that the images of senior adults are quite less and present an imbalance scenario. Therefore, we have added the dataset for senior adults from [7]. Due to the lack of dataset obtained in the age group, we have only a training set and validation set. To illustrate on this point, the training sets contain the data obtained from [6] and partially from [7], which are total numbers of 3062 for child, 3272 for adolescence, 2595 for adult and 2690 for senior adult. On the other hand, the testing data for age are obtained from [7], which is a total number of 748 for child, 757 for adolescence, 1398 for adult and 505 for senior adult.

3.2 PYTHON FRAMEWORK AND LIBRARY

The main frameworks and libraries we used in this project are TensorFlow, Keras, os, time, pandas, matplotlib, numpy, sklearn and seaborn. TensorFlow and Keras are the main frameworks/libraries we utilised to create in this project. To illustrate on this point, TensorFlow is a very flexible framework that allows numerous models to be supplied and Keras has a user friendly API where we call and create a neural network model. Based on these two frameworks, we can easily create deep learning models. On the other hand, os, time, pandas, matplotlib, numpy, seaborn are the common libraries that can help us to process and visualize the data.

3.3 DATA PREPROCESSING

As we mention previously, we have to preprocessing the age dataset by binning the age into categories as shown in Figure 3.1 below.

```
# create a list of our conditions
conditions = [
    (df['Yearsold'] <= 9 ),
    (df['Yearsold'] > 9) & (df['Yearsold'] <= 19),
    (df['Yearsold'] > 19) & (df['Yearsold'] <= 59),
    (df['Yearsold'] > 59)
]

# create a list of the values we want to assign for each condition
values = ['Child', 'Adolescence', 'Adult', 'Senior Adult']

# create a new column and use np.select to assign values to it using our lists as arguments
df['Bins'] = np.select(conditions, values)
```

Figure 3.1 Binning age group

```
#Copy the image to the respective file directory
import shutil
dest_path = "C:/Users/Jiek Lee/Desktop/Vip Dataset/New Dataset/Age/face_age"
for i in range(0,len(df)):
    original = image_path+"/"+df["Image"][i]
    print(original)
    dst_dir = dest_path+"/"+df["Bins"][i]
    print(dst_dir)
    if not os.path.exists(dst_dir):
        print(dst_dir)
        os.makedirs(dst_dir)
    shutil.copy(original, dst_dir)
```

Figure 3.2 Copying images

Figure 3.2 shows the sample code of copying the images to the corresponding age group categories.

After binning the age dataset, the main task is to detect the face using the face detection algorithms by [4]. When implementing the face detection algorithm, it will draw a bounding box on the face position. Other than that, we have also made some modifications on the algorithms which allow us to manually crop the region of interest from the images.

Furthermore, data augmentation is also applied in our project where we will flip the image horizontally and use the preprocess input which corresponds to the images. Other than the horizontal flip, while training age we have also applied width and height shift range to the images.

3.4 TRAIN TEST SPLIT

In the field of machine learning or deep learning, train test split plays an important role when fitting the data into the models. To illustrate on this point, train test split usually will split the data into train set and test set. In the training set, it will be used for the model to learn the pattern, on the other hand the test set is used for the evaluation for the model.

TRAIN	TEST
80%	20%
70%	30%
60%	40%

Table 3.1 Train test split ratio

Table 3.1 shows the common ratio for the train test split when creating and fitting the data into the models.

For our project, since the Gener Dataset we obtained from [5] is separated for the train test. So we only apply this technique into the age datasets. We will completely exploit the age dataset and only use a regular train test split with a 70:30 ratio in the experiment.

3.4 FEATURE EXTRACTION

Our strategy for feature extraction will be using transfer learning from the pre-trained models such as VGG16 and ResNet50.

VGG16 is a convolutional neural network model which achieves 92.7% top-5 test accuracy in ImageNet, which is a dataset of over 14 million images belonging to 1000 classes.

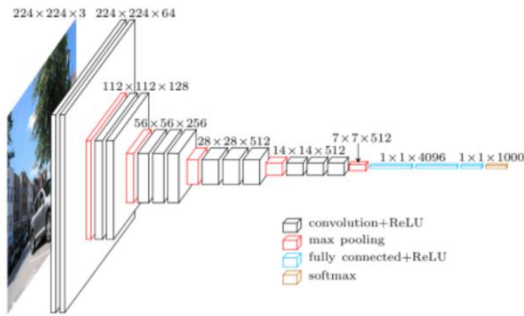


Figure 3.3 VGG16 architecture

Studies by [8], the input to conv1 layer is of fixed size 224x224 RGB image. VGG16 focused on having convolution layers of 3x3 filter with a stride 1 and always used same padding and maxpool layers of 2x2 filter of stride 2. Furthermore, it follow this arrangement of convolution and max pool layrys consistently throughtout the whole architecture. At the end of the output layers, it has two fully connected layers which followed by a softmax for an output.

ResNet50 is a variant of the ResNet model which has 48 Convolution layers along with 1 MaxPool and 1 Average Pool layer. It has 3.8×10^9 Floating points operations. It is a widely used ResNet model and the figure 3.4 shows the ResNet50 architecture in depth [9].

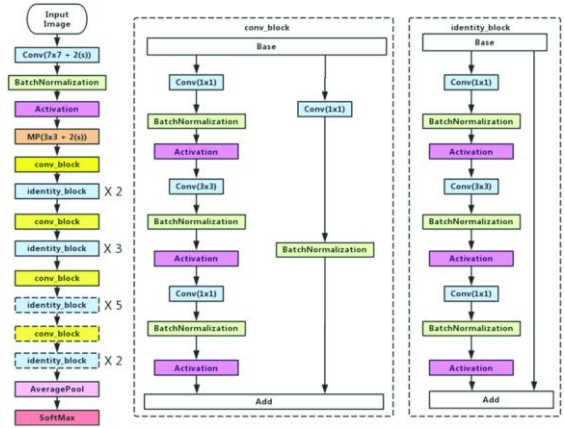


Figure 3.4 ResNet50 architecture

All in one, ResNet50 has 50 convolutional layers as compared to VGG16 with 16 layers. Hence, we can also benchmark the performance in both deep learning models.

3.5 CLASSIFICATION LAYER

The average pooling was chosen because it calculates the average value for patches of a feature map, and uses it to create a downsampled (pooled) feature map.

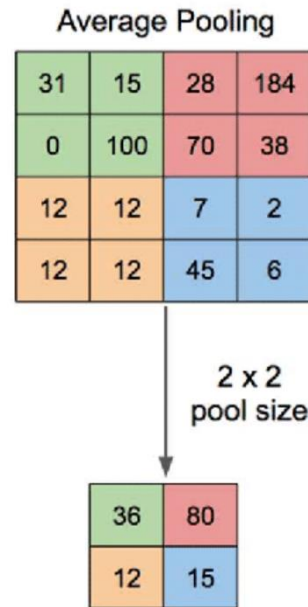


Figure 3.4 Avarage pooling

Meanwhile in the dense layer, we have applied the activation function as Relu which is a learnable parameter when doing classification. It will increase the neurons so that there are

more parameters that the layers can learn. Furthermore, a dropout rate is set to 0.5 to prevent the overfitting or underfitting issues.

The last layer will be a dense layer with softmax and sigmoid activation functions for classifying age group and gender respectively. Since there are two classes in gender, we used sigmoid as the activation functions. Apart from that, age groups consist of multiclass, therefore a multiclass labelling for age groups we apply is softmax. At the end, we can conclude that the last layer will act as a classification layer.

3.6 MODEL COMPILATION

During the compilation, ‘Adam’ is chosen for the optimiser algorithm because of its ability to provide an optimisation algorithm that can handle sparse gradients on noisy problems, and also it is one of the best optimisers that performs well generally across different models.

In the classification model in gender, the loss function is set with ‘Binary Cross entropy’ due to the output layer of two classes, therefore a binary loss function is used to help in calculating the gradient. On the other hand, the loss function in the classification model in age group is changed to ‘Categorical Cross entropy’ due to the output layer of 4 classes, therefore a categorical loss function is used to help in calculating the gradient. Furthermore, the metrics used to evaluate the model is set to ‘Accuracy’ because this is a classification problem and we think that this is the best fit.

Apart from that, an early stopping and checkpoint function is implemented in our model which will terminate the training after the results are not improving or even starts to perform worse, if such scenarios happens, the early stop function will save the latest best performing model and terminate the training process. Meanwhile in the checkpoint, it will save the model with the lowest loss in a specific model file.

4. EXPERIMENT

4.1 IMAGE DATA GENERATOR

In order to train a deep learning model, we have to separate the train and validation set. In the process we made use of the Keras library which can allow us to conduct Train / validation split from a single directory using ImageDataGenerator.

```
train_set = train_generator.flow_from_directory(
    directory = train_dir,
    target_size=(img_height, img_width),
    class_mode="categorical",
    batch_size=batch_size,
    seed = 10,
)

validate_set = test_generator.flow_from_directory(
    directory = validate_dir,
    target_size=(img_height, img_width),
    class_mode="categorical",
    batch_size=batch_size,
    seed = 10,
)

#use the cleaned data to generate train test set
img_height = 224
img_width = 224
batch_size = 16 #Tuneable params

#https://www.tensorflow.org/api_docs/python/tf/keras/preprocessing/image/ImageDataGenerator
#Image Generator
train_generator = ImageDataGenerator(
    preprocessing_function=preprocess_input,
    rotation_range = 30, #int. Degree range for random rotations.
    horizontal_flip = True, #Boolean. Randomly flip inputs horizontally.
    width_shift_range=0.2,
    height_shift_range=0.2,
    fill_mode = "nearest",
)

test_generator = ImageDataGenerator(
    preprocessing_function=preprocess_input
)
```

Figure 4.1 Sample code for generating train and validation set (Gender)

Figure 4.1 shows the sample code for us to split the dataset into train and validate sets for gender directory.

```
#use the cleaned data to generate train test set
img_height = 224
img_width = 224
batch_size = 16 #Tuneable params

#https://www.tensorflow.org/api_docs/python/tf/keras/preprocessing/image/ImageDataGenerator
#Image Generator (Data Augmentation during training)
img_generator = ImageDataGenerator(
    preprocessing_function=preprocess_input,
    rotation_range = 30, #int. Degree range for random rotations.
    horizontal_flip = True, #Boolean. Randomly flip inputs horizontally.
    width_shift_range=0.2,
    height_shift_range=0.2,
    fill_mode = "nearest",
    validation_split=0.3
)

test_generator = ImageDataGenerator(
    rescale = 1/255 #normalizing the images, ease for computation time
)

train_set = img_generator.flow_from_directory(
    subset = "training", #used for training set
    directory = age_dir,
    color_mode = "rgb",
    target_size = (img_height, img_width),
    class_mode = "categorical",
    batch_size = batch_size,
    shuffle = True,
    seed = 10,
)

validation_set = img_generator.flow_from_directory(
    subset = "validation",
    directory = age_dir,
    color_mode = "rgb",
    target_size = (img_height, img_width),
    class_mode = "categorical",
    batch_size = batch_size,
    shuffle = False,
    seed = 10,
)
```

Figure 4.2 Sample code for generating train and validation set (Age)

Figure 4.2 shows the sample code for us to split the dataset into train and validate sets for gender directory. In the age group dataset, we have set the validation to 0.3 which will split the dataset to 70:30 from the directory.

4.2 TRAINING LAYERS

4.2.1 TRAINING LAYERS IN GENDER CLASSIFICATION

1. Dataset obtained from Kaggle [5]
 - a. Consist three folders which are Train, Test and Validate
2. VGG16 Layers : Frozen all layers / ResNet50 : Frozen all layers
3. Optimizer : Adam
4. Learning Rate : 0.001
5. Classification Layers (Sigmoid, with activation Relu)

4.2.2 TRAINING LAYER IN AGE CLASSIFICATION

1. Dataset obtained from Kaggle [6][7]
 - a. Consist the images between age range from 0 to 116.
 - b. Bins the Age into categories (Child, Adolescence, Adult and Senior Adult)
2. VGG16 Layers : Frozen all layers / ResNet50 : Frozen all layers
3. Optimizer : Adam
4. Learning Rate : 0.001
5. Classification Layers (Softmax, with activation Relu)

While creating the deep learning model, we will insert “False” to the “include top” which will help us to remove the fully connected layers designed for identifying the objects that are trained by the respective models. As we noticed if we freeze

all the layers it might face the issues with underfitting, if we do not freeze, it might face the issues of overfitting. Therefore, to be best from our knowledge, we consider the number of images of classes should be more than thousand. Hence, at the end, we decided to freeze all the layers and it will help us to reduce the time taken.

4.3 VGG16 EVALUATION

4.3.1 VGG16 EVALUATION GENDER MODEL

The VGG16 model with total freeze layers consists of 14.978 million parameters. Average Pooling is used to calculate the average value for patches of a feature map, and uses it to create a downsampled (pooled) feature map. The total numbers of trainable parameters and non-trainable parameters are 263682 and 14714688 respectively.

Model: "sequential_1"

Layer (type)	Output Shape	Param #
vgg16 (Functional)	(None, 512)	14714688
dense_2 (Dense)	(None, 512)	262656
dropout_1 (Dropout)	(None, 512)	0
dense_3 (Dense)	(None, 2)	1026
Total params: 14,978,370		
Trainable params: 263,682		
Non-trainable params: 14,714,688		

Figure 4.3 Summary of the VGG16

Before starting to train the model, we apply the default number of epochs 100 and the patience 2. 2 patience will indicate that the validate loss is not improving continuously in two epochs. From the result, the highest accuracy values in the training set is 91.71% with the loss 0.2074. Meanwhile in the validation set, the highest accuracy is 93.05% in epoch 4 but the lowest validate loss is 0.1849 in epoch number 3.

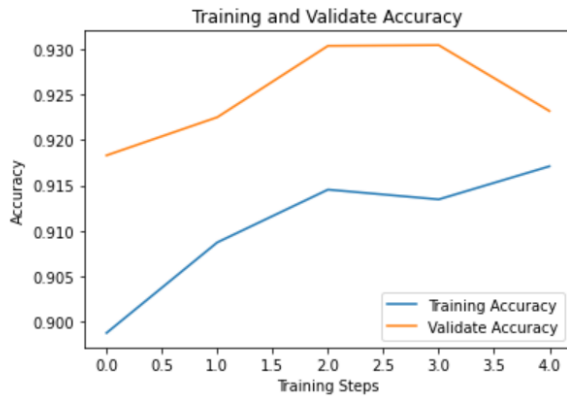


Figure 4.4 Accuracy in training and validation set

For model evaluation, a number of methods were used to test the performance of the model. The first method is a comparison plot of training and validation accuracy across the training iterations. In the figure 4.4 shown above, we could see that validation accuracy, in the neighbourhood of around 0.92 outperforms training accuracy which has a value of around 0.91 on average. This scenario indicates that the model is not overfitting and has a 93% accuracy rate on validation data.

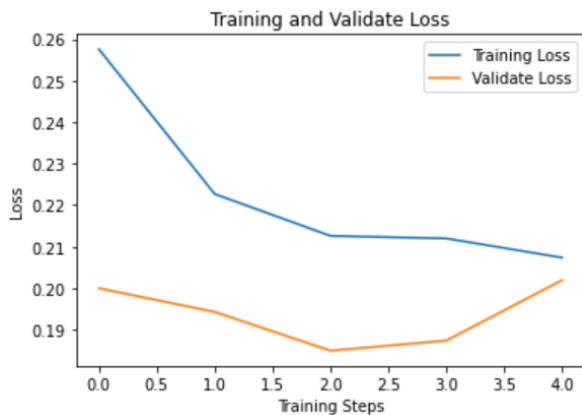


Figure 4.5 Loss in training and validation set

Continuing on, a comparison plot of training and validation loss across the training iterations is used. In the figure below, we could see that it has a roughly similar pattern with the accuracy plot above, in which the validation loss outperforms the training loss. This is an indication that the model is not overfitting and the gap between training and validation loss is relatively small.

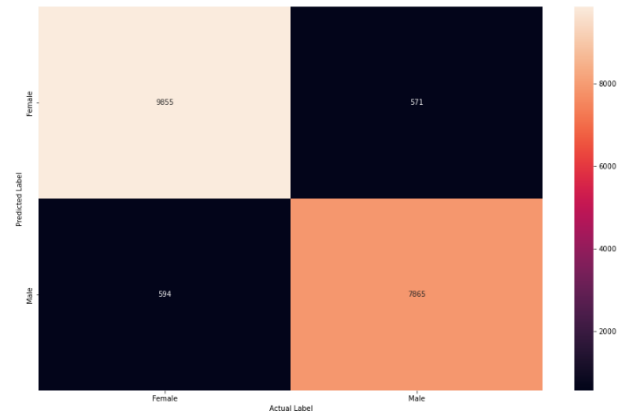


Figure 4.6 Confusion matrix

Then, a confusion matrix plot is used as an evaluation method to check the performance with unseen data of the model. In the figure below, we could see that both the TP to FP ratio and TN to FN ratio are both roughly at around 0.1. The output of this plot tells us that the performance of the model is very similar to the validation accuracy test, which is around 93% accurate.

1181/1181 [=====] - 1664s 1s/step

	precision	recall	f1-score	support
Female	0.94	0.95	0.94	18426
Male	0.93	0.93	0.93	8459
accuracy			0.94	18885
macro avg	0.94	0.94	0.94	18885
weighted avg	0.94	0.94	0.94	18885

Figure 4.7 Classification Report

This classification report shows a summary of the confusion matrix plot. Precision represents the TP rate of the forecast, which is 0.94 for female and 0.93 for male. Recall score is the ratio of TP to (TP + FN), which shows that it has a TP rate of 0.95 for female, and 0.93 for male. F1-score is used as a harmonic mean of precision and recall score which makes it a better measure of the incorrectly classified cases, scoring 0.94 for female and 0.93 for male.

4.3.2 VGG16 EVALUATION AGE MODEL

The VGG16 model with total freeze layers consists of 14.979 million parameters. Average Pooling is used to calculate the average value for patches of a feature map, and uses it to create a downsampled (pooled) feature map. The total numbers of trainable parameters and non-trainable parameters are 264788 and 14714688 respectively.

Model: "sequential"

Layer (type)	Output Shape	Param #
vgg16 (Functional)	(None, 512)	14714688
dense (Dense)	(None, 512)	262656
dropout (Dropout)	(None, 512)	0
dense_1 (Dense)	(None, 4)	2052
Total params: 14,979,396		
Trainable params: 264,708		
Non-trainable params: 14,714,688		

Figure 4.8 Summary of the VGG16

Before starting to train the model, we apply the default number of epochs 100 and the patience 3. 3 patience will indicate that the validate loss is not improving continuously in three epochs. From the result, the highest accuracy values in the training set is 74.16% with the loss 0.6154. Meanwhile in the validation set, the highest accuracy is 81.83% in epoch 7 and the lowest validate loss is 0.4894.

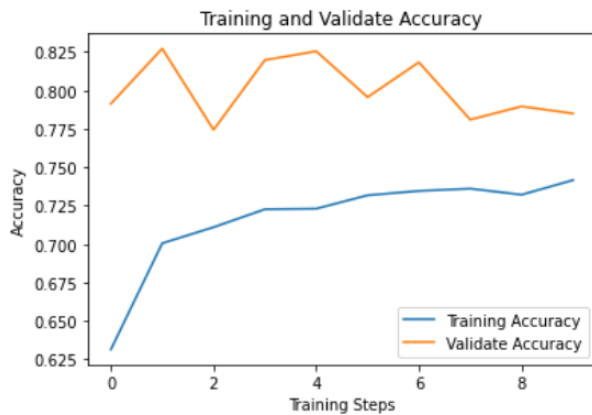


Figure 4.9 Accuracy in training and validation set

From the training and validation accuracy plot, the validation accuracy shows an average of around 0.8 accuracy while the training accuracy shows a value around 0.71 on average. This pattern indicates that the model is not overfitted, but the gap between training and validation accuracy is relatively big, therefore there might be a potential under fitting.

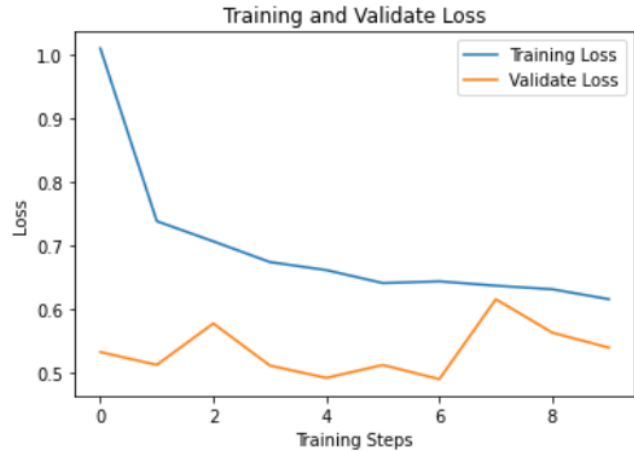


Figure 4.10 Loss in training and validation set

In the figure below, we could see that it has a similar pattern with the accuracy plot above, which the validation loss outperforms the training loss. This is an indication that the model is not overfitting and the gap between training and validation loss is relatively small.

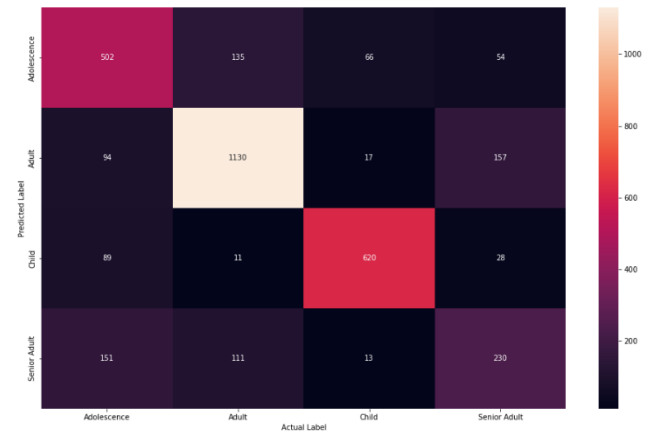


Figure 4.11 Confusion matrix

A confusion matrix plot shown in figure 4.11 is used as an evaluation method to check the performance of the model. In the figure below, we could see that the class 'Child' performs the best, followed closely by 'Adult' class which makes them the better trained class. 'Adolescence' has a TP rate of around 0.6 while the least performing 'Senior Adult' class only has a TP rate at roughly around 0.5. This indicates that the features for 'Child' and 'Adult' class is captured and trained while 'Adolescence' and 'Senior Adult' is currently under fitted in the model.

	precision	recall	f1-score	support
Adolescence	0.60	0.66	0.63	757
Adult	0.81	0.81	0.81	1398
Child	0.87	0.83	0.85	748
Senior Adult	0.49	0.46	0.47	505
accuracy			0.73	3408
macro avg	0.69	0.69	0.69	3408
weighted avg	0.73	0.73	0.73	3408

Figure 4.12 Classification report

This classification report in figure 4.12 shows a summary of the confusion matrix plot. Precision represents the TP rate of the forecast, which 'Child' and 'Adult' scoring 0.87 and 0.81 while 'Adolescence' has a precision rate of 0.6 and 0.49 for 'Senior Adult'. Recall score is the ratio of TP to (TP + FN), which shows that it has a TP rate of 0.81 and 0.83 for 'Child' and 'Adult' while 'Adolescence' has a recall score of 0.63 and 0.47 for 'Senior Adult'. F1-score is used as a harmonic mean of precision and recall score which makes it a better measure of the incorrectly classified cases, behaving very similar in terms of scores for the 4 classes. This is a clear indication of under fitting in 'Senior Adult' and 'Adolescence' class as the model is not able to learn the features for these 2 classes.

4.4 RESNET50 EVALUATION

4.4.1 RESNET50 EVALUATION GENDER MODEL

The ResNet50 model with total freeze layers consists of 24.637 million parameters. Average Pooling is used to calculate the average value for patches of a feature map, and uses it to create a downsampled (pooled) feature map. The total numbers of trainable parameters and non-trainable parameters are 1.050 million and 23.587 million respectively.

Model: "sequential"

Layer (type)	Output Shape	Param #
resnet50 (Functional)	(None, 2048)	23587712
dense (Dense)	(None, 512)	1049088
dropout (Dropout)	(None, 512)	0
dense_1 (Dense)	(None, 2)	1026

=====
Total params: 24,637,826
Trainable params: 1,050,114
Non-trainable params: 23,587,712

Figure 4.13 Summary of the ResNet50

Before starting to train the model, we apply the default number of epochs 100 and the patience 2. 2 patience will indicate that the validate loss is not improving continuously

in two epochs. From the result, the highest accuracy values in the training set is 93.84% with the loss 0.1580. Meanwhile in the validation set, the highest accuracy is 94.60% in epoch 4 but the lowest validate loss is 0.1426 in epoch number 5.

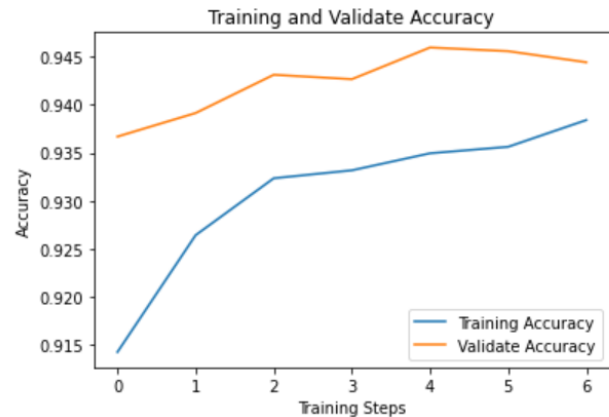


Figure 4.14 Accuracy in training and validation set

From the figure 4.14, we could see that validation accuracy is in the neighbourhood of around 0.94 on average, outperforms training accuracy which has a value of around 0.93 on average. This scenario indicates that the model is not overfitting as the gap between training and validation accuracy is very small and has a 94% accuracy rate on validation data.

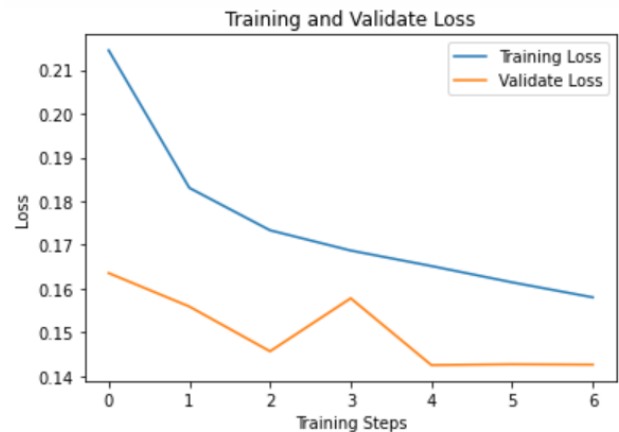


Figure 4.15 Loss in training and validation set

From the figure 4.15, the training and validation loss plot shows very similar patterns with the accuracy plot, with both the validation test outperforming the training set with validation at an average of around 0.15 while averaging around 0.17 on training. This small gap between training and validation loss indicates that the model is not overfitting and is performing well.

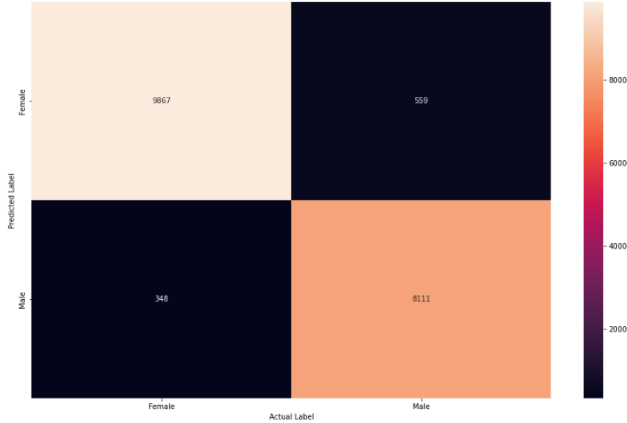


Figure 4.16 Confusion matrix

From the figure 4.16, we could see that both the TP to FP ratio and TN to FN ratio are both roughly at around 0.05. The output of this plot tells us that the performance of the model is very similar to the validation accuracy test, which is around 95% accurate.

	precision	recall	f1-score	support
Female	0.97	0.95	0.96	10426
Male	0.94	0.96	0.95	8459
accuracy			0.95	18885
macro avg	0.95	0.95	0.95	18885
weighted avg	0.95	0.95	0.95	18885

Figure 4.17 Classification report

This classification report from figure 4.17 shows a summary of the confusion matrix plot. Precision represents the TP rate of the forecast, which is 0.97 for female and 0.94 for male. Recall score is the ratio of TP to (TP + FN), which shows that it has a TP rate of 0.95 for female, and 0.96 for male. F1-score is used as a harmonic mean of precision and recall score which makes it a better measure of the incorrectly classified cases, scoring 0.96 for female and 0.95 for male. This indicates that the model is not overfitting and is performing well with around 95% accuracy for validation test images.

4.4.2 RESNET50 EVALUATION AGE MODEL

The ResNet50 model with total freeze layers consists of 24.638 million parameters. Average Pooling is used to calculate the average value for patches of a feature map, and uses it to create a downsampled (pooled) feature map. The total numbers of trainable parameters and non-trainable parameters are 1.051 million and 23.587 million respectively.

Model: "sequential"		
Layer (type)	Output Shape	Param #
resnet50 (Functional)	(None, 2048)	23587712
dense (Dense)	(None, 512)	1049088
dropout (Dropout)	(None, 512)	0
dense_1 (Dense)	(None, 4)	2052
Total params: 24,638,852		
Trainable params: 1,051,140		
Non-trainable params: 23,587,712		

Figure 4.18 Summary of the ResNet50

Before starting to train the model, we apply the default number of epochs 100 and the patience 3. 3 patience will indicate that the validate loss is not improving continuously in three epochs. From the result, the highest accuracy values in the training set is 76.06% with the loss 0.6511. Meanwhile in the validation set, the highest accuracy is 84.96% in epoch 4 but the lowest validate loss is 0.4341 in epoch number 5.



Figure 4.19 Accuracy in training and validation set

From the training and validation accuracy plot, the validation accuracy shows an average of around 0.82 accuracy while the training accuracy shows a value around 0.74 on average. This pattern indicates that the model is not overfitted, but the gap between training and validation accuracy is relatively big, therefore there might be a potential under fitting for the model.

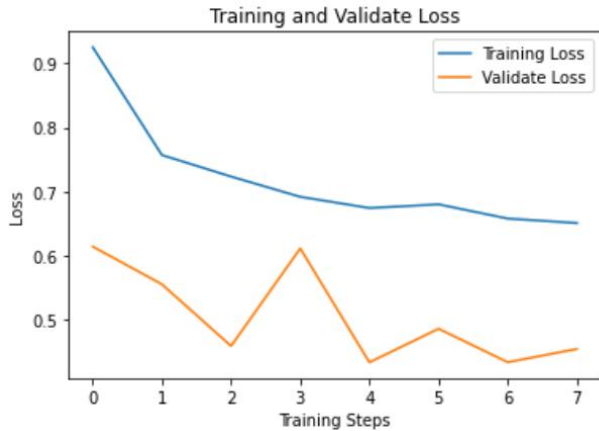


Figure 4.20 Loss in training and validation set

In the figure below, we could see that it has a similar pattern with the accuracy plot above, which the validation loss outperforms the training loss. This is an indication that the model is not overfitting and the gap between training and validation loss is relatively small with validation loss at around 0.55 on average and training loss at around 0.7 on average.

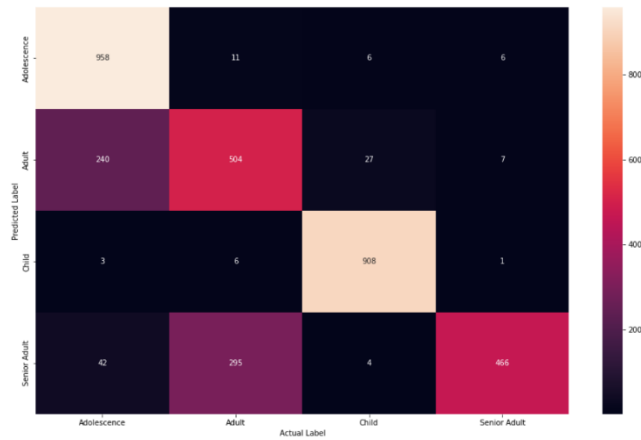


Figure 4.21 Confusion matrix

In the figure 4.21, we could see that the class 'Child' performs the best with a TP rate of around 0.98, followed closely by 'Adolescence' with a TP rate of around 0.975. These two class has successfully captured the features and indicates that these 2 class are not overfitted. 'Adult' and 'Senior Adult' both has a TP rate of around 0.6 making them the worst performers. This indicates that the features for 'Adult' and 'Senior Adult' are unable to capture the features well, making them under fitted.

	precision	recall	f1-score	support
Adolescence	0.77	0.98	0.86	981
Adult	0.62	0.65	0.63	778
Child	0.96	0.99	0.97	918
Senior Adult	0.97	0.58	0.72	807
accuracy			0.81	3484
macro avg	0.83	0.80	0.80	3484
weighted avg	0.83	0.81	0.81	3484

Figure 4.22 Classification report

Unlike previous classification report where the scores of 4 classes stays very similar across precision, recall and f1-score, this model has very different rates under different metrics across the 4 classes except child, therefore f1-score will be used as it is a harmonic mean of precision and recall score which makes it a better measure of the incorrectly classified cases. 'Child' and 'Adolescence' both have a f1-score of 0.97 and 0.86, making them the best performers and indicates that these two classes has captured the features without overfitting the model. 'Adult' class has a f1-score of 0.63 and 0.72 for 'Senior Adult', this indicates that the model are struggling more to learn the features for these two classes, therefore has an under fitted result.

4.5 DEPLOYMENT

After comparing in both VGG16 and ResNet50 models, we decided to make use of the ResNet50 in our web application. The hyperlink for our web application with the best model to predict age and gender in our experiment is <https://share.streamlit.io/jbterrylin/vipproject/Main.py>. The link provided can be accessed from a desktop device. The users are required to upload the image file from their directory for predictions.

After an image is uploaded, the input image and the predicted probability for each of the classes will be displayed.

The figures below shows the flow of the web application.

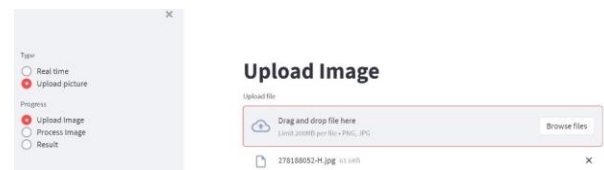


Figure 4.23 Upload image

Figure 4.23 shows the process of uploading an image for predictions.



Figure 4.24 Uploaded image

Figure 4.24 shows the sample uploaded image for predictions.

Chooosed head

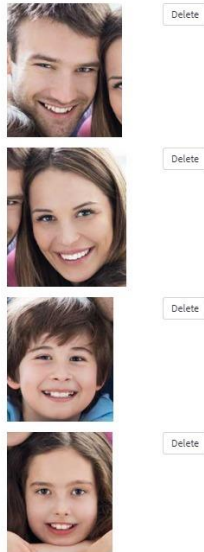


Figure 4.25 Cropped Face

As we mentioned previously, we make use of the algorithms introduced by [4] to detect the face and make the predictions for each of the images cropped.

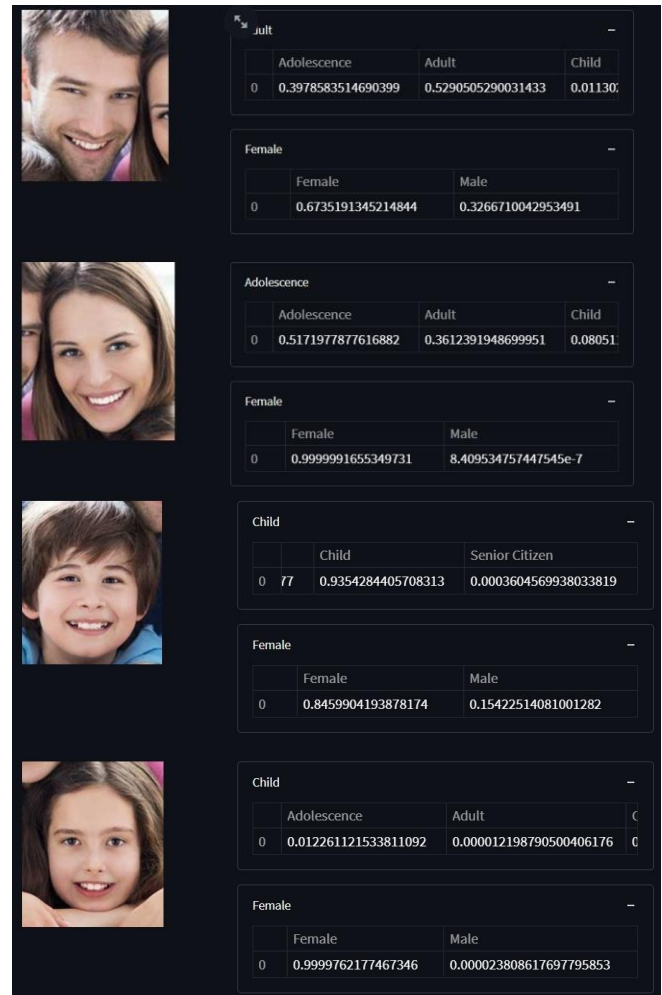


Figure 4.26 Results of the cropped images

Based on the figure 4.24, there are a total of 4 faces in the images, therefore 4 predictions are to be made by our models. From the figure 4.26, we can clearly see the predicted classes and the probability of the result.



Figure 4.27 Image of bounding box with face detected

Figure 4.27 shows the result and the face detection while using live camera detection.

6. CONCLUSION

Overall in this project, we learnt many knowledge and applied them through hands-on experience in the computer vision field. From the data collection, we notice that the Europe look might differ from Asia people. Other than that, the issues of the imbalance data will affect the accuracy when training the model. Besides, we are also exposed to various pre-trained deep learning frameworks and models in order to create our models for our application. According to this process, we learn how transfer learning works. Other than transfer learning, an early stopping and checkpoint technique are also applied which can help us to prevent our model from facing the overfitting issues and save the computation time. When we train the model, there is no background removal applied to the dataset due to all the images we obtained are aligned center.

After that once the web application is developed and hosted at heroku and accessed through the public, we have applied the face detection to the new input images. This is to ensure that the images are centered and cropped correctly from an original image. From this process, this might produce a more precise prediction result.

However, the project is limited to the age group due to limited data collected. Future development directions include numerous regions appearance of humans which can cover not only Europe but also Asia people. Overall, the team is satisfied with the results and the work done through this subject and the project given.

7. TASKS DISTRIBUTION AND FYP DECLARATION

	Wang Lin	Jia Heng	Xi Jie	Zhen Fan
Data Preprocessing	X		X	
Model Creation		X	X	X
Deployment	X			

FYP Declaratio

Lee Wang Lin	Experiments to Verify the Reproducibility of the Crossover Method of Evolutionary Algorithms
Lee Xi Jie	Sentiment Analysis with Objectivity and Subjectivity
Cheok Jia Heng	Macro-economic Time Series Forecasting using Machine Learning Techniques
Oi Zhen Fan	Objectivity and Subjectivity Classification with BERT

8. REFERENCES

- [1] Jana, R., & Basu, A. (2017, February). Automatic age estimation from face image. In *2017 International Conference on Innovative Mechanisms for Industry Applications (ICIMIA)* (pp. 87-90). IEEE.
- [2] Agbo-Ajala, O., & Viriri, S. (2021). Deep learning approach for facial age classification: a survey of the state-of-the-art. *Artificial Intelligence Review*, 54(1), 179-213.
- [3] T. V. Janahiraman and P. Subramaniam, "Gender Classification Based on Asian Faces using Deep Learning," 2019 IEEE 9th International Conference on System Engineering and Technology (ICSET), 2019, pp. 84-89, doi: 10.1109/ICSEngT.2019.8906399.
- [4] Menon, A. (2019, April 23). Face detection in 2 minutes using OpenCV & Python. Medium. Retrieved November 15, 2021, from <https://towardsdatascience.com/face-detection-in-2-minutes-using-opencv-python-90f89d7c0f81>.
- [5] Jangra, A. (2020, May 22). *Gender 200K images: Celeba*. Kaggle. Retrieved November 16, 2021, from <https://www.kaggle.com/ashishjangra27/gender-recognition-200k-images-celeba>.
- [6] Subedi, S. (2018, August 16). *Utkface*. Kaggle. Retrieved November 16, 2021, from

<https://www.kaggle.com/jangedoo/utkface-new>.

[7] *Automatic face aging in videos*. Automatic Face Aging in Videos via Deep Reinforcement Learning. (n.d.). Retrieved November 16, 2021, from <https://dcnhan.github.io/RL-VAP/>.

[8] *VGG16 - convolutional network for classification and detection*. VGG16 – Convolutional Network for Classification and Detection. (2021, February 24). Retrieved November 16, 2021, from <https://neurohive.io/en/popular-networks/vgg16>.

[9] Kaushik, A. (2020, July 21). *Understanding Resnet50 architecture*. OpenGenus IQ: Computing Expertise & Legacy. Retrieved November 16, 2021, from <https://iq.opengenus.org/resnet50-architecture/>.