DATS 6203: Machine Learning II

Instructor: Dr. Amir Jafari

Final report

Group member : Rayna Liu
Group member: Zheyue Wang

Date: 4/11/2021

1 Introduction:

In recent years, the extraction of age, gender, and ethnicity from facial images has been widely used in many applications, including biometrics, security controls, and entertainment. Convolutional Neural Network has achieved remarkable success in face recognition, image classification, and object recognition. This final project uses the Convolutional Neural Network (CNN) on face recognition tasks for age estimation and ethnicity and gender classification.

2 Describe of the Dataset

This dataset includes a CSV of facial images labeled based on age, gender, and ethnicity from Kaggle. The dataset includes 27305 rows and 5 columns: age, ethnicity, gender, image name, and pixels of the image. Based on the pixels of the image, which is 2,304, the image will be reshaped to 48 x 48. Age is a continuous variable with a range of 1 to 116. Ethnicity is a categorical variable in encode labels 0 to 4, representing the White, Black, Asian, Indian, and others. The gender is a categorical variable in the encoded binary label, in which 0 is male, and 1 is female.

2.1 Data overview:

We can first print out first few row of our data frame and look at it:

	age	ethnicity	gender	img_name	pixels
0	1	2	0	201612192036...	129 128 12...
1	1	2	0	201612192227...	164 74 111 ...
2	1	2	0	201612192228...	67 70 71 7...
3	1	2	0	2016122014491...	193 197 19...
4	1	2	0	2016122014491...	202 205 2...
5	1	2	0	201612201449...	195 198 20...
6	1	2	0	201612201450...	208 216 21...
7	1	2	0	2017010919112...	99 142 169...
8	1	2	0	201612192227...	127 127 13...
9	1	2	0	2017010919120...	199 211 21...
10	1	2	0	2017010919210...	136 138 14...
11	1	2	0	201701091922...	253 253 2...
12	1	2	0	201701091935...	223 222 2...

Figure 1: Data overview

Based on Figure 1 above, we have three features which are age, ethnicity, and gender. Besides, our target is pixels.

2.2 Age:

The Age value is showing below:

Age	Number of Image
1	1123
2	482
3	289
...	...
115	3
116	4

For the value of age, the range is from 1 to 116 years old. Besides, there has some missing age (e.g. 106,107,108).

- **Distribution**

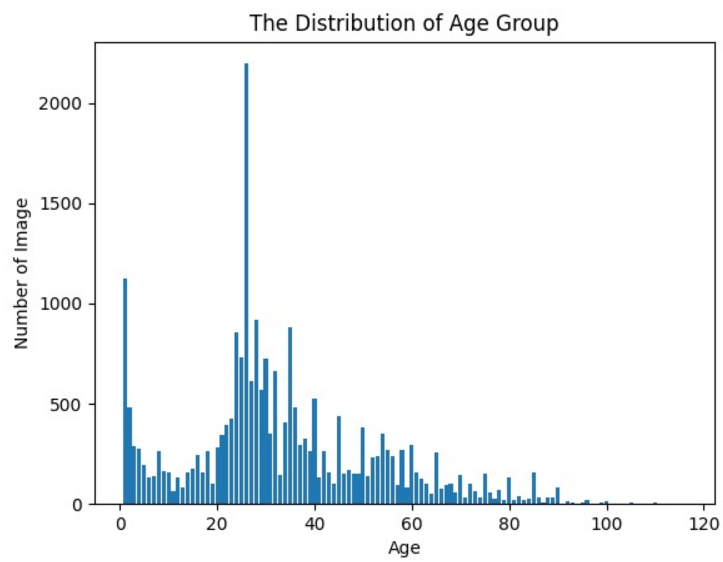


Figure 2: The Distribution of Age group

Figure 2 also shows the unequal distribution of age data. Besides, most of data is concentrate on 26 to 40 years old.

2.3 Gender:

In Gender value, 0 represent as male, and 1 is femal For gender data, we can general look at the data:

Gender	Number of Image
0	12391
1	11314

- **Distribution**

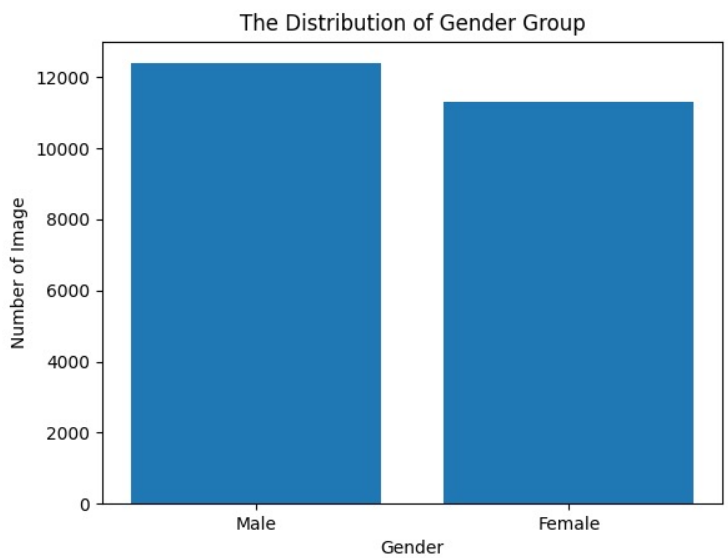


Figure 3: The distribution of Gender group

The figure above shows that the number of male data is little bit large than the number of female data.

2.4 Ethnicity:

In the ethnicity dataset, 0 represent as White, 1 represent as Black, 2 represent as Asian, 3 represent as Indian, 4 represent Others(like Hispanic, Latino, Middle Eastern). The data set is shown below:

Ethnicity	Number of Image
0	10078
1	4526
2	3434
3	3975
4	1692

- **Distribution**

We can also general look at the distribution of Ethnicity:

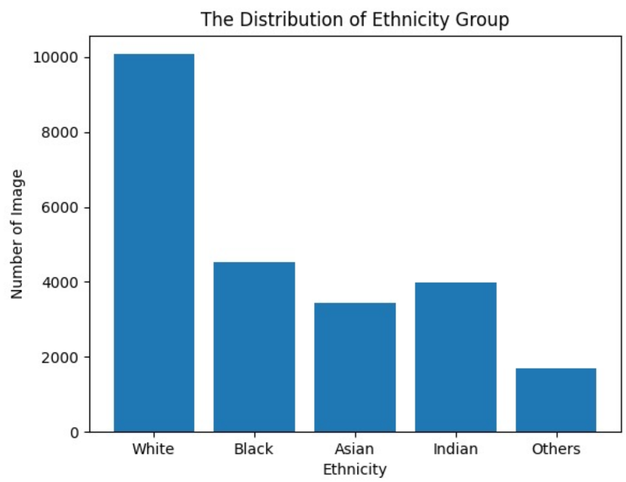


Figure 4: The distribution of Ethnicity group

Figure 12 shows that data comes from the white people is the most.

2.5 Summary Table:

Below is the summary table for the total data-set.

```
In[4]: data.describe()
Out[4]:
```

	age	ethnicity	gender
count	23705.000000	23705.000000	23705.000000
mean	33.300907	1.269226	0.477283
std	19.885708	1.345638	0.499494
min	1.000000	0.000000	0.000000
25%	23.000000	0.000000	0.000000
50%	29.000000	1.000000	0.000000
75%	45.000000	2.000000	1.000000
max	116.000000	4.000000	1.000000

Figure 5: The summary Table

2.6 Image:

Below are some example of images:

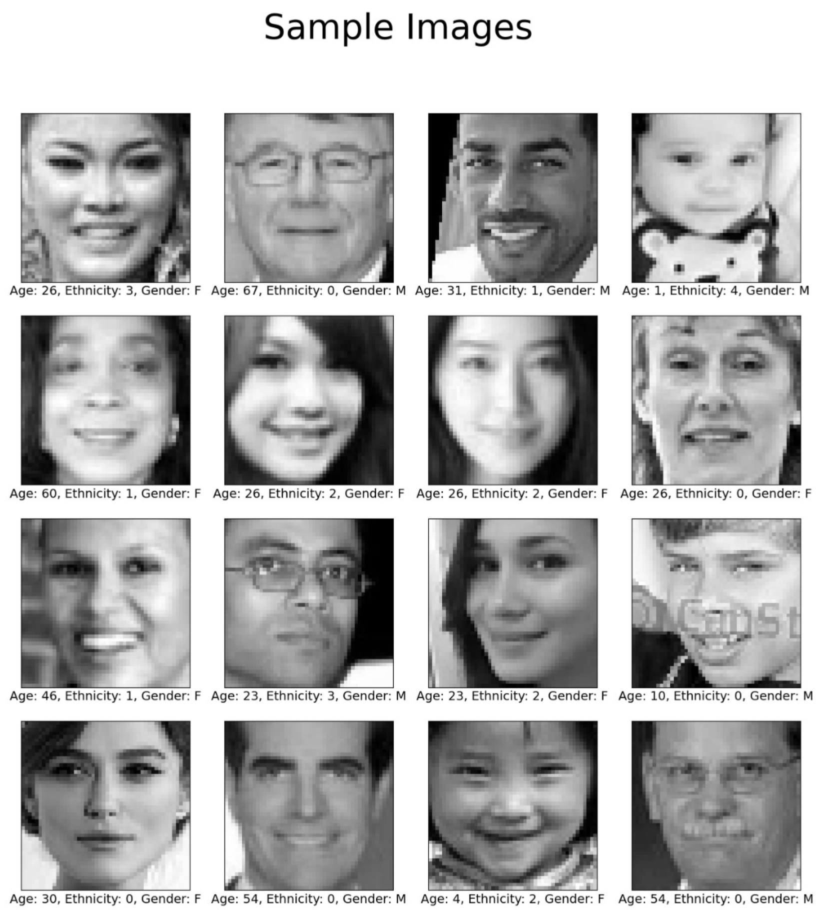


Figure 6: Some image example

3 Deep Learning Network:

The Convolutional Neural Network (CNN) is a multi-layer feed-forward Network, which has the excellent capability of feature extraction of images. The principal convolutional network performs a convolution operation on the image using the convolution kernel. The pooling layer, which follows a convolution layer, reduces the spatial size of the feature map. The fully connected feed-forward layer converts the matrix to vector, generally used before the output layer.

In this project, we used two CNN models to do face recognition tasks, one for age estimation, the other for ethnicity and gender classification. The training algorithm we used for the two models is Adam since it is well suited for problems that are large in terms of data and parameters. In our two models, Adam converges is faster than SGD.

1. **Age Model**

   The biggest challenge with automatic age estimation is that humans age unevenly. The architecture of our age model draws inspiration from The article Deep Convolutional Neural Network for Estimation based on the VGG-face model.

   - Input shape

   For Age model, we use keras to constructed the model:
   Since the size of the photo is 48 x 48, and only has 1 color chanel, so the input shape is (48,48,1).

   - Convoluted layer

   We convoluted 11 layers. For the first 4 convoluted layers, we choose the max-pooling to reduce both time and space complexity and bath normalization for handling the vanishing and exploding problem. After that, we stacked 7 convoluted layers, which both had 512 neurons, 3x3 kernel size, 1x1 stride size, and zero paddings.

   - Activation function

   For this convoluted layer, we choose the ReLU as our activation function, since the age target is regression and the relu function only returns 0 and positive value. We also add a dropout rate of 0.5 to prevent overfitting.

- Flatten layer

The Flatten layer in CNN is similar to the mlp, as the dimension of the input is 1d. For this project, we total have 4 layers include the target layer at the end of the model. The code is shown below:

```
model_age = tf.keras.Sequential([

    InputLayer(input_shape=(48,48,1)),
    Conv2D(64, (3, 3), strides=(1,1), padding='valid', activation="relu"),
    BatchNormalization(),
    MaxPooling2D((1, 1)),
                 Conv2D(128, (2, 2), activation="relu"),
    BatchNormalization(),
    MaxPooling2D((2, 2)),
    Conv2D(256, (3, 3), activation="relu"),
    BatchNormalization(),
    MaxPooling2D((2, 2)),
    Conv2D(512, (3, 3), activation="relu"),
    BatchNormalization(),
    MaxPooling2D((2, 2)),
    Conv2D(512, (3, 3),strides=(1, 1), padding='same', activation="relu"),
    Conv2D(512, (3, 3), strides=(1, 1), padding='same', activation="relu"),
    Conv2D(512, (3, 3),strides=(1, 1), padding='same', activation="relu"),
    Conv2D(512, (3, 3),strides=(1, 1), padding='same', activation="relu"),
    Conv2D(512, (3, 3),strides=(1, 1), padding='same', activation="relu"),
    Conv2D(512, (3, 3),strides=(1, 1), padding='same', activation="relu"),
    Conv2D(512, (3, 3),strides=(1, 1), padding='same', activation="relu"),
    MaxPooling2D((2, 2)),

    Flatten(),
    Dense(512, activation="relu"),
    Dense(4096, activation="relu"),
    Dropout(DROPOUT_age),
    Dense(5000, activation="relu"),
    Dropout(DROPOUT_age),
    BatchNormalization(),
    Dense(1, activation="relu")
```

```
                                    ])
```

2. **Gender & Ethnicity**
   Unlike age targets, ethnicity and gender targets are encoded labels. Therefore, we concatenate ethnicity and gender label in one-hot encoded target and create a model for classifying the multiple-label in each Image. The advantage of using a model is that improving the model's accuracy allows for accurate predictions of both race and gender.

- Input Shape

For Gender + Ethnicity model, we also use keras to constructed the cnn model: Since the size of the photo is 48 x 48, and only has 1 color chanel, so the input shape is (48,48,1).

- Convoluted layer

We convoluted the 6 layers for ethnicity & gender model. For the first 2 convoluted layers, we choose the pooling to reduce both time and space complexity and batch normalization for handling the vanishing and exploding problem. After that, we stacked 4 convoluted layers with batch normalization.

- Activation function

For this convoluted layer, we choose the ReLU as our activation function. Since the target is multiple-label and the sigmoid function only returns between 0 to 1, we placed the sigmoid function in the output layer of the model, which serves to convert the model's output into a probability score, which can be easier to work with and interpret.

- Flatten layer

Similar to age model, we add 11 layer include the target layer at the end of the model. Besides, we input 4096 neurons, and add drop out layer for avoid outfitted.

- Output Shape

We concatenate ethnicity and gender labels in one-hot encode 7 labels. For example, if an image is a white female, the target will be [1,0,0,0,0,0,1].

Below is part of the code:

```
model_eg = tf.keras.Sequential([
        InputLayer(input_shape=(48,48,1)),
        Conv2D(32,(3,3),activation="relu"),
        BatchNormalization(),
        MaxPooling2D((2,2)),
        Conv2D(64,(3,3), activation="relu"),  #64, 21,21
        BatchNormalization(),
        AveragePooling2D((2,2)),      #64, (10,10)
        Conv2D(128,(3,3), activation="relu"), # 128, (8,8)
        BatchNormalization(),
        Conv2D(200,(3,3), activation="relu"),  #200, (6,6)
        BatchNormalization(),
        Conv2D(400,(3,3),activation='relu'), #400,(4,4)
        BatchNormalization(),
        Conv2D(500,(3,3),activation="relu"), #500,(2,2)


# MLP
        Flatten(),
        Dense(500, activation="relu"),
        Dropout(DROPOUT_eg),
        BatchNormalization(),
        Dense(4096,activation='relu'),
        Dropout(0.3),
        BatchNormalization(),
        Dense(4096,activation="relu"),
        Dropout(0.2),
        BatchNormalization(),
        Dense(4096,activation='relu'),
        Dropout(0.2),
        ........
        Dense(4096,activation='relu'),
        Dropout(0.2),
        BatchNormalization(),
        Dense(7, activation="sigmoid")
         ])
```

4 Experimental setup:

4.1 Data splitting

We were reshaping and normalizing the pixels of the input image in the size of (48,48,1). Also, we were extracting the vector of age target (23705,1) and a combined matrix of ethnicity and gender target with the size of (23705,7). After that, we were going to use the train test split from sklearn package and split 70 % into the training data set, 15 % on the validation data set, and the rest 15% on the test data set for both models. Since the ethnicity & gender model has an imbalanced class label, we split the data into a stratified image. From the previous EDA statistical graph, considering the problem of data imbalance, we tried to use Image augmentation to preprocess the data, but it did not improve the model, so it was not used in the end.

4.2 Framework, Performance

For this project, we chose Keras as our framework as Keras is easy and convenient for building the model. Besides, we are going to use the loss function to judge our performance. Since age is a quantitative variable, so we use MSE in loss function and MAE for metric to measure the average magnitude of the errors in a set of age estimation. Gender and ethnicity is a categorical variable and multiple labels, so we use Binary Cross Entropy in loss function and Accuracy for metric.

4.3 Mini-Batches:

Besides, we are going to use a mini-batch for both two models. We have 16593 on training dataset for both models, and setting the batch size to 64 in age model, and setting batch size to 512 in ethnicity and gender model. Thus, for age model, we are going to train $16593/64 = 259.26$ times, train $16593/512 = 32.4$ times for each epoch. Here, the age model setting 200 epoch, ethnicity, and gender setting 64 epoch. Based on the suggestion from the article *How to Choose the Right Mini-Batch Size in Deep Learning* and the memory of our GPU, we were using a power of 2 as the mini-bath size, and we were going to try 64,128,256,512.

4.4 Training parameters:

From the previous experiment, the learning rate and epochs number should be changed simultaneously, a lower learning rate needs more epochs, and a higher learning rate needs fewer epochs. A learning rate greater than 1e-2 is significant in the age model, causing the model to converge too quickly to a suboptimal solution. Learning rate less than 1e-5 is too low to cause the process to get stuck. After 100 epochs, Age's MAE remains below 6.0 and gradually decreases as the epoch increases. However, the training time of the age model is as long as half an hour. We finally set the learning rate to 0.001 with 200 epochs. In the ethnicity & gender model, it could converge to large accuracy within 100 epochs. We finally set the learning rate to 0.0001 with 60 epochs. Considering that both models use many neurons, we select a large number of dropout rates: 0.5 for the age model and 0.3 for the ethnicity & gender model. In both models, we were using performance schedule to adjust the learning rate greatly increase the utility of the model.

4.5 Over Fitting & Extrapolation:

Since our model has a large number of parameter, so the validation accuracy will rise first. When the accuracy is up to the best score, it will decrease as the model is over fitting. Thus, we add a drop-out layer and add early stopping. When the validation accurately hits the best score, the model will save weight and bias. For the age model, when the validation loss stops improving for 25 consecutive epochs, it will stop. For the ethnicity & gender model, when the validation accuracy stops improving for 30 consecutive epochs, it will stop.

5 Results:

5.1 Model Accurate:

Since the age model is hard to reflect on validation accuracy, so we use mae to measure the accuracy. Based on our trained model, the mae of age model is 6.059, and the validation accuracy of ethnicity and gender is 66%. Below are some images with real labels and predicted labels based on our two models.

Figure 7: Sample prediction image

5.2 Analyse the result:

According to our model result, we can analyse the predict accurate by each feature.

- Age

Since our age is goes from 1 to 116, we divided it as 11 age groups, and each group contain 10 age for better analyze.

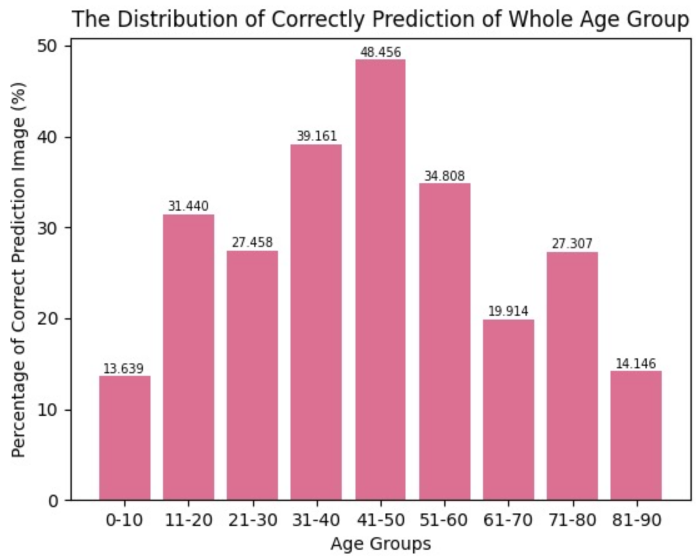
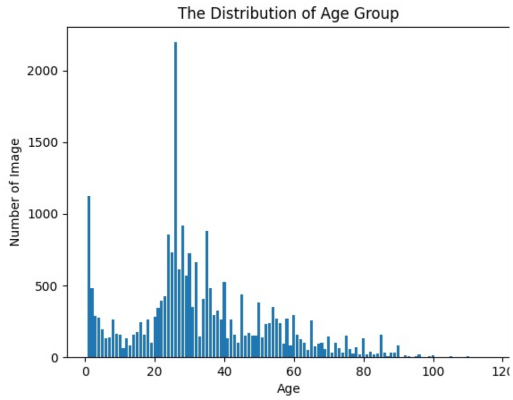
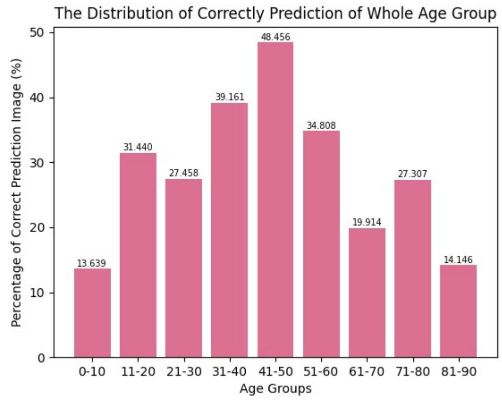


Figure 8: The Distribution of correct prediction

From the Figure 8, since no accurate observation was found in the age groups of 91-100 and 100+, these two age groups were not shown in the figure above. Besides, the 41-50 age group has the highest prediction probability in accuracy compared to the other group age, which is 48.456%. One possible reason is that the sample size of the 41-50 age group is comparatively less to other age groups. Look back to our age data distribution and compare it to our correct prediction of the whole age group.



(a) 1



(b) 2

The sample data of the 20 - 40 age group is higher, especially in age 26, the sample size over 2000. Thus, the large sample might reduce the accuracy of the prediction. Besides, age 0 also contains a large sample which is above 1000, so the accuracy is extremely lower in all age groups. What's more, if the sample size reduces, the accuracy of prediction might be improved. For example, in the age group 11 - 20, and age group 41 - 90, the prediction accuracy will improve. Besides, we can also see the prediction error of the age model.

- **Error Analyse for Age Model:**

For better see our model error, we visualize it by bar chart:

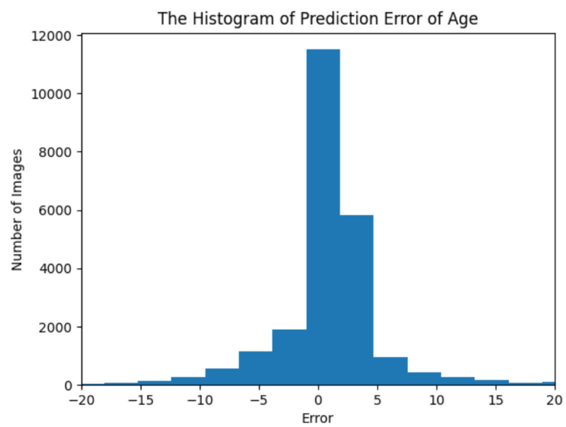


Figure 10: The histogram of prediction error of age

From Figure 10, the predicted error of age is followed the normal distribution. The graph intuitively shows that the average error of the age-predicted by our age model compared with the actual age is less than 5 years. That is to say, if our model gives an image label a person as 28, the predicted age probability will be between 26 and 31. We can also see the accuracy of age model:

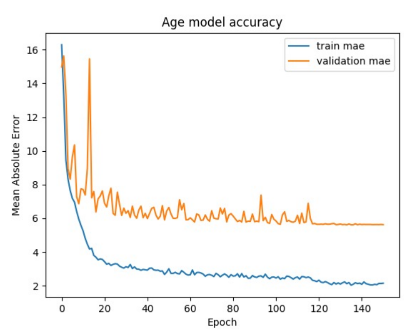


Figure 11: Age accuracy

- Ethnicity

We have 5 different ethnic types, and we visualized the correct prediction image for 5 types by the bar chart. Based on the Figure 12, White, Black, Asian, Indian have

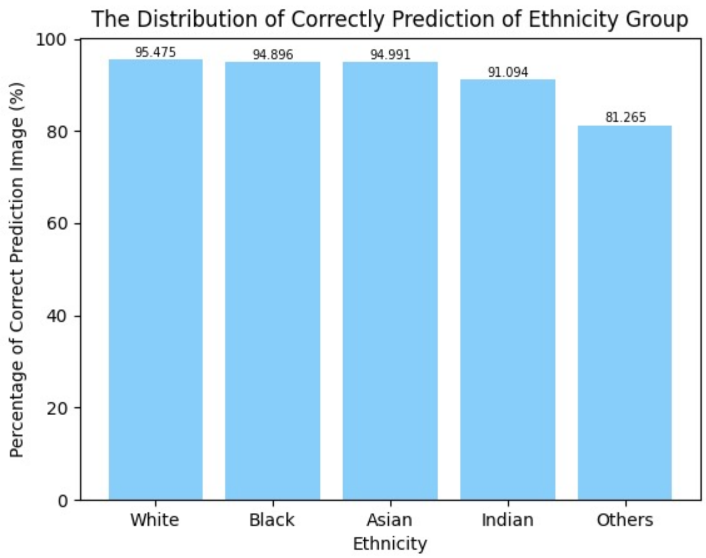
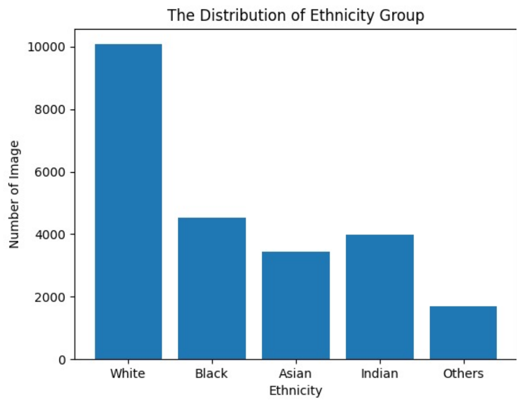
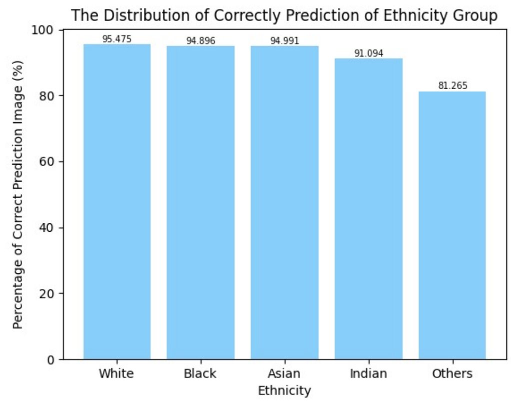


Figure 12: The Distribution of prediction ethnicity

relatively high predict accuracy, as all these 4 types above 90% predict accuracy. Again we can continue to compare the distribution of Ethnicity and the prediction accuracy for each ethnic group.



(a) 1



(b) 2

As shown in the graph, our racial distribution is very uneven, with the major the white people and the minor others. However, the stratification in the split data set plays a crucial role in our model's ability to predict other races well and achieve high accuracy. On the other hand, the prediction ability of our model is weak in others.

- **Confusion Matrix of Ethnicity**

Since ethnicity is a categorical variable, we use a confusion matrix to analyze the model. Below shows our confusion matrix.

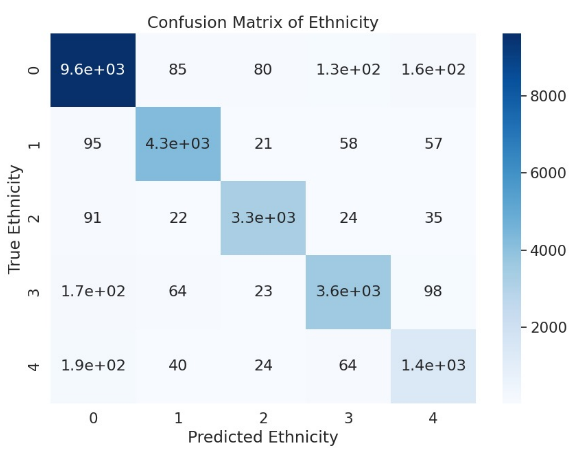


Figure 14: Confusion matrix

Let's first general look at the correct prediction images. Figure 14 shows that around 9600 images of white are predicted correctly, 4300 images of black predict correctly, 3300 images of Asian predict correctly, 3600 images of Indian predict correct, and 1400 images predict correctly.
Let's continue to analyze the predicted data. Among all categorical of the correct prediction, the number of the white image is the heights. Looking back to our data distribution, the data sample of white is significantly higher than others. Thus, it may be explained that most classification of the correct image is white. Besides, even has 9600 images of white predict correctly, still has 170 images are classified as India, and 190 images predict as others. From those wrong classified example, we would say some features between White and Asian, Others are not distinguished well.

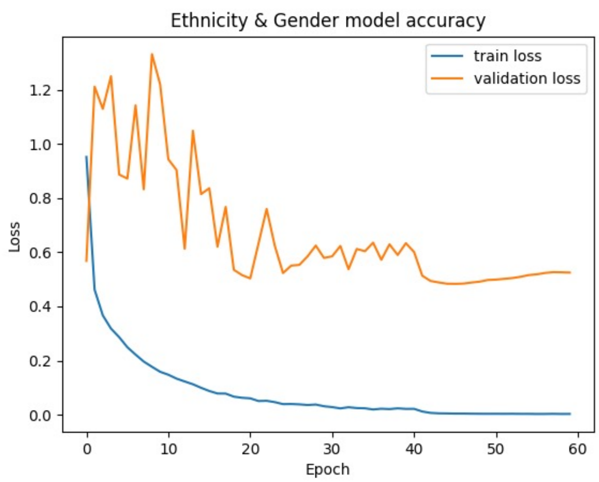


Figure 15: The Accuracy of ethnicity and gender

- Gender

For gender value, we can also visualize the distribution of correct prediction:

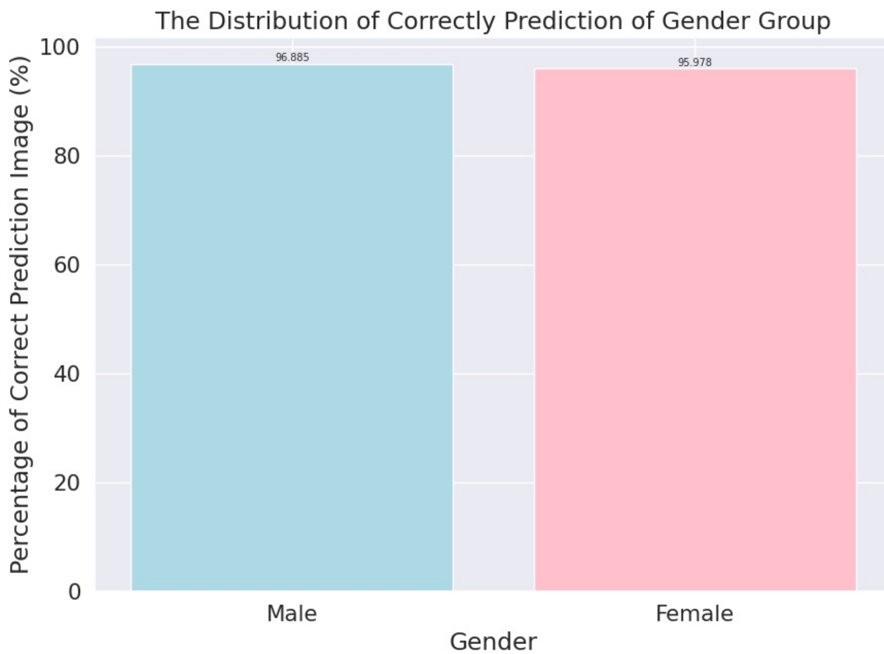
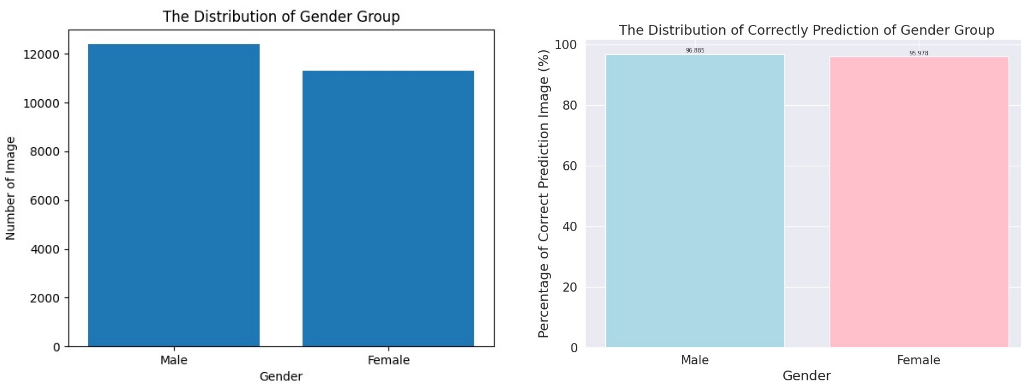


Figure 16: The Distribution of Correct prediction

From the Figure 16, the accuracy for gender was more than 95 % both on male and female. Again, let's compare the distribution of sample size and the distribution of correct prediction



(a) 1 (b) 2

compare the distribution of gender and the prediction accuracy for each gender group. The importance of stratification is demonstrated here to allow both male and female colleagues to be accurately predicted.

- **Confusion Matrix of Gender**

Similar to the ethnicity, we use confusion matrix to analyses the model.

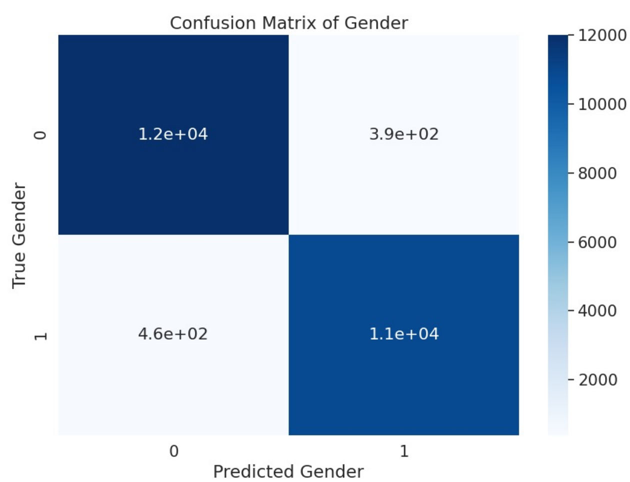


Figure 18: Confusion matrix of age

From the Figure 18, around 12000 male images predict correctly, and around 11000 female images predict correctly. We can calculate some statistics to measure our confusion matrix:

1. Total $= 12000 + 460 + 390 + 11000 = 23850$
2. Accuracy : $(11000 + 12000)/23850 = 0.964$
3. Misclassification Rate (Error Rate): $(390 + 460)/23850 = 0.036 = 1 - 0.964$
4. Actural yes:$(460 + 11000) = 11460$
5. True positive rate (Recall) : $11000/114600 = 0.95986$

Based on the statistic above, around 96.4% image classifier correctly, around 3.6% of image classifier wrong. Among those classified as true images, around 95.986% image indeed classified correctly.

5.3 Show Incorrectly Age Image

Let's see some incorrect age image :

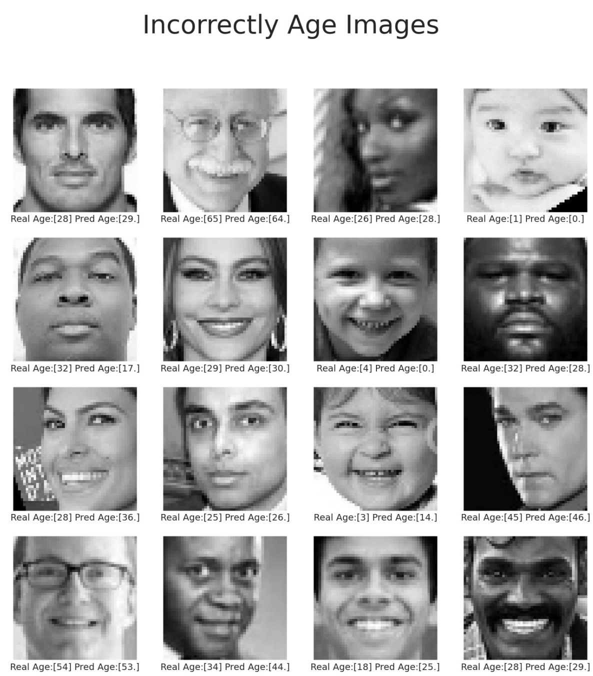


Figure 19: Some incorrectly image (1)

For most adults, the margin of error for age prediction was about a year. After several random checks of incorrectly age images, we found that our age model could hardly predict the age of children (less than 5) accurately and had significant errors because the model could not well identify the age characteristics of children. Once there were some odd expressions, like grimaces, there would show a significant margin of error.

5.4 Show Incorrectly Ethnicity & Gender Image

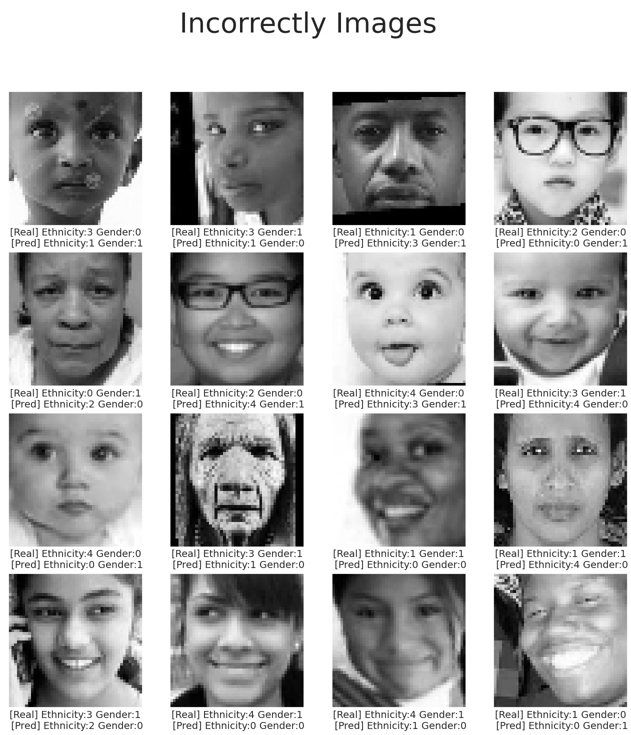


Figure 20: Some incorrectly image (2)

Like the age model, the ethnicity & gender model is inaccurate in predicting a child's ethnicity and gender. The fact that a boy is usually fully physically developed man at the age of 18 and girl is usually fully physically develop woman at the age of 16.

6 Application:

We can also use our model to predict some other images from google:

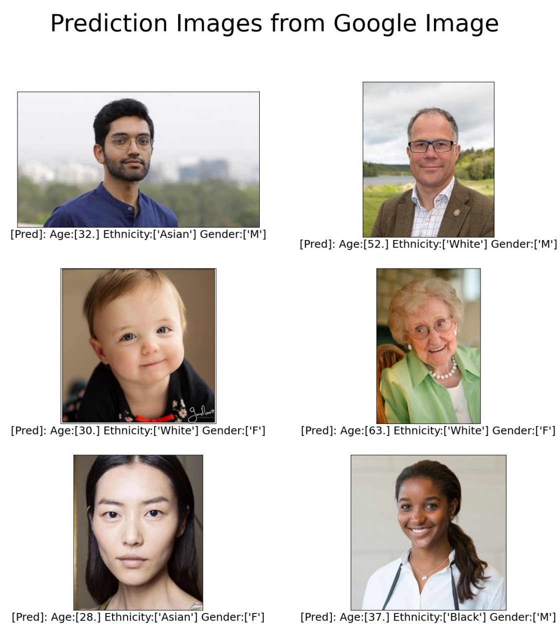


Figure 21: Prediction from google image

7 Summary and Conclusion:

We have presented the two 2-dimensional Convolutional Neural models for age estimation and ethnicity and gender classification with face images. After repeated training of the model, we find that the MAE of age is stable at about 5.7 and that the accuracy of race and gender at the same time is maintained at about 50%. One of the biggest challenges in age estimation is that humans age grows at different rates, with different races and genders showing different characteristics at the same age, So we looked up some literature and worked hard to develop a deep neural network to let it catch up with the age characteristics as much as possible. Since our input image size is (48x48x1), which only has one channel, It is hard for us to resize to 3 channels to fit the pre-trained model VGG16 or ResNet. We put ethnicity and gender together in one model because we want to make sure that we get race rights and gender rights. We did the image augmented for data preprocessing to rotate, flip, and zoom the image for training. However, We found that the model did not improve. In the future, we suggest figuring out to change the channel from 1 to 3 and use the pre-trained model to improve the accuracy.

(Notation: The result we show on the report is based on our trained best model which saved on local. However, as Github limits the upload file size, we are not able to upload our trained best model on Github currently. Thus, the training result might differ.)

References

[1] Kaggle: Age, Gender & Ethnicity Prediction
https://www.kaggle.com/rithikb24/age-gender-ethnicity-prediction.

[2] Deep Convolutional Neural Network for Estimation
https://arxiv.org/pdf/1709.01664.pdf

[3] How to choose the right mini-batch size in deep learning
https://www.mikulskibartosz.name/how-to-choose-the-right-mini-batch-size-in-deep-learning

[4] Amir-jafari: Deep Learning,
https://github.com/amir-jafari/Deep-Learning

[5] Gentle Introduction to the Adam Optimization Algorithm for Deep Learning
https://machinelearningmastery.com/adam-optimization-algorithm-for-deep-learning/

[6] Building powerful image classification models using very little data
https://blog.keras.io/building-powerful-image-classification-models-using-very-little-data.html

[7] Training a TensorFlow model to recognize emotions
https://medium.com/@jsflo.dev/training-a-tensorflow-model-to-recognize-emotions-a20c3bcd6468

[8] Google image
https://www.google.com/imghp?hl=en