香 港 大 學
THE UNIVERSITY OF HONG KONG

MSBA 7011
GROUP 14

AGE AND GENDER DETECTION

Su Jingyi 3035817701
Liu Ruoyang 3035811202
Liu Xiaohan 3035815492
Wu Yi 3035808750
Chen Sijia 3035809792
Li Zixin 3035809778

Work Allocation		
Chen Sijia	3035809792	Face detection for data processing; analysis of ResNet model, corresponding code, report, PPT and presentation
Li Zixin	3035809778	Data preparation and data cleaning, corresponding code, report, PPT and presentation; simple Multilayer CNN model; PPT Formatting
Liu Ruoyang	3035811202	Analysis of VGG-Face model and tuning parameters, corresponding code and report; PPT and presentation
Liu Xiaohan	3035815492	Proposal of the topic, analysis of Multilayer CNN models and tuning parameters, corresponding code, report, PPT and presentation part.
Su Jingyi	3035817701	Age Classification, Analysis of Machine Learning Models, corresponding code, report and presentation; Video Editing
Wu Yi	3035808750	Analysis of Inception V3 Model, corresponding code, report, PPT and presentation; Report Formatting

1. Executive Summary

Our project focuses on age and gender detection. Age and gender detection is one of the most popular research directions in the field of facial intelligent recognition in recent years. One excellent detection function can be used in commercial fields, such as in photo interactive software that is extremely popular with young people, or in real scenarios, such as helping public security systems to conduct investigations. In our project, we used the face images contained in Wikipedia as a database. We then built models of multiple dimensions, including deep learning models that have performed well in related industries these years, and traditional algorithms that are easy to adjust for us. We finally build the age and gender detection models with industry-competitive recognition accuracy. We believe that the models we build have great revenue potential and social value.

2. Introduction

We conduct models to detect age and gender on facial photographs. The overall procedure includes 2 steps, face recognition and result prediction. We mainly focus on the second step. Our models' training is based on the Wikipedia dataset, which is the dataset that was finished cleaning and basic assessment by us, containing 27000+ facial photographs. We conduct traditional machine learning models like PCA, SVC, Tree-Based Models and deep learning models like multilayer CNN, VGG, Inception_V3 and ResNet. Finally, we find that the model based on the ResNet performs best with a rather high test accuracy.

The rest of the report is organized as follows. In section III, we introduced the specific procedure of how we clean and select our data. We also do data exploration in this section, giving a clear picture of how our data distributes. In the section IV, we introduce our approaches detailly, especially introducing how we build structures of our models, how we adjust them and how are the performances. In section V, we list our future worked on the analysis results in the section before, concluded from the data aspect and the models' structure aspect.

3. Data

Our dataset is built on the base of IMDB-WIKI, which is a well-known dataset containing 524,230 celebrity data images crawled from IMDB and Wikipedia. The origin dataset records the date of birth of the celebrity in each photograph, the year that each photograph was taken, the celebrity's name, the face quality in the photograph accessed by the origin team and the judge about whether the photograph contains more than one face.

Concerning the data quality and the overall data amount that we need, we choose the images which were crawled from Wikipedia. We finished the calculation of the celebrity age, using information of the photograph taken year and the birth date of the celebrity. In the data cleaning procedure, we delete photographs with more than one face, face quality score lower than 2.5 and age lower than 0 due to some error information. We then use MTCNN to do facial detection and further delete over 20 photographs which are with bad facial qualities.

The following figures show the way that our final data distributes. Due to the data source's property, most of the celebrities' ages in photographs we have gathered in the range of 20-60. We then further subclass our age data into 5 classes, including ages range from 0 to 18, from 18 to 30, from 30 to 40, from 40 to 60 and over 60. Concerning the age detection functions we build may be mainly used for implications whose main users would be young adults, we think that the distribution of the age in the dataset is acceptable. Then proportions of two different genders in our dataset are 72.34% and 27.66%, the male proportion is larger than the female proportion.

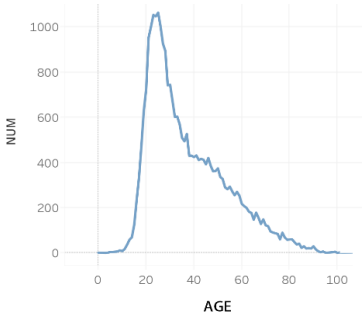

Figure 3-1: Age Distribution

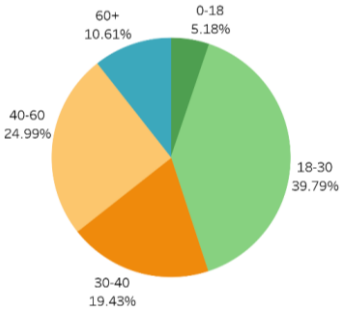

Figure 3-2: Age-subclass Distribution

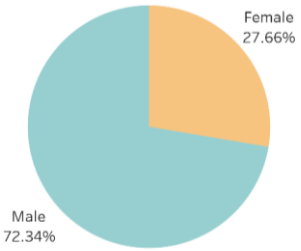

Figure 3-3: Gender Distribution

4. Analysis

4.1 Traditional Machine Learning Methods

First, all the image data are trained by traditional machine methods including PCA, SVC and Random Forest based on the existing research. After shuffling the original data set, all the data are resized to 32 × 32, converted to 3D vectors, and then encoded by Eigenface. Encoder to a final 27,123 × 1,024 dataframe including 1,024 features. Considering the data volume, the whole set is divided into a training set which contains 25,000 images accounting for about 90 percentages of the original set and a test set which contains the remaining 2,123 images.

The performance of each simple machine learning model is shown as Table 4.1-1 and Table 4.1-2 below, which tell that the PCA model gives the most accurate test results among all the models no matter detecting age or gender, and SVC Linear ranks second, followed by Random Forest and then SVC Polynomial.

Model	Train Accuracy	Test Accuracy	Training Time
PCA	60.63%	46.87%	9m 54s
SVC Linear	50.75%	43.48%	17m 46s
SVC Polynomial	24.91%	25.95%	12m 6s
Random Forest	75.08%	41.22%	18m 2s

Table 4.1-1: Models of Age Detection

Model	Train Accuracy	Test Accuracy	Training Time
PCA	60.63%	46.87%	9m 54s
SVC Linear	50.75%	43.48%	17m 46s
SVC Polynomial	24.91%	25.95%	12m 6s
Random Forest	75.08%	41.22%	18m 2s

Table 4.1-2: Models of Gender Detection

Besides, there are some details in training and tuning these machine learning models. In the PCA model, it shows that 600 features explain 98 percentages of variance and then the remaining 424 features have little effect on explaining the variance. And when tuning parameters of bootstrap in Random Forest, randomly selecting 600 features at most at each splitting node gives the highest age and gender test accuracy.

4.2 Multilayer CNN Model

CNN model is very useful for image analysis, which is based on the convolutional layers, pooling layers, and fully connected layers, combined with other proper functions. Here, we first do so some model building and hyperparameters tuning to make some improvements.

Before building the model, the data are required to conduct some data preprocessing. For the CNN model, we first convert the images into a size 128×128. The original length of observations after cropping is 27123, and the shape of the Image data we use for the CNN model is (27123, 128, 128, 3). The age label is settled at [(0,18), (18,30), (30,40), (40,60), (60,100)]. And the age label is settled at [‘Male’, ‘Female’].

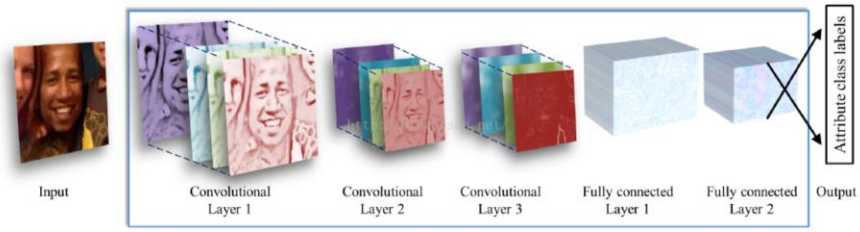

Figure 4.2-1: Framework of Multilayer CNN model

The multilayer CNN models are constructed based on convolutional layers, pooling layers and the fully connected layers. And there also exist several hyperparameters to tune, such as kernel size, activation function, pooling method and the number of layers (convolutional, pooling or fully connected). Considering this, we tune the parameters using the 80% training data and 10% validation data, then test the accuracy through the rest 10% testing data.

We try both TensorFlow and Pytorch Package to conduct the analysis and the results of the two packages are relatively similar. And the activation function is set to be "ReLU", and the pooling method is set to be Max Pooling. Then, we conduct some tuning based on different combinations of hyperparameters. Here, we mainly choose four directions. 1) Kernel Size; 2) Number of convolutional layers; 3) Number of fully connected layers and output channels; 4) Dropout Rate. Under each direction, we randomly pick the proper values to combine with other unchanged hyperparameters.

We first randomly select the number of epochs, and add correspondingly, but there may be overfitting problems. Considering this, in the tuning process, we not only add the dropout rate and tune it to add penalties, but also introduce a function in the "keras" package called 'EarlyStopping'. With this function, we ought to choose a large number for epochs and have a better training outcome. In addition, we also introduce a function called 'ReduceLROnPlateau' to constrain on the learning rate and gradually reduce the learning rate to get to the optimal model accurately. The performances of each tuning process are listed in Table 4.2-2 for Age model and Table 4.2-3 for Gender model.

Number of Conv layers	Kernel Size	Number of Fully Connected layer	Dropout rate	Train Accuracy	Validation Accuracy	Test Accuracy	# epochs
2	2	2(256, 5)	0.3	46.33%	53.12%	46.81%	38
2	3	2	0.3	47.66%	57.81%	49.28%	54
2	5	2	0.3	44.84%	53.91%	45.89%	35
2	2	2	**0.1**	45.70%	53.91%	44.08%	28
2	2	2	**0.2**	46.56%	50.78%	47.36%	36
2	2	2	**0.3**	46.33%	53.12%	46.81%	38
2	2	2	**0.5**	40.62%	48.44%	39.43%	23
2	2	**2 (256, 5)**	0.3	46.33%	53.12%	46.81%	38
2	2	**3 (256, 128, 5)**	0.3	40.39%	49.22%	41.42%	31
2	2	**3 (128, 64, 5)**	0.3	44.06%	52.34%	45.37%	48
2	2	2	0.3	72.13%	57.03%	52.49%	24
3	3	2	0.2	61.03%	59.38%	54.11%	17
5	3	2	0.2	57.27%	64.84%	57.35%	25

Table 4.2-2: Hyperparameters and Accuracy for CNN Age model

Number of Conv layers	Kernel Size	Number of Fully Connected layer	Dropout rate	Train Accuracy	Validation Accuracy	Test Accuracy	# epochs
2	**2**	2	0.3	86.48%	83.59%	86.06%	62
2	**3**	2	0.3	87.03%	85.94%	85.87%	52
2	**5**	2	0.3	84.53%	83.59%	83.25%	47
2	2	2	**0.1**	87.27%	85.16%	85.91%	32
2	2	2	**0.2**	85.02%	83.59%	85.61%	51
2	2	2	**0.3**	86.48%	83.59%	86.06%	62
2	2	2	**0.5**	85.31%	82.03%	83.99%	45
2	2	**2 (256, 2)**	0.3	86.48%	83.59%	86.06%	62
2	2	**3 (256, 128, 2)**	0.3	83.13%	82.03%	83.77%	52
2	2	**3 (128, 64, 2)**	0.3	72.34%	80.47%	79.49%	12
2	2	2	0.1	97.43%	85.94%	88.97%	50
3	2	2	0.1	94.79%	87.50%	90.49%	24

Table 4.2-3: Hyperparameters and Accuracy for CNN Gender model

The results are shown below:

1) Tuning the hyperparameter like kernel size, number of fully connected layers, and dropout rate, indeed increases the accuracy but not so much.
2) Kernel Size: The kernel size for Age model is best to be set at 3 while the kernel size for gender model should be size at 2.
3) For both Age and Gender models, increasing the number of fully connected layers will not affect the accuracy much, but changing the output channels will help change the accuracy level.
4) For both Age and Gender models, increasing the number of convolutional layers will indeed increase the total accuracy level. Compared to other tuning processes, adding convolutional layers has the most efficient outcomes.
5) The largest accuracy level attained by the age model is 54.11% while the largest accuracy level attained by the gender model is 90.49%.

4.3 VGG Mode

VGG-face model is an advanced neural network trained for face recognition tasks, it is proposed by O. M. Parkhi, A. Vedaldi, and A. Zisserman in 2015 (Reference 2). This package contains the Torch models for computation of "VGG Face" descriptor which can be directly gained from the public website *http://www.robots.ox.ac.uk/~vgg/software/vgg_face/*. In our study, we first reproduce part of the VGG-face model, including 8 conventional layers and 3 fully connected layers. After that, we add a dropout layer with the rate of p = 0.5 to avoid overfit. The training process is initiated by learning rate = 0.1 and decreasing to tune until there is no improvement in validation accuracy. Then, we try to change the activation functions (ReLU), and fine tune the fully connected layers.

Finally, the test accuracy for age is 57.1% while the test accuracy for gender is 84.5%. According to the result, although there is no improvement of gender estimation through the VGG-Face CNN algorithm, the performance of age prediction is more efficient compared to the previous multilayer CNN algorithm, which exhibits an obvious 9.7% improvement in age estimation.

To further explore the performance of VGG in age estimation, table 4.3-1 provides the confusion matrix of the 5 distinct age-classifications. According to the confusion matrix, the subgroup of age in (30,40) confuses the most and exhibits the lowest accuracy of 41.85%, and most of the mistakes happen in the adjacent groups of (18,30) and (40,60). It is reasonable since there is relatively little facial changes among middle-aged people. Similar situation also appears in the group of (40,60), which displays the second lowest accuracy of 48.91%. In contrast, images for younger and older people always display more distinguishable facial features, which can lead to more accurate classifiers. Both (0,18) and 60+ subgroups show relatively better age-estimation. The subgroup of age between 18 and 30 exhibits the highest accuracy, 68.14%. In addition to the cause of typical facial features in this age group, the other main reason should be the large number of samples of (18,30) in original data. 39.79% of the age-data are classified as (18,30), hence it can be trained more and classified more accurately compared to other age subgroups.

Pred (%)	(0,18)	(18,30)	(30,40)	(40,60)	60+
(0,18)	**63.84**	26.76	7.56	1.77	0.07
(18,30)	9.32	**68.14**	17.29	4.38	0.87
(30,40)	1.42	21.58	**41.85**	27.71	7.44
(40,60)	0.07	1.62	17.81	**48.91**	31.59
60+	0.01	0.83	4.37	31.28	**63.51**

Table 4-3.1: The confusion matrix of the 5 distinct age-classification by using VGG model

4.4 Inception V3 Model

Inception is a convolutional neural network architecture introduced by Google. The model itself is made up of symmetric and asymmetric building blocks, including convolutions, average pooling, max pooling, concats, dropouts, and fully connected layers. Batchnorm is used extensively throughout the model and applied to activation inputs. Loss is computed via Softmax.

We used transfer learning from a pre-trained model, setting the parameters within the pre-trained model to be non-trainable while only optimizing the parameters of the subsequent dense layers during training. The whole procedure of transfer learning using Inception V3 is as below:

1) Data preprocessing and augmentation
2) Download model wights, import model, load weights into model
3) Set layer to be non-trainable for pre-trained model
4) Obtain last layer output of the pretrained model
5) Adding dense layers after pre-trained model
6) Train the model
7) Evaluate the model

As in original Inception V3, it uses $299 * 299$ size. So we stick with that size in all of our experiments. Apart from these we also perform data augmentation to make the data more robust and generalized. Basically, in data augmentation training data are altered and trained through the network to expose more variety of samples to the network. It helps the network to be trained in a robust and generalized way. As our dataset are not quite large enough for deep networks, proper augmentation plays a great role here. Augmentation involves image rotation, flipping, cropping, shearing etc. In our experiments, we created images considering up to 40-degree rotation, 20% width and height shift, 20% shearing, 20% zoom range, randomly flip inputs horizontally and fill the points outside the boundaries of the input according to the nearest points. This helps expose the model to different aspects of the training data and reduce overfitting.

We have tried to analyze the results in terms of different optimization techniques, learning rate, batch size, and so on. After tuning, the best parameters that we got are shown in table 4.4-1. To improve the model performance and avoid over-fitting, we utilized a callback function to monitor the validation accuracy, it will reduce the learning rate when the validation accuracy has stopped improving for 5 epochs. Also, we did an early stopping, observing the validation accuracy with a patience of 10 epochs during training. After training, we select the best weights for our model based on maximum validation accuracy.

Bests Tuning Parameters		
Tuning Parameters	**Gender**	**Age**
Optimizers	Adam	Adam
Learning Rate	0.0001	0.0001
Batch Size	32	32
Number of Dense Layers	3 layers	4 layers
Activation Function	ReLU	ReLU
Initializer	He Initializer	He Initializer

Table 4.4-1: Best Parameters of Inception V3 Models

After the whole training process, we got a 58% test accuracy for age detection. From the confusion matrix, we can see that age between 18-30 got the highest true-positive rate. While for other categories, most of them are mistakenly categorized to age between 18-30. The reason might be the sample size of 18-30 is much larger than other categories. Similarly, the gender detection, with a test accuracy of 93%, also shows that male has higher true-positive rate than female, caused by the unbalanced sample size.

4.5 ResNet Model

A residual neural network (ResNet) is a kind of advanced neural network, which is proposed to avoid the problem of vanishing gradients, or to mitigate the degradation problem, where adding more layers to a suitably deep model leads to higher training error.

ResNet uses residual blocks as its unit. Denoting the desired underlying mapping as H(x), the residual block uses nonlinear layers fitting another mapping of F(x)=H(x)-x, and the original mapping is recast into F(x)+x. ResNet stacks residual blocks on top of each other to form a network. For example, ResNet-50 has fifty layers using these blocks.

We make multiple steps to improve the performance of the ResNet method. As for the dataset, we do data augmentation to enlarge our dataset. Besides, after using ResNet-50 as base model, we add Dropout layer with the probability of 0.5 to avoid overfit, and add several dense layers as fully connected layers with 'ReLU' as activation function. To avoid overfitting, we use 'EarlyStopping' function with the patience of 20, which means that when the validation accuracy reaches the highest value and could not be larger than the peak in next 20 epochs, the training process stops even if the maximum epoch does not reach. Additionally, we adjust the learning rate by the function of 'ReduceLROnPlateau'. We set the 'patience' as 3 and 'factor' as 0.1, which means that when the validation accuracy reaches highest value and no higher value occurs in the next three epochs, the learning rate will be multiplied by 0.1. Finally, we adjust our 'step_per_epoch' to higher value to use more data in one epoch. Lastly, the test accuracy of ResNet algorithm for age is 59.2%, and for gender is 96.7%.

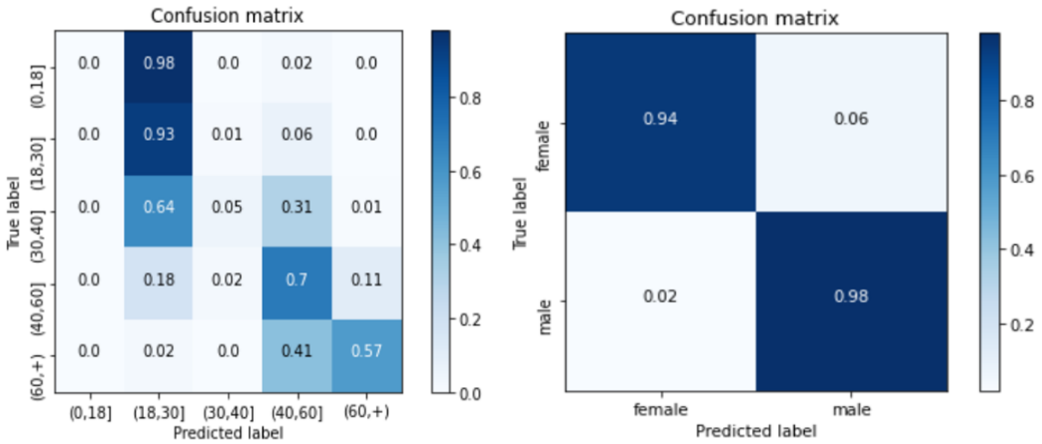

Figure 4.5-1: Confusion matrix of test data for age (left) and gender (right)

As figure 4.5-1 shows, for age detection, class 1 (age from 18 to 30) and class 3 (age from 40 to 60) are more accurate, and other three classes tend to be misclassed. For class 0 (age from 0 to 18) and class 2 (age from 30 to 40), the model tends to classify them to class 1. For class 4 (age over 60) nearly half cases are classified to class 3. Therefore, the high accuracy of age detection happens because of the high proportion of class 1. For gender detection, most of the cases are accurate. Therefore, this model is quite suitable for gender detection.

5. Conclusion and Future Work

5.1 Conclusion

After applying several traditional machine learning methods, multilayer CNN, VGG, Inception and ResNet, we find ResNet method gives us highest test accuracy. Table below shows the test accuracy and the number of trainable parameters for age and gender for each

model. Although the ResNet model gives us highest test accuracy, it has a larger number of trainable parameters than the Inception V3 model, and the training time for the ResNet model is also higher than that of the Inception V3.

Model	Features	Test accuracy	Number of Trainable Parameters
PCA	age	46.87%	1,024
	gender	85.12%	1,024
SVC Linear	age	43.48%	1,024
	gender	79.93%	1,024
SVC Polynomial	age	25.95%	1,024
	gender	73.81%	1,024
Random Forest	age	41.22%	1,024
	gender	77.44%	1,024
Multilayer CNN	age	54.11%	8,410,405
	gender	90.49%	8,398,434
VGG	age	57.1%	36,803,100
	gender	84.5%	27,176,000
Inception V3	age	58.74%	11,345,669
	gender	93.7%	20,845,058
ResNet	age	59.2%	23,751,122
	gender	96.6%	23,750,969

Table 5.1-1: Results of All Models

We use the model trained by the ResNet method to make age and gender detection for a photo from Downton Abbey as a real case for age and gender detection. The original photo is shown as figure 5.1-2. After face detection by MTCNN, we input those faces into the model and output is shown in figure 5.1-3. Most of the age and gender prediction for the figures in the photos are correct, which confirmed the feasibility of this method.

Figure 5.1-2: Original photo from Downton Abbey

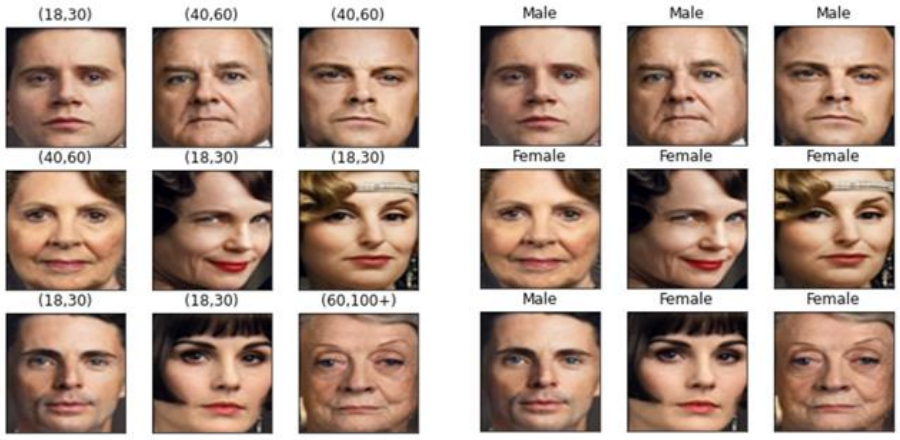

Figure 5.1-3: Age Detection (left) and Gender Detection (right) Results

5.2 Future Work

1) The current dataset is not well diversified. This dataset contains about 72.34% male versus 27.66%. And age ranging from 18 to 40 takes up about 60% while age ranging from 0 to 18 is only 5.18%. Therefore, in the future, we can try our models and analysis on a better diversified and complete dataset to gain better training outcomes and classification results.

2) Due to the time limit, we can only do the data preprocessing and filtering in a basic way. But the reality is that there may be some outliers and false data contained in the dataset. In our dataset, we computed the age of the person by subtracting the birthday from the date when the photo was taken, but the recordings of some dates are obviously misleading, leading to the wrong age. Besides, some photos in the dataset are cartoon figures but not real figures, which bring noise to our training model. But those checking and filtering may need careful observation by humans which may take a long time.
3) Our dataset has only 27,123 photos, which is not large enough for big data analysis. In the future, we can use dataset containing larger size of photos to get a more robust model.
4) For each model, we have to tune some hyperparameters and find the best matching. But also due to the limited time span, we can not go through each of the combinations. Therefore, for future work, we can not only get to know the trend by tuning parameters but also try enough combinations to get a more precise conclusion.
5) In this project, we only consider two targeting variables, that are age and gender. But for future work, we could add more features of a facial image for detection, like the ethnicity, and even the emotion of that person.
6) In this project, what we use to identify the age and gender are static photos, but currently the algorithm could help us to identify those features through videos or even live cameras. Considering this, we could update our algorithm to identify dynamic facial expressions and give a quick response within seconds.

6. Bibliography

G. Levi and T. Hassncer, "Age and gender classification using convolutional neural networks," 2015 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 2015, pp. 34-42, doi: 10.1109/CVPRW.2015.7301352. Batprem. (2020, September 17). Age Group Classification withEigenface and SVM. Kaggle. https://www.kaggle.com/batprem/age-group-classification-witheigenface-and-svm.

Goncharov, I. (n.d.). ivangrov/Datasets-ASAP. GitHub. https://github.com/ivangrov/Datasets-ASAP.

IMDB-WIKI – 500k+ face images with age and gender labels. IMDB-WIKI - 500k+ face images with age and gender labels. (n.d.). https://data.vision.ee.ethz.ch/cvl/rrothe/imdb-wiki/.

Mansar, Y. (2018, May 5). Predicting apparent Age and Gender from face picture : Keras + Tensorflow. Medium. https://medium.com/@CVxTz/predicting-apparent-age-and-gender-from-face-picture-keras-tensorflow-a99413d8fd5e.

Omkar M. Parkhi, Andrea Vedaldi and Andrew Zisserman. Deep Face Recognition. In Xianghua Xie, Mark W. Jones, and Gary K. L. Tam, editors, Proceedings of the British Machine Vision Conference (BMVC), pages 41.1-41.12. BMVA Press, September 2015.

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., & Wojna, Z. (2015, December 11). Rethinking the Inception Architecture for Computer Vision. arXiv.org. https://arxiv.org/abs/1512.00567.

Team, K. (n.d.). Keras documentation: Keras Applications. Keras. https://keras.io/api/applications/.

K. Zhang, Z. Zhang, Z. Li and Y. Qiao, "Joint Face Detection and Alignment Using Multitask Cascaded Convolutional Networks," in IEEE Signal Processing Letters, vol. 23, no. 10, pp. 1499-1503, Oct. 2016, doi: 10.1109/LSP.2016.2603342.