Age Prediction with Blurry Image by Deep Learning

Yu-Chen Chao[*], Wei-Jer Chang [†], Chih-Ling Chang [‡], and Tsungnan Lin[*†‡]
[*] Department of Mechanical Engineering, National Taiwan University
[†] Department of Mechanical Engineering, National Taiwan University
[‡] Department of Mechanical Engineering, National Taiwan University
[§] Cybersecurity Technology Institute, Institute for Information Industry

***Abstract*—Nowadays, with the help of deep learning, image recognition has been widely implemented in different applications, such as face detection, image searching. However, most of the models were trained by clear images, which means, at best, machines only have the same ability of recognizing images as humans do. This problem has limited the variance of applications, especially in security aspect. For examples, Surveillance cameras, or so-called CCTVs, often capture low quality images, which causes the difficulties during investigation.**

In this paper, we will mainly tackle the low quality image recognition problem in CCTVs by training a *Convolutional Neural Network* (CNN) model which can predict the age of the person with his/ her blurry face images. We will implement different blurring methods, including Gaussian, Bilateral, Median. Also, rotation is considered.

I. Introduction

Recently, a BBC reporter carried out an interesting experiment in Guiyang, the capital of Guizhou province of Southwest China. The experiment showed how long the reporter can stay hidden from CCTV cameras in the city. And it ended up, astonishingly, taking only seven minutes to find the reporter by using CCTVs. The reason behind this impressive result was the implementation of image recognition using deep learning.

Without a doubt, recognition ability in CCTV cameras has greatly improved since the machine learning was introduced. However, most of CCTV cameras are still suffering from low-quality problem, which makes it hard to realize the potential value of CCTVs.

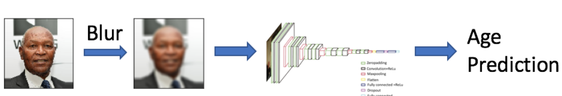

Fig. 1. Overview of our project

In this paper, we mainly focus on training a *Convolutional Neural Network* (CNN) model which can recognize the age of the person by his or her blurry and rotated image. Also, we will find the maximum recognition ability in each blurring methods. However, different training mechanisms show considerable variances in result.Thus, constructing process is also written in the context.

The rest of the paper is organized as follows. The next section will introduce the data set used in this report and each blurring method is explained here. Section III elaborates the architecture of neural network and presents training mechanism. Section IV provides comprehensive experimental results. Eventually, we will conclude this paper in section V.

II. Preparation

A. Data Set

The human face data set used in this report is from IMDB-WIKI 500k+ face images with age and gender labels published on International Conference on Computer Vision (ICCV) in 2015. It contains 62328 images and age labels. We delete 1402 invalid images and labels corresponded from the dataset due to invalid age label reason. Table I shows the age distribution after modifying data set.

TABLE I
AGE LABELS DISTRIBUTION IN DATASET

Range of age	Number of images
under 0	1402
0-10	202
11-20	6693
21-30	23945
31-40	10607
41-50	7250
51-60	5361
60+	6368
Total	62328

B. Blurring methods

1) Average Blur: Average blur is done by convolving the image with a normalized box filter. It simply takes the average of all the pixels under kernel area and replaces the central element with it.

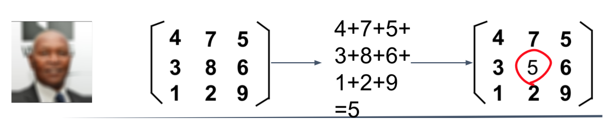

Fig. 2. Using average blur on a 3x3 metric

2) Gaussian Blur: Gaussian blur lies in the way that the pixels in the kernel are weighted when the new center pixel value is computed. The pixels in the kernel closer to the center pixel carry more weight than the pixels near the edge of the kernel.

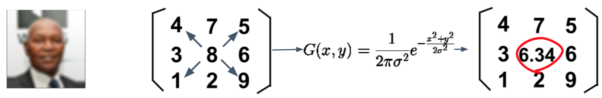

Fig. 3. Using Gaussian blur on a 3x3 metric

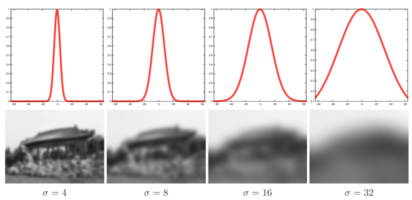

Fig. 4. Blurring effects of different sigmas

3) Median Blur: Median blur is done with a similar manner as Average Blur. The center pixel is replaced with the median value among the pixels under the square kernel.

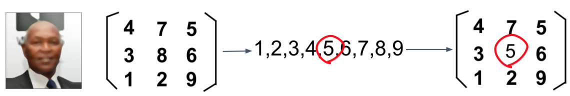

Fig. 5. Using Median blur on a 3x3 metric

III. Network Architecture

The CNN network used in this report is based on the former work, VGG-16, and is revised by inserting different layers into the structure. The input shape of the RGB image is 224x224x3. The detail of the network is following:

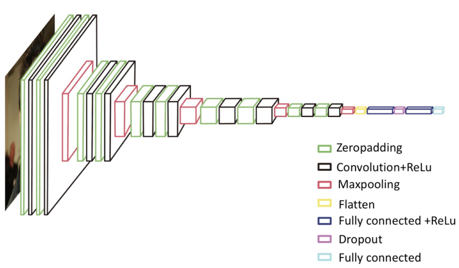

Fig. 6. Outlay of Network Architecture

A. First block

To control the size of all images, we add a Zero-padding layer with (1,1) padding, which can add rows and columns of zeros at the top, bottom, left and right side of all image tensor. Later, connecting to a Convolutional layer with 64 filters of size 3x3 pixels. The next layer is followed by a rectified linear operator (ReLU). Procedure mentioned above has been done twice in this block. Output of this block is 112x112x64

B. Second block

Second block is added with a Zero-padding layer, a Convolutional layer with 128 filters of size 3x3 pixels, followed by ReLU. Procedure mentioned above has been done twice in this block. At last, this layer is connected by a Max-pooling layer by extracting out the maximum value from 3x3 region with (2,2) stride. Output of this block is 56x56x128

C. Third block

Third block is added with an Zero-padding layer, a Convolutional layer with 256 filters of size 3x3 pixels, followed by ReLU. Procedure mentioned above has been done three times in this block. A Max-pooling layer is connected at the end of this block. Output of this block is 28x28x256

D. Fourth and Fifth block

Fourth block and fifth block are identical. Both are added with an Zero-padding layer, a Convolutional layer with 512 filters of size 3x3 pixels, followed by ReLU. Procedure mentioned above has been done three times in this block. At last, a Max-pooling layer is connected. Output of forth block and fifth block are 14x14x512 and 7x7x512 respectively.

E. Sixth block: Flatten layer and Fully Connected layer

4096 neurons are fully connected by the 25088 output neurons from a flattened layer, following by dropout 0.5. Identical two layers are connected again later. The output of the fully connected layer is input to softmax which its output is a normalized age value ranging from 0 to 1. Later, we use inverse function to map back to the original age value.

IV. Experiments

A. Training Methods

We train two kinds of models respectively. Details of each methods are as follow:

1) Method One: Clear Model+Blur Model: First training method we used is based on a fine-tuned pre-trained model for clear image recognition.

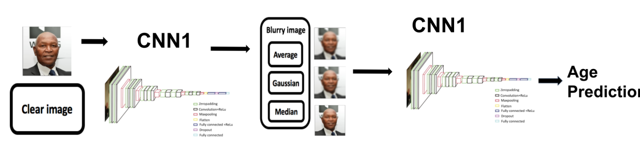

Fig. 7. Training Method One

2) Method Two: Blur Model: First training method we used is based on a fine-tuned pretrained model for clear image recognition. And we use 3 different type of blur method, including Average method, Gaussian method, and Median method.

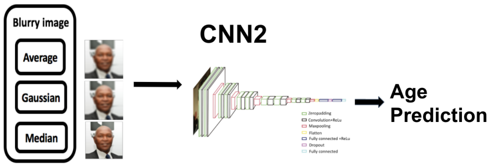

Fig. 8. Training Methods Two

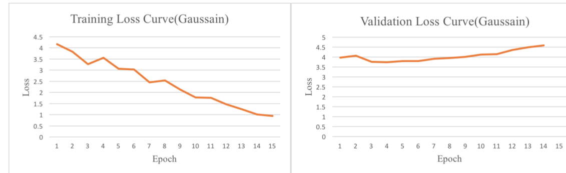

Fig. 11. Training Loss Curve(Gaussian)

B. Optimizing Methods

We have tried different approach to optimize our result, including changing batch sizes, learning rate, momentum, number of epochs, layer numbers, adding regularizer in fully connected layers (normalizing weights and output) and adding dropout layers.

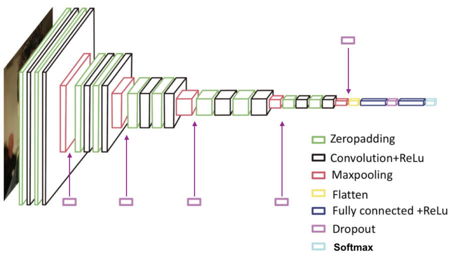

Fig. 9. Adding dropout layers

The optimal learning rate is 0.1/epoch with half decadence for every 20 iterations. Momentum is 0.9, neither higher nor lower than this value is better. With a view of preventing over-fitting, we add regularizer at the fully connected layers and dropout layers at the end of every block.

V. Preliminary Result

In our preliminary approach, we trained two models. One is implemented with method one, and the other is implemented with method two featured with Gaussian blur. Due that predicting the actual age precisely is difficult, we use loss instead of accuracy to evaluate our model performance. Predicted results are following:

A. Loss

We observed the descendence in training loss and the ascendance in validation loss in both clear and blur model. We presume that overfitting or vanishing gradient may occur. To get further information and discussion, we use confusion matrix instead of accuracy curve to analyze our testing result.

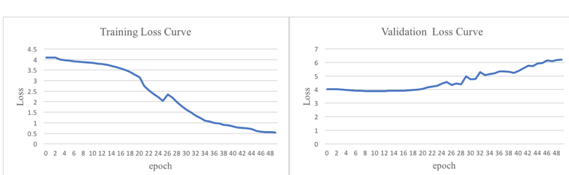

Fig. 10. Training Loss Curve(clear)

B. Confusion Matrix

After we check our 8000 data set , we found that most of the age lie between 20-40. The imbalanced data may be the reason to cause the model to predict age mostly in 20 to 40 years old. Since we checked the distribution of training data set, and found that the there are 57.5 percent of images of which age labels in 21-40.

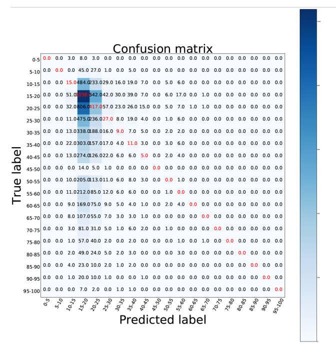

Fig. 12. Clear Model Confusion Matrix

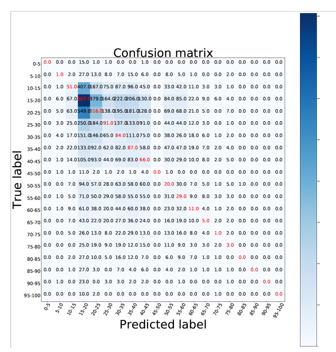

Fig. 13. Blur Model Confusion Matrix

TABLE II
Age labels distribution in training data

Range of age	Number of images
0-10	26
11-20	839
21-30	3105
31-40	1500
41-50	974
51-60	724
60+	832
Total	8000

VI. Conclusion

In this paper, though we firstly aimed to create six models and chose the best one which had most comprehensive ability in predicting age with blurring human images. However, due to limited resource, namely GPU usage, we failed to create six models and ended up with only two preliminary models which were trained by clean data set and Gaussian blur-processed data set. Nevertheless, we still learned lessons from this project, such as problem shooting, in specific, finding the imbalanced distribution of data set. In future, we will modify the distribution of data set and try to fit the models with data in batches to minimize the memory storage. Also, if possible, we would like to have a better GPU facility to facilitate our work. Last but not least, we are all grateful to have this chance to learn and implement Machine Learning knowledge in such a preliminary course. Knowing the high practicality of Machine Learning in such early stage will surly make impact on our future choice.

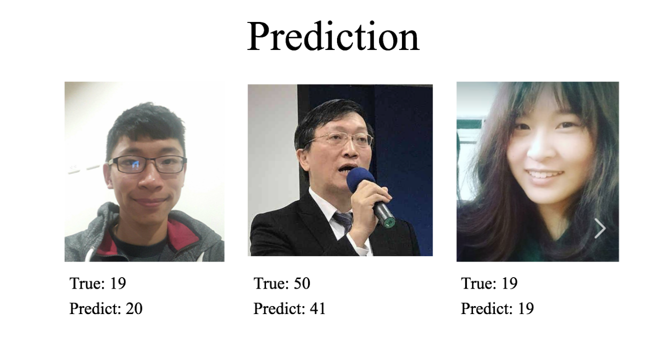

Fig. 14. Age Prediction of the Clear Model

Fig. 15. Age Prediction of the Blur Model