Model Initialization for Age Estimation

Jakkaphob Kongthanarith

Department of Engineering, Faculty of engineering, Kasetsart University

129/55 Perfect Place Traima, Thait 10600 Thailand

Assoc.Prof.Dr.Somying Thainimit

***Abstract* – It’s been a long time since Machine Learning method and Model Training was introduced to the world. There are various methods for using the trained model to complete the task. Each approach gives a different result and performance which leave us human to make an adjustment to reach the most ideal result, and initialization method is another way to complete the task.**

***Keywords* – Deep Learning, Model Training, CNN, Age Estimation, Machine Learning, Initialization**

I. Introduction

The process of model initialization is an alternative approach that has been extensively employed in training models. By carefully selecting the initialization method, the outcome of the model can be significantly varied. The main objective of this project is to carefully compare various initialization methods that will be applied to a Convolutional Neural Network (CNN) model, with a focus on age estimation. The goal is to find the initialization method that gives the best model performance for using in the age estimation task, considering its ability to affected on the model's performance and accuracy which will be determined using Mean Absolute Error (MAE). By performing an analysis and comparison of different initialization techniques within the CNN model, we can determine the initialization method that gives greatest results for accurately estimating age.

In this project, a total of X initialization methods on the same CNN model and observe each result for comparison. The data set will be imported and preprocessed before using them to train the model with different initialization methods applied.

II. Dataset

After browsing through the internet, the authors concluded that UTK Face from Kaggle website. It contained 23708 faces of Male and Female Human. All of them are straight faces which are fit to train with my Age Estimation model. The files have been arranged by their ages, gender, ethnicity, and file names respectively. the authors have a thought of splitting the Ethnicity and comparing with the initialization method I prepared but due to all datasets have been stored in a single folder with the only sorting factor is its file name. After spending over half a day, the authors have to abandon my thought on ethnicity and focus only on the age of the file inevitably. The shape of both train and test Data set is (None,224,224,3) with 25 image files for male and female in each age range 1-79 years old, reaching a total of 3950 image files.

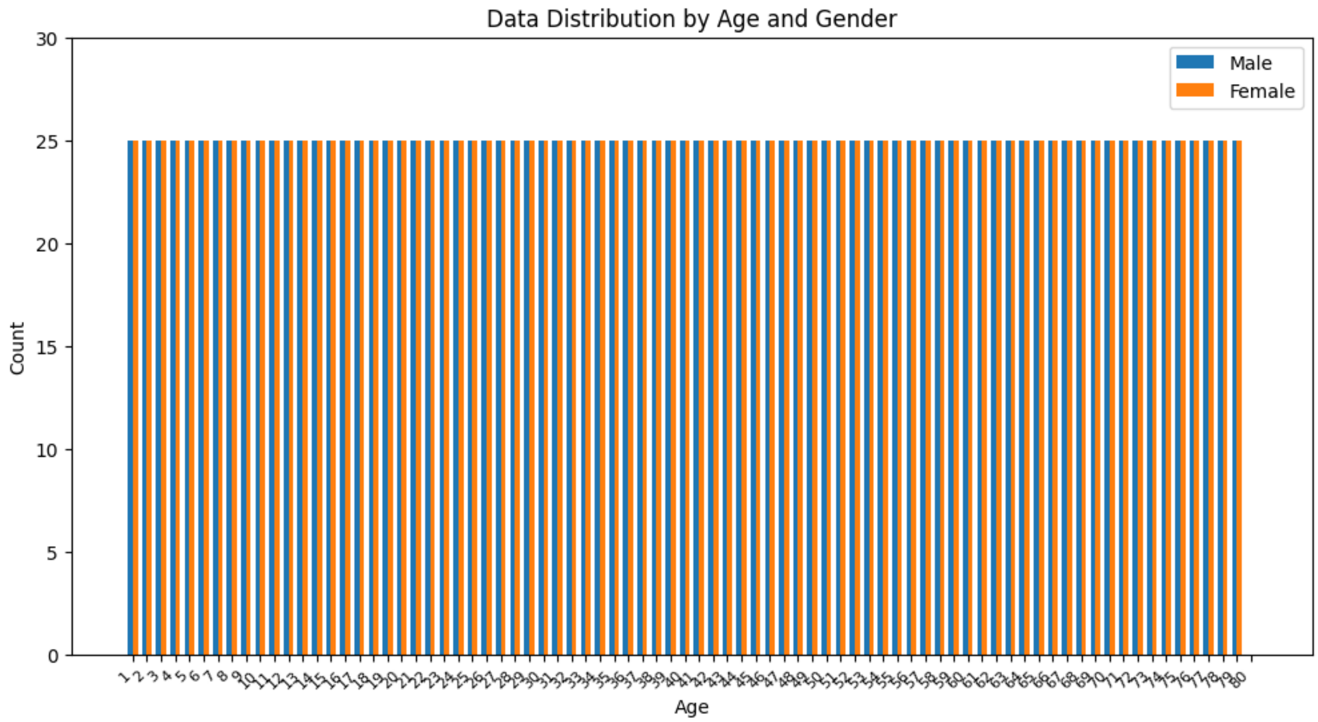


Fig 1. Dataset from UTK Face

III. Technique

A. Image augmentation

A technique commonly used in computer vision tasks to increase the diversity and size of the training dataset, resulting in decreasing the bias and improving the model learning rate by editing the image into a various pattern depend on the technique used.

Default Enhance Brightness Enhance Contrast

Fig 2. An augmentation example with original picture, picture with enhance brightness augmentation, picture with enhance contrast, consecutively.

The result (e.g. Fig 2.) show that to prevent the model confusion from training, the image that may not have a detailed or a clear straight face, Augmentation method that will be used in this project will be divided into 2 techniques, which is 'Enhance Brightness' and 'Enhance Contrast'. Enhance Brightness affects the overall brightness of an image. It increases or decreases the intensity of pixel values for the entire image, resulting in a brighter or darker appearance. Enhance Contrast adjusts the difference between the light and dark areas of an image, making the image appear more visually distinct. It affects the range of pixel values and enhances the level of detail and separation between different objects or features in the image.

B. Image Normalization

A process that changes the range of pixel's intensity into different values but contains the same ratio for other pixels in the images. The method for Normalization is to divide all pixel values in the image by 255. Doing this will result in better contrast detection of a model, lower over fitting problem, and compatibility of the model that require a fixed size of image.

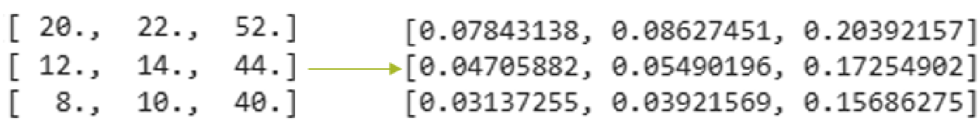


Fig 3. A part of pixel values before and after normalization

C. ReLU

A popular choice for activation function since it can erase the Vanishing-gradient problem, also easy to find the derivative since the output is only 0 or 1. ReLU may have a disadvantage such as return 0 if the input is negative or the output have no limit(0 to infinity) It may have a difficult time to solve this problem, but since the main problem like vanishing gradient that can't be solved once happen won't occur if using ReLU It considered to be a beneficial trade-off.

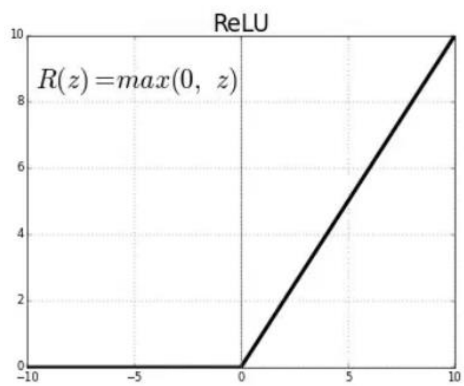


Figure 1 ReLU Activation Function
Reference: https://towardsdatascience.com/activation-functions-neural-networks-1cbd9f8d91d6

$$\text{ReLU Function: } f(x)=\max(0, x) = \begin{cases} 0 & \text{for } x \leq 0 \\ x & \text{for } x > 0 \end{cases} \tag{3}$$

$$\text{Derivative of ReLU Function: } f'(x) = \begin{cases} 0 & \text{for } x \leq 0 \\ 1 & \text{for } x > 0 \end{cases} \tag{4}$$

D. Model Initialization

1) Kernel Initialization: In convolutional layers, the kernel refers to the set of learnable filters that perform the convolution operation on the input data. Kernel initialization refers to setting the initial values of these filters. The choice of kernel initialization can impact the network's ability to learn useful features and its convergence speed.

1.1) Zero Initialization: This initializer sets all the weights to zero. It effectively removes any initial randomness from the weights, which can be problematic because it leads to symmetric gradients and identical updates during training. It is not recommended as it may cause the network to get stuck in a symmetrical state.

1.2) Glorot Normal Initialization: Glorot initializer draws the initial weights from a normal distribution with zero mean and a variance computed based on the number of input and output units of the layer. It is designed to keep the variance of the inputs and outputs roughly the same, which helps in maintaining stable gradients during training.

1.3) Glorot Uniform Initialization: Similar to Glorot Normal Initialization, this initializer draws the initial weights from a uniform distribution within a specified range. The range is determined based on the number of input and output units of the layer, ensuring the same variance preservation as in Glorot Normal Initialization.

1.4) He Normal Initialization: This initializer, also known as He Initialization, is similar to Glorot Normal Initialization but adjusts the variance based only on the number of input units. It is commonly used in networks with ReLU as activation functions.

1.5) He Uniform Initialization: Similar to He Normal Initialization, this initializer draws the initial weights from a uniform distribution within a specified range, based on the number of input units. It is useful when ReLU activations are used.

1.6) Random Normal Initialization: This initializer randomly initializes the weights from a normal distribution with a specified mean and standard deviation. It allows for more flexibility in initializing the weights but may require careful tuning of the mean and standard deviation values.

1.7) Random Uniform Initialization: This initializer randomly initializes the weights from a uniform distribution within a specified range. It provides more flexibility than the zero initializer but can still lead to initialization-related challenges.

2) Bias Initialization: Bias is an additional parameter associated with each neuron in a model. It allows the model to shift the activation function's output, introducing flexibility in the model's decision boundaries. Bias initialization refers to setting the initial values of these bias parameters.

2.1) Zero Initialization: the bias terms in a neural network layer are set to zero, making the layer's computations won't have any additional bias adjustment applied, and the layer will be initially biased towards producing output values close to zero. It is a commonly used approach, especially when the weights are initialized using other techniques, as it ensures no initial bias is present.

2.2) Constant Initialization: This initialization method requires the use to set a value for the bias for the initializer to sets all the biases to a constant value specified by the user set a specific value for the bias terms in a layer, resulting in potentially shifting the layer's output towards a specific value or pattern that you define. It can be useful when a specific bias value is desired, for example, to bias the initial predictions towards a particular class or outcome.

2.3) HE Normal Initialization: Main purpose of Bias HE initialization is to initialize the biases of a neural network layer using the He initialization technique, which scales the bias values based on the number of inputs to the layer. It helps prevent signal decay or explosion during forward.

2.4) HE Uniform Initialization: Bias HE Uniform initialization is a variant of the He initialization method that initializes biases using a uniform distribution. This random selection from a defined range adds diversity to bias values, allowing the neural network to explore different solutions during training. It can introduce additional randomness and is suitable when seeking a broader range of initial bias values.

2.5) Random Normal Initialization: This initializer initializes the biases from a normal distribution with a specified mean and standard deviation. Doing so will make all the biases to be set as a random value, making the layer's computations less deterministic and potentially enabling the model to explore different solutions during training.

2.6) Random Uniform Initialization: Similar to random normal initialization, random uniform initialization sets the bias terms to random values. The difference is those random values are drawn from a uniform distribution, meaning they will have a high chance to be in a specified range. It can introduce variability into the biases, promoting exploration and adaptability in the model's learning process.

E. Mean Absolute Error

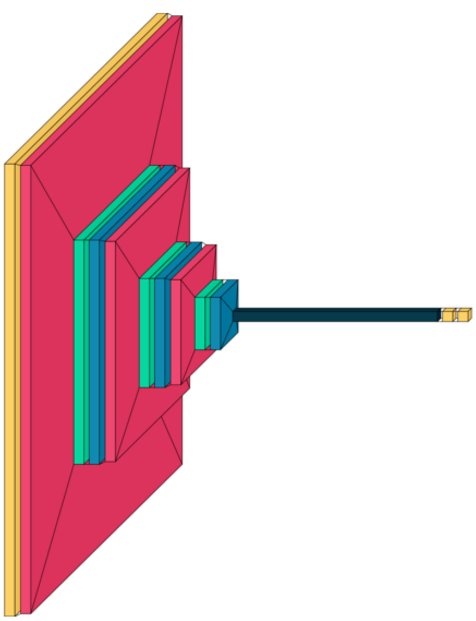
In short for MAE. It is a method used to evaluate the average difference between predicted values and the actual values. I used it to evaluate my age model performance because to the ages ranged is very difficult to pin-point the exact values dues to various factor such as Similar face structure of babies, those who have their faces not developed along with their ages, achene, or stain on their faces, etc. Thus, I will use the MAE to indicate the overall difference of the prediction result of my model and the actual ages of the Dataset.

$$MAE = \frac{1}{n} \sum_{i=1}^{n} |Xi - X| \tag{5}$$

X_i and X can be swapped since after differentiation of each, the value will be converted into positive value before being kept in the formula.

IV. Core Model

The model used as a base is CNN Model with Linear Activation as an output. The initialization methods will be applied at each Conv2D Layer. After performing one initialization method and evaluating the model, we will switch to the another Colab tab and running a model with different kind of initialization method so their won't be any variable that will affect the result.



Layer (Type)	Output Shape
Input Layer (224,224,3)	(224,224,3)
Conv2D (filter = 32), (kernelize = 3*3)	(222,222,32)
MaxPooling2D (2*2)	(111,111,32)
Dropout (0.5)	(111,111,32)
Conv2D (filter = 64), Kernel size = 3*3)	(109,109,64)
MaxPooling2D (2*2)	(54,54,64)
Dropout (0.5)	(54,54,64)
Conv2D (filter size = 128)	(52,52,128)
MaxPooling2D (2*2)	(26,26,128)
Dropout (0.5)	(226,226,128)
Flatten	(86528)
Dense (filter = 64)	(64)
Output Layer (activation = linear)	(1)

Figure 5. CNN model visualization

V. Result and discussion

The Evaluation method used will be Mean absolute error and loss after training the model from Fig 5 with each different Initialization method.

Table 1.

Evaluation for each method that belongs to the Kernel Initializer

Initialization method	**Train MAE**	**Test MAE**	**Train Loss**	**Test Loss**
Glorot_uniform (Default Method for CNN)	8.5246	13.3773	119.6415	278.9876
Zero	28.8623	28.0063	1240.9595	1167.8342
Glorot_normal	9.5571	13.7887	150.2741	289.0040
He_normal	28.7123	29.2987	1227.6096	1275.9395
He_uniform	9.2964	20.0871	137.044	606.2070
Random_normal	28.6285	28.9267	1222.9497	1238.9324
Random_uniform	8.7701	14.0576	125.3125	329.7053

Table 2.

Evaluation for each method that belongs to the Bias Initializer

Initialization method	**Train MAE**	**Test MAE**	**Train Loss**	**Test Loss**
Zero (Default)	12.2524	16.4194	231.6260	397.4644
Constant (Manually adjusted) (bias = 0.1)	13.8253	22.1904	286.2624	706.7732
HE_normal	37.5159	37.8965	1951.6508	1968.0397
HE_Uniform	36.4908	35.4343	1838.8444	1774.4475
Random_normal	10.163	14.8020	173.1014	344.1577
Random_uniform	11.6425	14.2584	212.9210	312.1348

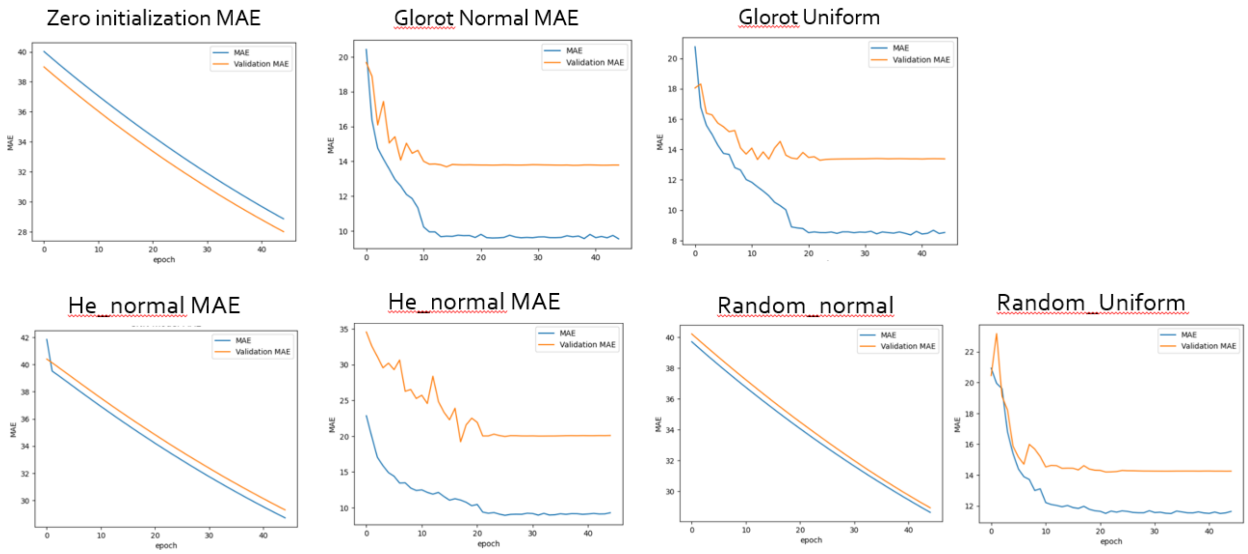


Figure 6. Kernel initialization MAE evaluation graph

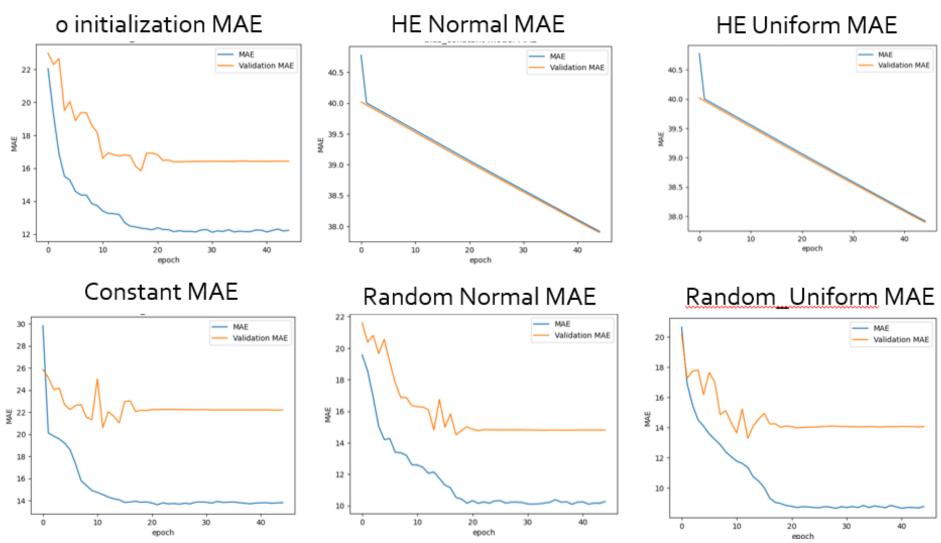


Figure 7. Bias initialization MAE evaluation graph

Bias will adjust the bias of the model as of its name, but since my data set is being equally distributed, adjusting bias only return the worsen result. Also, both Glorot Normal and Unifor are not available to use in Bias initialization method.

When comparing the result of two methods, all results from bias are worse than weight, this means that in my case for age estimation, adjusting the bias will only worsen the result.

The Initialization method with best performance is Kernel Glorot Uniform or the default initialization method. Since it helps manage gradient flow by keeping the variance of activations and gradients consistent across layers, leading to more stable training and also help the model to solve vanishing gradient problem. Glorot Uniform also gives a better generalization and learning capacity by keeping neurons receive inputs of similar magnitudes, enhancing the model's ability to learn meaningful features.

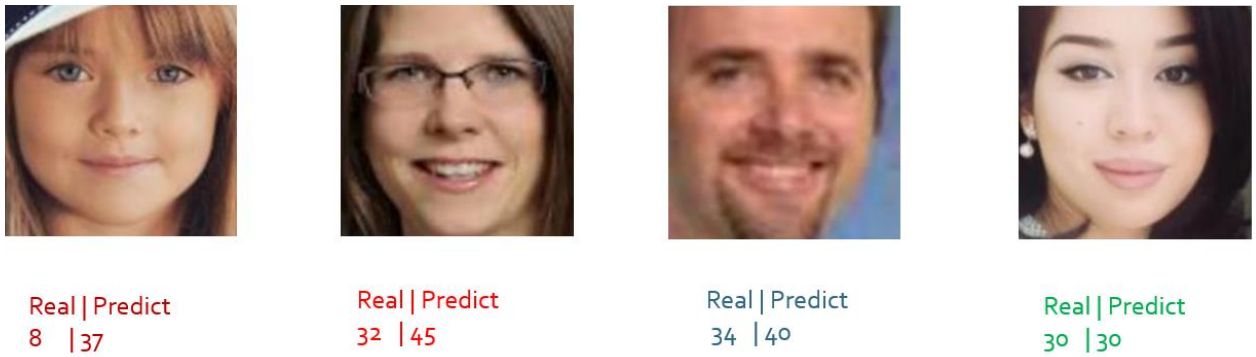


Figure 8. Prediction result example

VI. Conclusion

Even though Table 1 and Table 2 Best Performance is Kernal Glorot Uniform initialization, the loss and MAE result is very bad, initialization method might be related to model performance just for a bit. Almost half of the model states that it can be used and able to improve the model performance if applied together with ReLU, this made the author question my own designed CNN model or dataset. From Figure 8. The prediction of a baby to young age are often incorrect, this might be the case where feature of the young human face is constantly progress at a rapid speed when the time flows, resulting in insufficient sample for the model to fin, recognize and learn the image pattern. The author might have to split the data set of children from adults and train them separately to observe the result. In order to make the idea come true, further fine-tuning and model adjustment must also be performed case by case.

VII Acknowledgement

I would like to express my sincere gratitude to my Professor, Assoc.Prof.Dr.Somying Thainimit. Her words of advice, resource, comment, and kindness make me able to make it this far. Truly, thank you.

VIII Reference

[1] Kamekshi Gaur, Dr. Pawan Kumar “Literature Review on Gender Prediction Model using CNN Algorithm” Internet: https://www.irjet.net/archives/V9/i3/IRJET-V9I356.pdf , Mar 2022

[2] Syed Taskeen Rahman, Asiful Arefeen, Shashoto Sharif Mridul, Asir Intisar Khan, Samia Subrina “Human Age and Gender Estimation using Facial Image Processing” Internet: https://ieeexplore.ieee.org/document/9230933, January 2020

[3] Koichi Ito, Hiroya Kawai, Takehisa Okano, Takafumi Aoki “Age and Gender Prediction from Face Images Using Convolutional Neural Network” Internet: https://ieeexplore.ieee.org/abstract/document/8659655, 07 March 2019, Publicher: IEEE, DOI: 10.23919/APSIPA.2018.8659655

[4] Chenjing Yan, Congyan Lang, Tao Wang, Xuetao Du & Chen Zhang " Age Estimation Based on Convolutional Neural Network” Internet: https://link.springer.com/chapter/10.1007/978-3-319-13168-9_22, PCM 2014, pp 211–220

[5] Jun-Cheng Chen, Amit Kumar, Rajeev Ranjan, Vishal M. Patel “A cascaded convolutional neural network for age estimation of unconstrained faces” Internet: https://ieeexplore.ieee.org/abstract/document/7791154, December 2016, DOI: 10.1109/BTAS.2016.7791154, Publisher: IEEE

[6] Meenal Narkhede, Prashant P. Bartakke, Prashant P. Bartakke “A review on weight initialization strategies for neural networks” Internet: https://www.researchgate.net/publication/352813104_A_review_on_weight_initialization_strategies_for_neural_networks, January 2022 10.1007/s10462-021-10033-z