Age Estimation: A Convolutional Neural Network Based Regression Approach

*Proposed to: Hamdi Dibeklioglu*

Emre Doğan   
*Department of Computer Engineering*  
*Bilkent University*Ankara, Turkey  
[emre.dogan@bilkent.edu.tr](mailto:emre.dogan@bilkent.edu.tr)   
Hamdi Alperen Çetin  
*Department of Computer Engineering*  
*Bilkent University*Ankara, Turkey  
[alperen.cetin@bilkent.edu.tr](mailto:alperen.cetin@bilkent.edu.tr)

*Abstract*—This document is the technical report of CS559 Deep Learning Course Homework. The main purpose is to build a CNN (*Convolutional Neural Network*) based regression model to accurately estimate ages of people from their facial images.

Keywords—convolutional neural networks, age estimation, regression

# Introduction

In the last decade, Convolutional Neural Network (CNN) has achieved promising results in computer vision tasks due to its self-learning power and ability of working with very large data. In our task, we employ CNN models to estimate human ages from given their face images. Different types of CNN architectures are employed and compared in our study. Another important point regarding deep network models is the process of validating hyperparameters. We applied a set of values for each hyperparameter to result with the best model.

This paper is organized as follows. Section 2 describes the background and previous studies that we took advantage of. Section 3 gives brief information about our dataset. Section 4 describes our model. Section 5 gives the implementation details. In section 6, experiments regarding hyperparameters of the model are shared. Section 7 describes the performance of our finally tuned model. Finally, Section 8 presents our conclusions.

# Background

With the recent developments for last 10 years, deep learning has achieved a success in most of the computer vision tasks. This is due to the fact that deep learning can deal with a large amount of data and explore it by itself.

# Dataset

With In our study, we used a modified version of UTKFace Dataset[[1]](#footnote-1). The original dataset includes samples of people with an age range from 0 to 116 years old. It also has some additional information such as age, gender, and ethnicity. But for our case, the dataset includes people aged in a range of 0 to 80 with no additional information.

Our dataset consists of 5400 training, 2315 validation and 1159 test samples. The histogram of age distribution in our dataset is available in Figure 1.

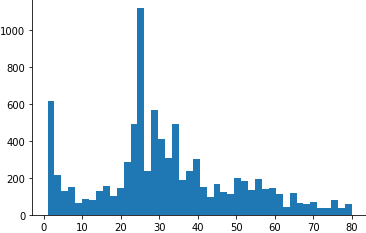


Figure 1. Histogram of Age Distribution in Our Data

# Our Model

Through this homework, we implemented different architectures to observe differences in terms of performance. First, we started with the standard

We start with convolutions then continue with fully connected layers. To keep the number of parameters in a reasonable range, we need to reduce the parameters before fully connected layers.

There are three different mostly used ways to decrease the number parameters: pooling, 1x1 convolution with less filters and convolution with an enough amount of stride.

For example, let's say we have a layer with 40x40 dimensions. Applying 2x2 max pooling with stride 2 or applying 8 filters 1x1 convolution with stride 1 will result with a 20x20x32 layer. Using 32 filters 5x5 convolution with stride 2 will produce a 19x19x32 layer. At the end, all of them will end up with a layer having less parameters.

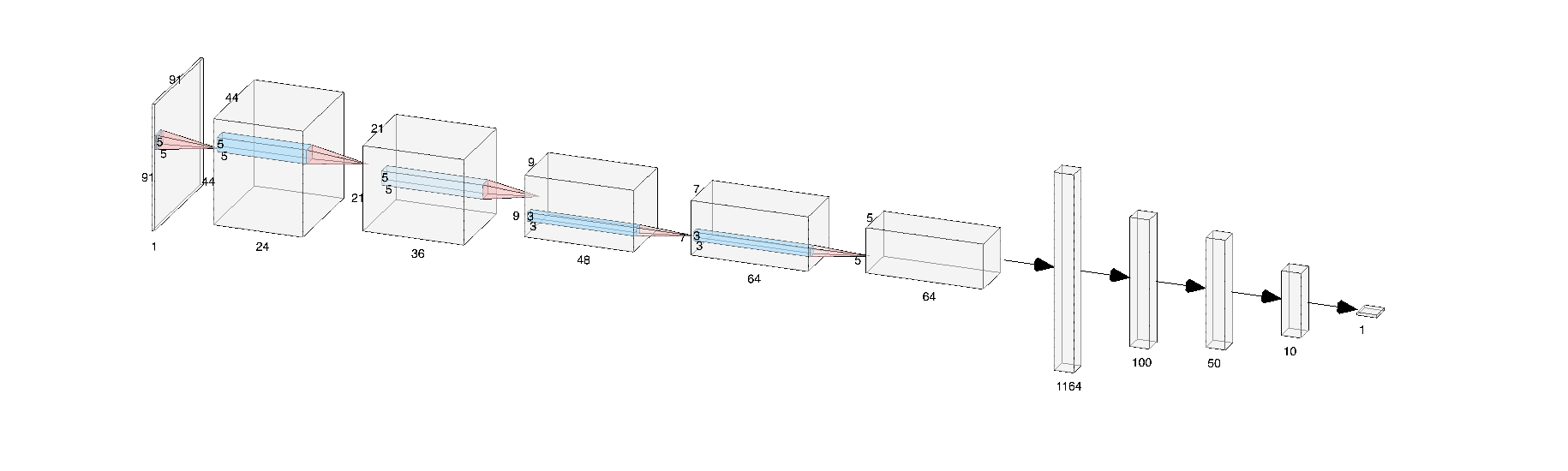


Figure 2. Structure of Our Model

In our model, we employ the last way by using 5x5 convolutions with stride 2 at the beginning and using 3x3 convolutions with stride 1 towards the end of convolutions. Figure 2 shows the overall architecture of our model.

In our first model, there is a max-pool layer after each con2d layer. But in this model, we do not use any max-pool operation at all. While investigating different studies on CNN architectures, we noticed that some studies do not use max-pooling layer, or any other pooling methods. In [1], Kuen et al. proposed a new idea ‘*DelugeNets*’ in which information across many layers is propagated with greater flexibility compared to ResNets by using convolution with stride instead of pooling. Similarly, Springenberg et al.[2] question the necessity of different components in the pipeline and find that max-pooling can simply be replaced by a convolutional layer with increased stride without loss of accuracy. With an inspiration from these studies, we replaced pooling layers with convolution layers. By adjusting stride values of convolution layers, reduction of input size through the network can be done, similar to max-pooling.

# Experıments

As required in the description of homework, there are some methods & hyperparameters to explore so that we can improve our model. Effects of each improvement will be discussed in this section step by step.

## Batch Normalization

Batch normalization is the idea of normalizing layer values just like input values. Beyond its contribution to training speed, it helps to solve two important problems regarding deep networks:

1. Vanishing & Exploding Gradient:

Due to very large and very small weight values, the signal proceeding through the network may converge to 0 or diverge through infinity. By normalizing each layer, we avoid to have such kind of a problem.

1. Covariance Shift:

It is simply the bias of the model through the distribution of input features. If an algorithm learns a mapping from X to Y, the model may fail when a new X with different distribution of features is tested.

Beyond these two problems, batch normalization also;

* reduces overfitting
* lets us to use higher learning rates
* allows each layer to learn more independent of other layers.

We conducted an experiment to observe the effect of BatchNorm for our model. Other parameters are kept constant and given below so that the effect of batch normalization can be observed in a better way:

learning rate :0.01

batch size = 100

number of epochs = 100

no drop-out or L2 regularization

Mean Average Error (MAE) values for models with and without batchnorm layers can be observed in Figure 3.

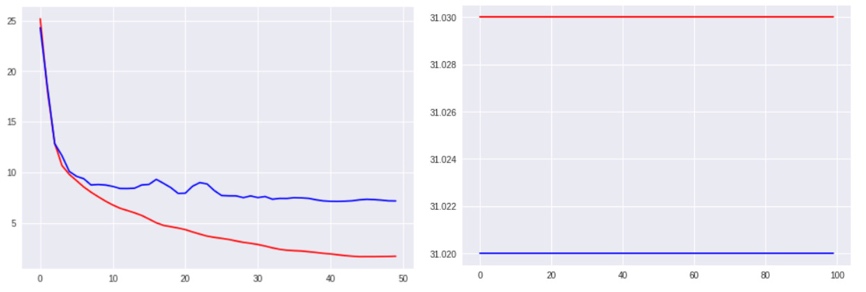


Figure 3. MAE Graphs of the model using batch normalization (left) and the model not using batch normalization (right)

ALPEREN YORUM 11!!!1!!!

## Adam Optimizer & Early Stopping

## Loss Function

To be able to make a controlled experiment successfully, other hyperparameters such as learning rate, batch size and methods such as regularization, batch normalization are kept constant.

As a loss function, we used Mean Absolute Error, Mean Square Error and Huber Loss. The resulting error graphs for these 3 different losses are given in Figure 4.

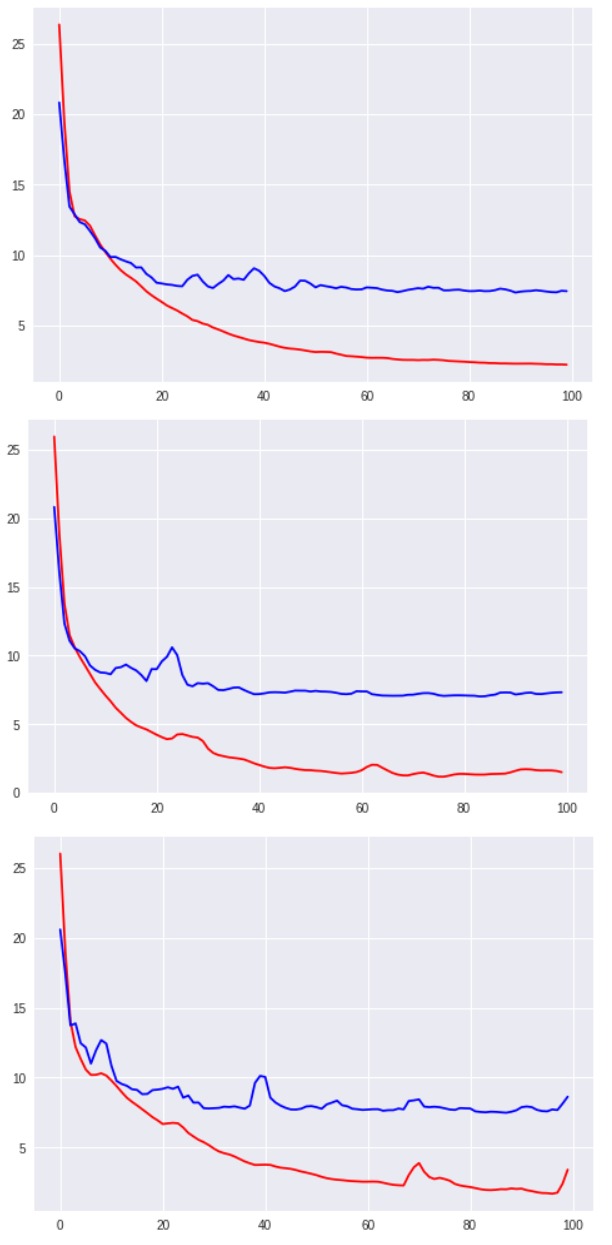


Figure 4. Comparison of errors 3 different loss functions used in our model:MAE (upper), MSE(middle), Huber Loss(lower).

In MSE and Huber Loss, any error in estimation is punished with its square which in our case may not be a good option. For example, ±2 error in age estimation is not a too bad option. But in these loss functions, this error is punished with its square . For this reason, we decided to use MAE as our loss function.

## Initialization Type

To be commented.

Figure 5

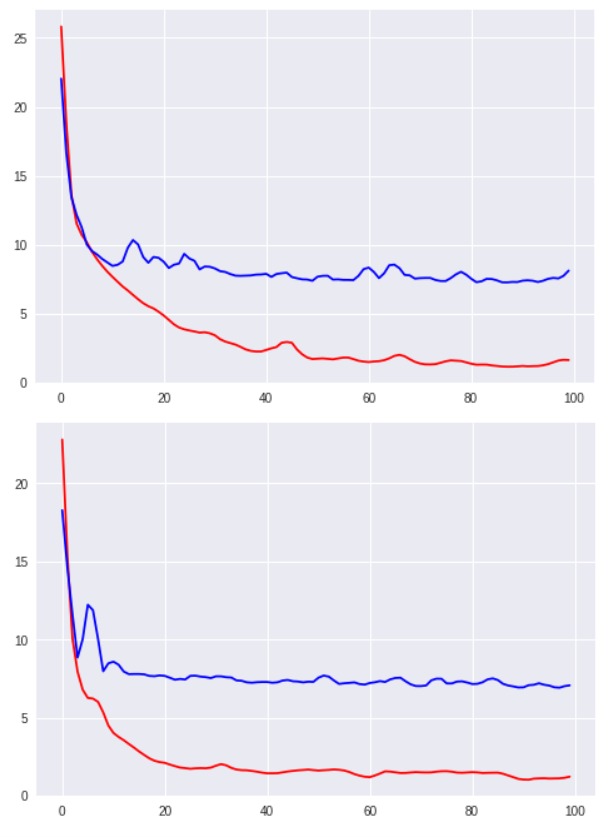


Figure 5

## Regularization (Dropout & L2)

* ***Dropout:***

Validation and test MAE values for different dropout keep probabilities are available in Table 1.

Mean square error

learning rate :0.01

batch\_size = 100

filter coef = 2

To fill here.

Table 1. MAE Values for Different Dropout Keep Probabilities.

|  |  |  |
| --- | --- | --- |
| Keep Probability | Validation MAE | Test MAE |
| 0.4 | 6.93 | 7.05 |
| 0.5 | 6.70 | 6.86 |
| 0.6 | 6.76 | 7.02 |
| 0.7 | 6.83 | 6.91 |
| 0.8 | 6.63 | 6.70 |
| 0.9 | 7.01 | 7.07 |

Reference and explain Figure 6.

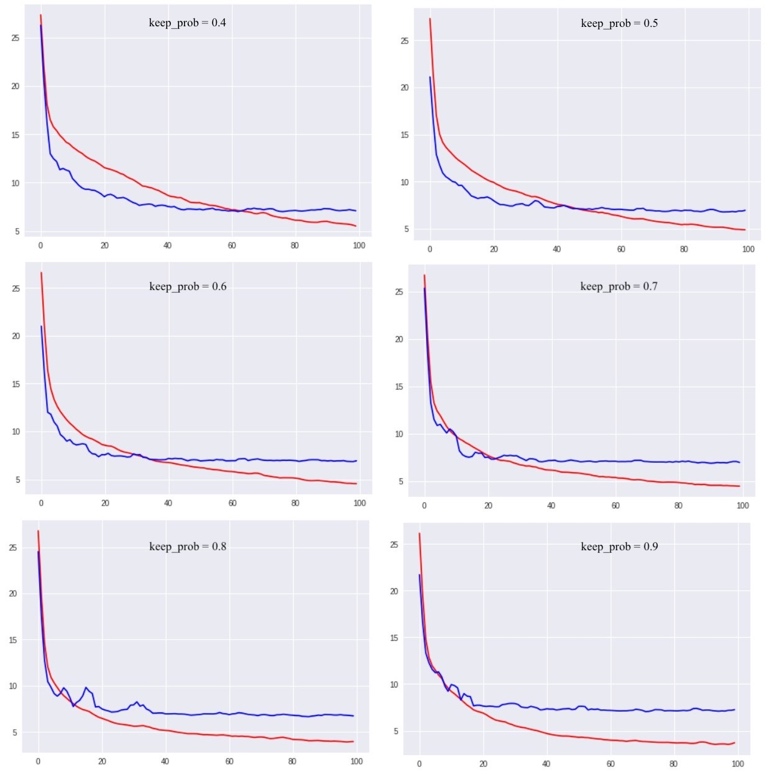


Figure 6. MAE Values for Different Dropout Keep Probabilities.

* ***L2 Regularization:***

Mean square error

learning rate :0.01

batch\_size = 100

filter coef = 2

dropout keep:0.8

Reference the Figure 7.

Didn’t put MAE table. Because main constraint is the stability, not mAE values.

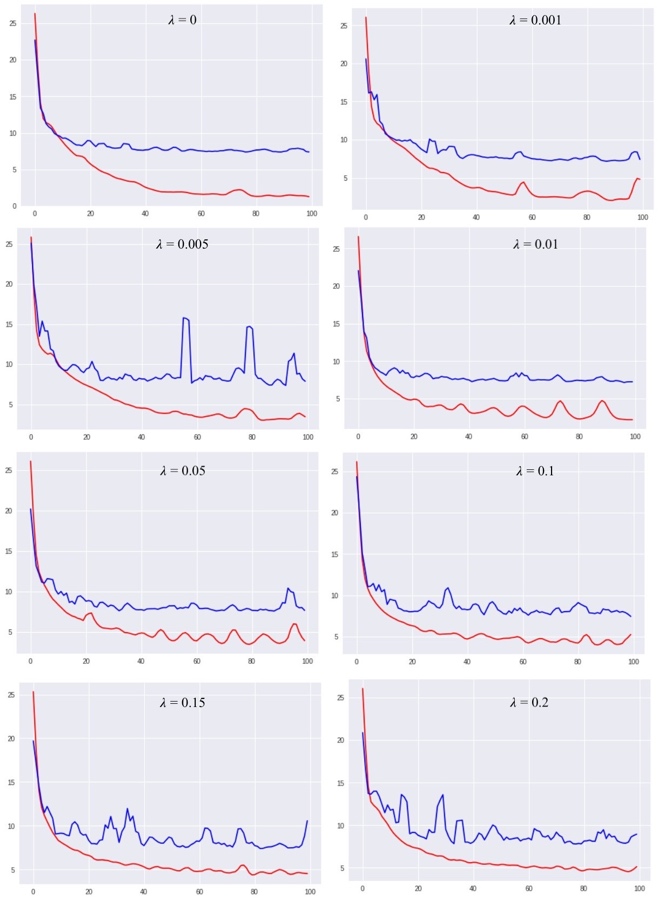


Figure 7. MAE Values for Different L2 constants.

# Conclusıon

Ssss

1. Table Type Styles

| Table Head | Table Column Head | | |
| --- | --- | --- | --- |
| Table column subhead | Subhead | Subhead |
| copy | More table copya |  |  |

1. Sample of a Table footnote. (*Table footnote*)
2. Example of a figure caption. (*figure caption*)

##### Acknowledgment *(Heading 5)*

The preferred spelling of the word “acknowledgment” in America is without an “e” after the “g”. Avoid the stilted expression “one of us (R. B. G.) thanks ...”. Instead, try “R. B. G. thanks...”. Put sponsor acknowledgments in the unnumbered footnote on the first page.

##### References

[1] J. Kuen, X. Kong, G. Wang, and Y. P. Tan, “DelugeNets: Deep Networks with Efficient and Flexible Cross-Layer Information Inflows,” *Proc. - 2017 IEEE Int. Conf. Comput. Vis. Work. ICCVW 2017*, vol. 2018–January, pp. 958–966, 2018.

[2] J. T. Springenberg, A. Dosovitskiy, T. Brox, and M. Riedmiller, “Striving for Simplicity: The All Convolutional Net,” pp. 1–14, 2014.

The template will number citations consecutively within brackets [1]. The sentence punctuation follows the bracket [2]. Refer simply to the reference number, as in [3]—do not use “Ref. [3]” or “reference [3]” except at the beginning of a sentence: “Reference [3] was the first ...”

Number footnotes separately in superscripts. Place the actual footnote at the bottom of the column in which it was cited. Do not put footnotes in the abstract or reference list. Use letters for table footnotes.

Unless there are six authors or more give all authors’ names; do not use “et al.”. Papers that have not been published, even if they have been submitted for publication, should be cited as “unpublished” [4]. Papers that have been accepted for publication should be cited as “in press” [5]. Capitalize only the first word in a paper title, except for proper nouns and element symbols.

For papers published in translation journals, please give the English citation first, followed by the original foreign-language citation [6].

1. G. Eason, B. Noble, and I. N. Sneddon, “On certain integrals of Lipschitz-Hankel type involving products of Bessel functions,” Phil. Trans. Roy. Soc. London, vol. A247, pp. 529–551, April 1955. *(references)*
2. J. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68–73.
3. I. S. Jacobs and C. P. Bean, “Fine particles, thin films and exchange anisotropy,” in Magnetism, vol. III, G. T. Rado and H. Suhl, Eds. New York: Academic, 1963, pp. 271–350.
4. K. Elissa, “Title of paper if known,” unpublished.
5. R. Nicole, “Title of paper with only first word capitalized,” J. Name Stand. Abbrev., in press.
6. Y. Yorozu, M. Hirano, K. Oka, and Y. Tagawa, “Electron spectroscopy studies on magneto-optical media and plastic substrate interface,” IEEE Transl. J. Magn. Japan, vol. 2, pp. 740–741, August 1987 [Digests 9th Annual Conf. Magnetics Japan, p. 301, 1982].
7. M. Young, The Technical Writer’s Handbook. Mill Valley, CA: University Science, 1989.

1. <http://aicip.eecs.utk.edu/wiki/UTKFace> [↑](#footnote-ref-1)