2022-2023 FALL

CS-464

Age Detection Using Convolutional Neural Networks

Final Report for Term Group Project

Team Number: 15

Instructor: Asst. Prof. Ayşegül Dündar

Team Members:

1) Kaan Hamurcu, 21902012, EEE
2) Efser Efe Kantik, 21901418, EEE
3) Cemhan Kaan Özaltan, 21902695, CS
4) Ömer Özaltan, 21902973, EEE
5) Ahmet Batuhan Sancak, 21902087, EEE

I. Introduction

Age classification is a challenging task, even for humans. People age at different rates and aging depends on some factors such as race, diet, lifestyle or even occupation. By looking at many examples of people with a variety of races, occupations, etc. meanwhile possessing the knowledge of their ages, a pattern may be detected among the divergent group of people. When it comes to inspecting many examples and catching a pattern, a convolutional neural network (CNN) model can be a viable option for accomplishing such a task. A CNN with transfer learning weights obtained from ResNet-18 will also be subjected to the age classification task. Their performances will be analyzed using traditional performance metrics and an unique metric taking notice of task-specific constraints.

II. Background Information

The goal of this project was to predict the age interval for an individual from his or her face. To begin with, we determined that the problem was a classification problem rather than a regression task. Since it was mentioned before, people age at different rates and it can prove to be quite challenging to predict an age right on the nail. For instance, the facial appearance of a person who is 27 would not differ much from their appearance when they were 25. Therefore devising intervals to assign individuals rather than opting for precise age predictions will be a more feasible approach.

We have decided to employ a convolutional neural network model to achieve our task. Convolutional neural networks are very competent in reducing the number of parameters existent in the input [1], which is also the case for our task. An image possesses pixels in the scale of thousands. Considering the fact that each pixel is considered a feature, training a model which would pass this data from multiple layers would be computationally very heavy.

Finally, we will use a convolutional neural network utilizing transfer learning weights to attain a comparable model. We expect to have the performance of CNN utilizing transfer learning better than CNN trained from scratch accordingly with its number of pre-trained parameters.

III. Methods

While implementing our project, we used convolutional neural networks (CNNs) along with transfer learning. Our network consists of 6 convolution blocks, where each block contains a convolution layer, a rectified linear unit (ReLU) activation layer, and a max-pooling layer, which are all connected through a forward pass that takes an image input, and outputs the confidence of each class as output layer neuron activations. The channel sizes of our convolution layers are 16, 32, 64, 128, 256, and 512 respectively. We used 3x3 kernels for convolution operations. Stride and padding values are both 1. We used a batch size of 32 in order to find the balance between memory utilization and execution time. We used a learning rate of 0.01 and a weight decay of 0.0005. We used the categorical cross-entropy loss function along with the stochastic gradient descent (SGD) optimizer. The listed hyperparameters were tuned using the results of the validation dataset.

Our dataset comprises the “facial age” dataset [2] and the “UTK Face” dataset [3]. They contain 9778 and 23,708 colored images respectively. The images in both of these datasets are of human faces and are composed of 200x200 pixels. Moreover, all of them are labeled with ages ranging from 1 to 110 and 1 to 116, respectively. The ages are mostly balanced throughout this interval except for a spike at

age 26, where there is an intensity of available samples. Furthermore, the concentration of samples is lower for ages larger than 65.

According to Mustapha et al., intervals in age classification using a set of ranges should be as follows: 2-3, 4-6, 7-8, 9-11, 12-20, 21-35, 36-50, and 51-80; which correspond to “infancy, early childhood, middle childhood, late childhood, adolescence, early adulthood, midlife, and mature adulthood” respectively [4]. Since we used an extended dataset, our project included samples outside these ranges. Therefore, we increased the size of the first interval and added a new interval for ages between 81 and 116. Our final interval ranges (classes) are as follows. Interval ranges are inclusive with integer elements:

Class Index	**Interval Start**	**Interval End**
0	1	3
1	4	6
2	7	8
3	9	11
4	12	20
5	21	35
6	36	50
7	51	80
8	81	116

Table 1: Dataset classes

We also used transfer learning with the pretrained ResNet-18 model, which has 11.7 million already trained learnable parameters and it is a well known image processing neural network model [5]. We will also compare the performance with and without transfer learning in the following sections.

IV. Experiments

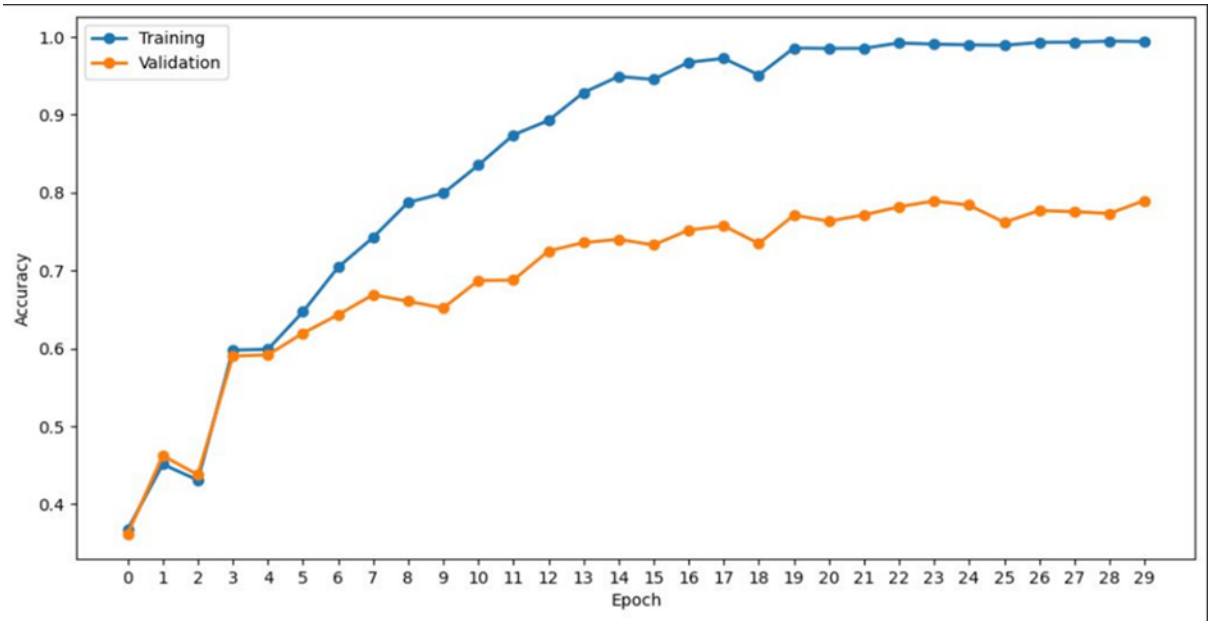

Figure 1: The accuracy vs epoch plot for the CNN model

Figure 1 shows the accuracy versus epoch plot for the CNN model which was trained from scratch. It can be observed that the accuracies of both the training set and the validation set start to flatten after around the 15th epoch. Even though the two accuracies remain close to each other for the first five epochs, the training set accuracy surpasses the validation accuracy at the later epochs. It eventually reaches almost 1 while the validation accuracy fluctuates around 0.75.

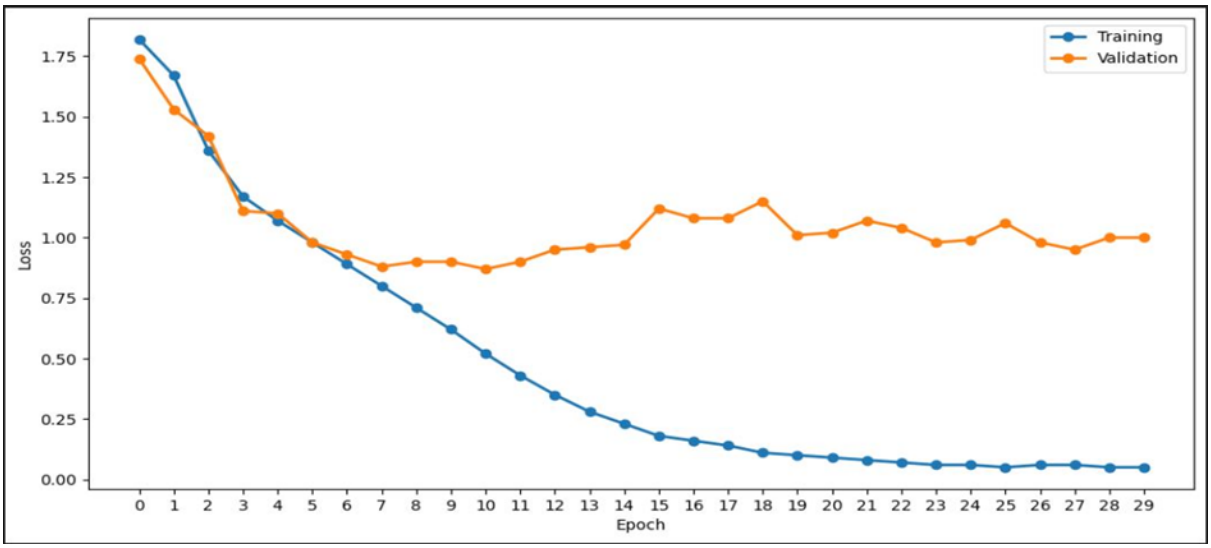

Figure 2: The loss vs epoch plot for the CNN model

Figure 2 shows that the training loss starts to plateau only after around the 25th epoch. On the other hand, the validation loss flattens much earlier, at around the 8th epoch. As was the case with the accuracies, the two losses are close in value for the first five epochs. Afterward, the training loss

experiences a rapid decay while the validation loss does not change much. By the end of the training, the training loss at around 0.1, is much smaller than the validation loss which is around 1.

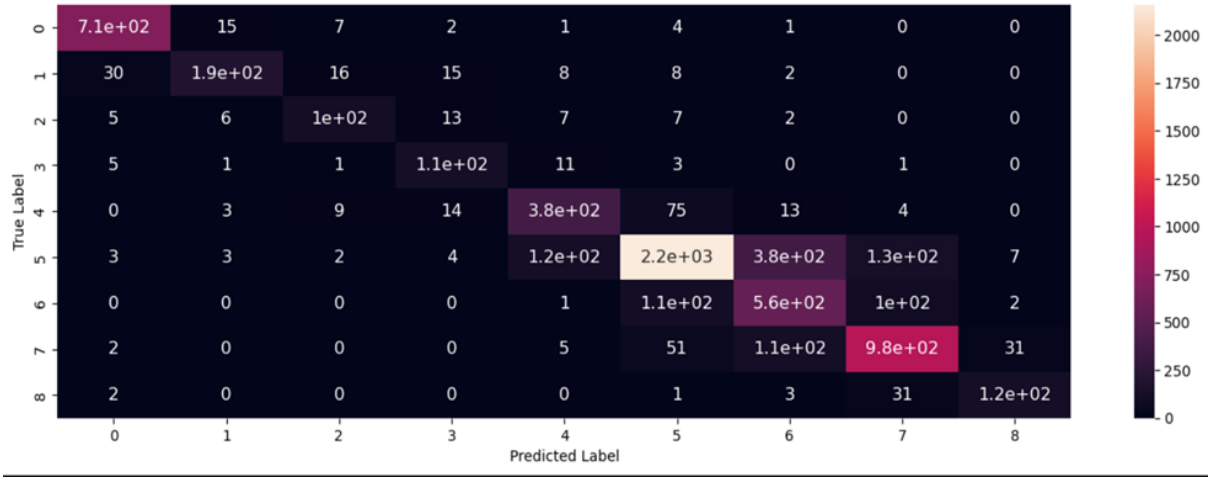

Figure 3: The confusion matrix of the CNN model

After achieving reasonable validation accuracies, the trained CNN model was tested on the test set. The confusion matrix was obtained as given in Figure 3. The accuracy was found as 79.3%. Furthermore, the macro precision, recall and F1 scores were determined to be 0.785, 0.768, and 0.772 respectively.

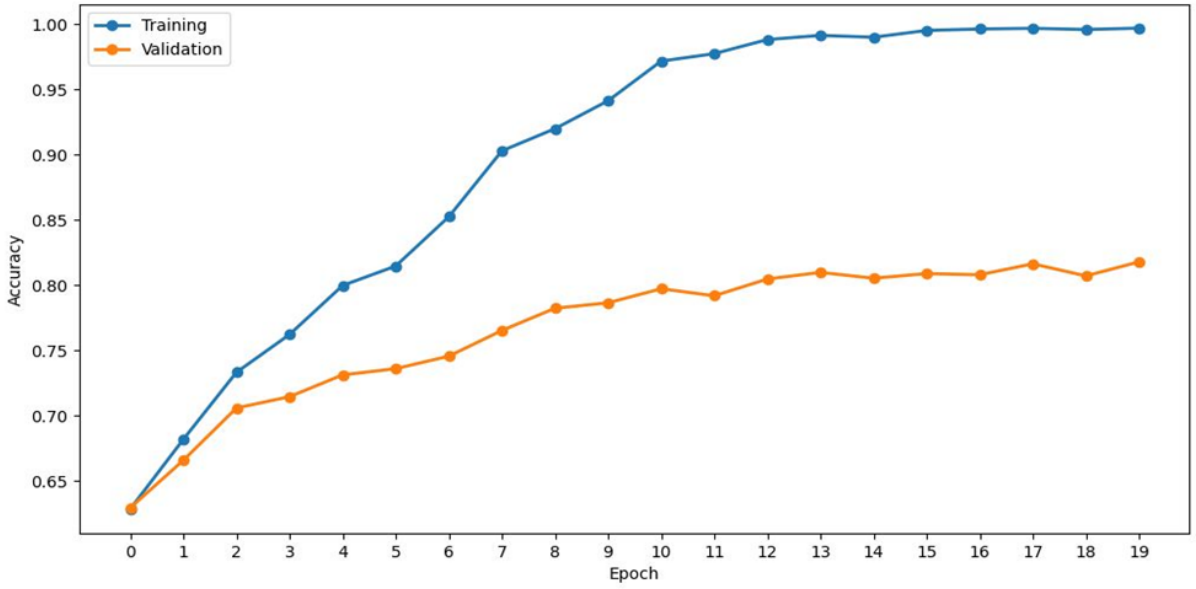

Figure 4: The accuracy vs epoch plot for the transfer learning model

The accuracy versus epoch plot for the transfer learning model was obtained as shown in Figure 4. Both the training and validation accuracies start to flatten at around the 12th epoch. Although they are close in the first three epochs, the training accuracy is higher than the validation accuracy at all epochs. The training accuracy reaches almost 1 while the final validation accuracy is around 0.80.

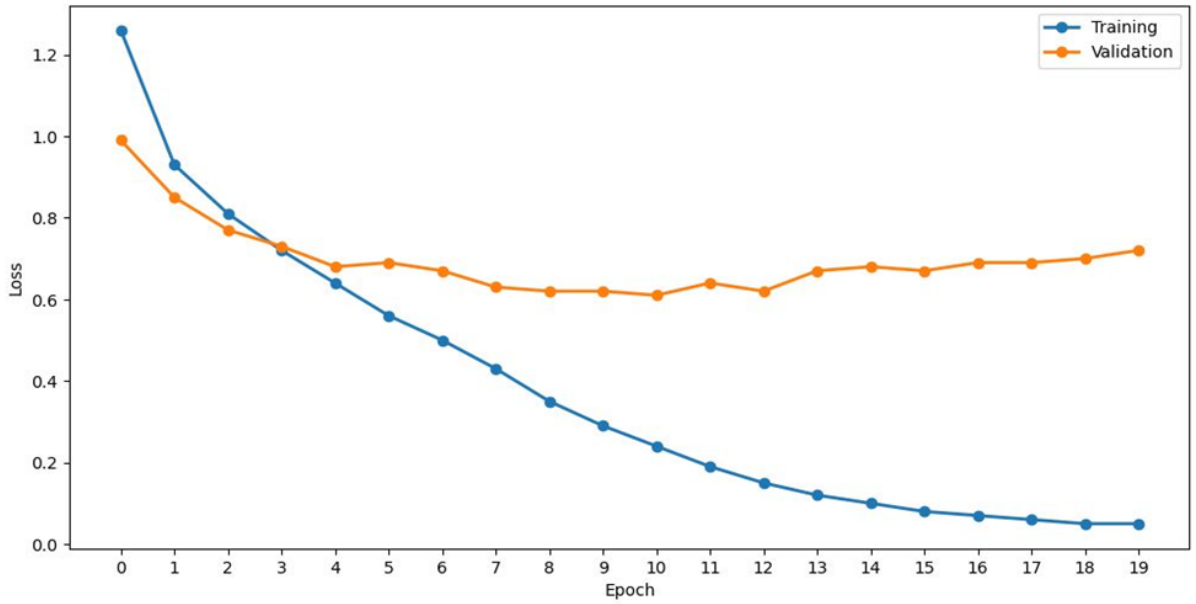

Figure 5: The loss vs epoch plot for the transfer learning model

Figure 5 shows that the training loss starts to flatten only towards the end. In contrast, the validation loss becomes flat after around the 8th epoch. Even though the training loss is larger than the validation loss in the beginning, it decreases rapidly while the validation loss does not improve much from its initial value. The final training and validation losses were observed to be roughly 0.1 and 0.7 respectively.

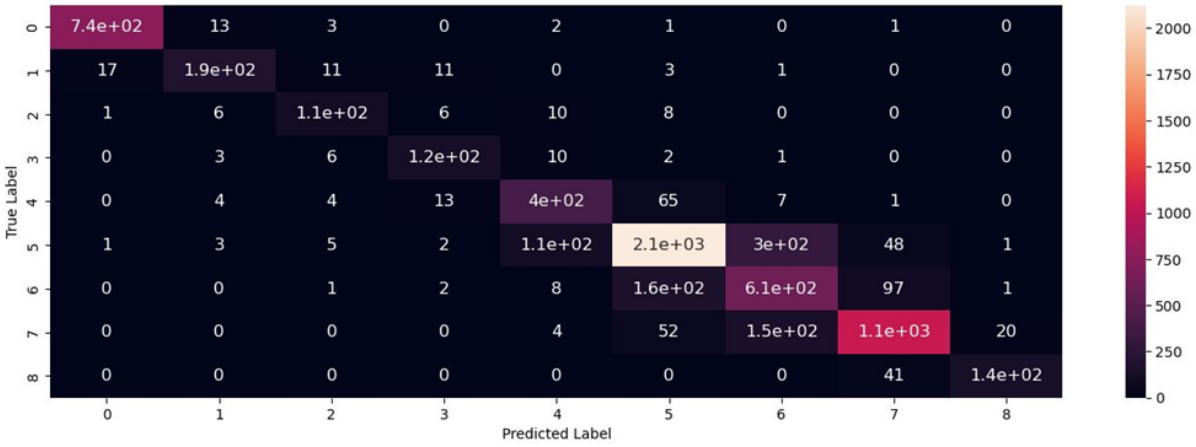

Figure 6: The confusion matrix of the transfer learning model

After completing the training, the transfer learning model was tested on the test set which produced the confusion matrix given in Figure 6. The accuracy of the model was found to be 81.9%. In addition, the macro precision, recall and F1 scores were computed as 0.814, 0.812, and 0.811 respectively.

The results of test phases for CNN trained from scratch and the CNN utilizing ResNet-18 weights were taken into comparison. Accuracy for the first CNN was 79.3%, meanwhile the accuracy for the second CNN was 81.9 %. Initially, the first impression regarding the results was that there was not a

considerable augmentation in model performance. The same trend was also existent for precision, recall and F1 score metrics: Greatest increase in these metrics was at most 4 %. The raise of performance for these metrics was modest.

Nevertheless, it was insufficient to be solely limited to the perspective obtained from the general performance metrics. Different deep learning tasks involve disparate constraints and it would be meager to confine the effort to comprehend the performance of the model to a handful of performance metrics. Therefore the competence of the model at the task of assigning individuals to age intervals can be better understood via inspecting the number of false predictions whose actual and predicted value distance is larger than two classes. The motivation behind this metric is that these types of false predictions constitute a more severe error than false predictions which place the given image in an interval close to the actual age interval. For convenience, this metric will be referred to as the "label distance metric" in the rest of the report.

In the results of the first CNN model, the label distance metric is 69. For the CNN model with transfer learning, the label distance metric is 39.

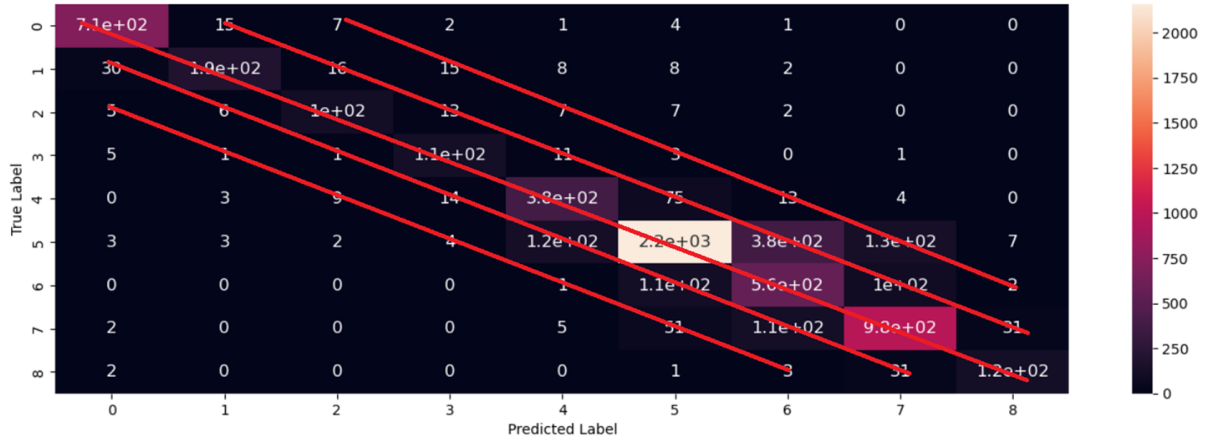

Figure 7: The confusion matrix of the CNN model with diagonals indicating the "perfect" diagonal and the "acceptable" diagonals.

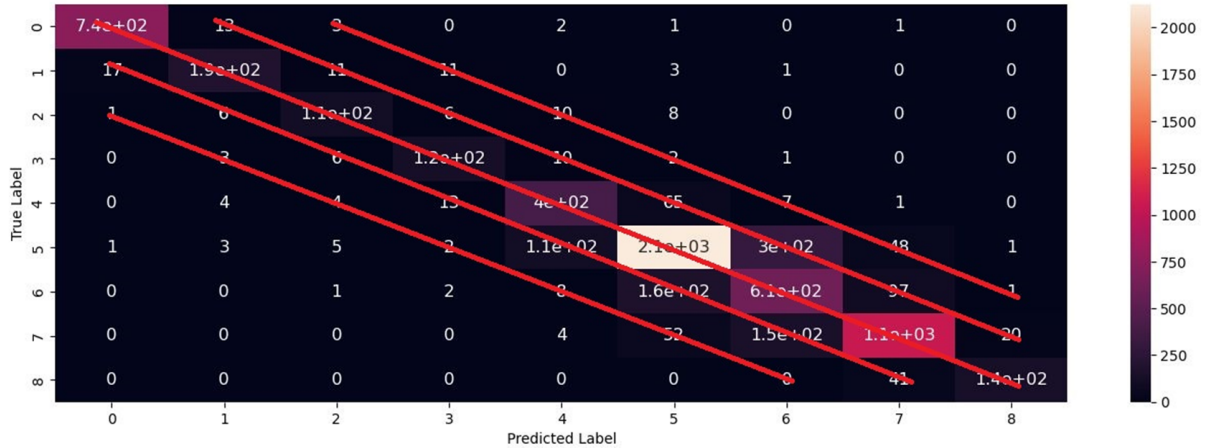

Figure 8: The confusion matrix of the transfer learning model with diagonals indicating the "perfect" diagonal and the "acceptable" diagonals.

The inner red diagonals correspond to the predictions which deviate from the actual label by 1 class, whereas the outer red diagonals correspond to the predictions which deviate from the actual label by 2 class. The remaining values in the upper triangular and lower triangular are summed to detect the predictions that are actually problematic and cannot be accepted.

Our dataset comprised a total of 33,405 images. We have divided the dataset into 70 % training, 10 % validation and 20 % test parts. As a result of this split, the test dataset consisted of 6681 samples.

Following the determination of the label distance metric for the both models and the length of the test dataset, we can calculate the ratio of error in terms of this specialized metric.

The ratio for CNN trained from scratch was 69/6681 = 0.01032779524, whereas the ratio for the CNN utilizing transfer learning weights was 39/6681 = 0.00583744948. The number of unacceptable false predictions decreased by half. This was not imminent in the initial investigation done via general performance metrics.

V. Conclusion

Final accuracy for the first CNN model was 79.3% the CNN with transfer learning 81.9%. Notwithstanding, we have demonstrated the improvement in model performance from CNN trained from scratch to CNN utilizing transfer learning parameters via the introduction of a task specific metric “label distance metric”. The ratio of unacceptable false predictions was cut by half in the second model, verifying our expectation that the transfer learning would yield significant increase in model performance.

We would like to note that we have observed a slight overfit to the dataset for our model, in both the CNN trained from scratch and the CNN utilizing transfer learning weights. We have monitored that in the epoch vs. accuracy and epoch vs. loss plots even if the training loss kept on decreasing, the validation loss stayed more or less the same. This behavior indicated a slight overfit. Nevertheless, success of our model was proven in the test phases both by general performance metrics accuracy, precision, recall, F1 score and the specifically introduced “label distance metric”.

VI. Appendix

Work Distribution

Kaan Hamurcu: Prepared the dataset for reading and made it available for operations, worked in the process of building the convolutional neural network layers and their effective operation.
Efser Efe Kantik: Conducted research on age classification, worked in the process of building the convolutional neural network layers and their effective operation.
Cemhan Kaan Özaltan: Provided hands-on experience regarding the multiple libraries of Python, worked in the process of building the convolutional neural network layers and their effective operation.
Ömer Özaltan: Worked in the process of building the convolutional neural network layers and their effective operation, prepared the dataset for reading and made it available for operations.
Ahmet Batuhan Sancak: Worked in the process of building the convolutional neural network layers and their effective operation, implemented transfer learning to the model, took part in hyperparameter tuning.

VII. References

[1] P. Mishra, “Why are convolutional neural networks good for image classification?,” *Medium*, 20-Jul-2019. [Online]. Available: https://medium.datadriveninvestor.com/why-are-convolutional-neural-networks-good-for-image-classification-146ec6e865e8. [Accessed: 01-Dec-2022].

[2] F. Rabbi, “Facial age,” Kaggle, 25-Jan-2019. [Online]. Available: https://www.kaggle.com/datasets/frabbisw/facial-age. [Accessed: 01-December-2022].

[3] “UTKFace,” Utkface. [Online]. Available: https://susanqq.github.io/UTKFace/. [Accessed: 01-December-2022].

[4] Mustapha, M. F., Mohamad, N. M., Osman, G., & Ab Hamid, S. H. (2021). Age group classification using Convolutional Neural Network (CNN). Journal of Physics: Conference Series, 2084(1), 012028. https://doi.org/10.1088/1742-6596/2084/1/012028

[5] “Models and pre-trained weights,” *Models and pre-trained weights - Torchvision main documentation*. [Online]. Available: https://pytorch.org/vision/stable/models.html. [Accessed: 17-Dec-2022].