2022-2023 FALL

CS-464

Age Detection Using Convolutional Neural Networks

Progress 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

While proposing this project, our aim was to generate a model which will classify presented test images to predetermined age intervals and therefore produce a tool for detecting the age of a given individual. In order to successfully realize this framework, we have to utilize a large dataset. Reason for such a necessity manifesting itself is that age classification is a challenging task and feeding the model with as much training data as possible will be to our benefit. Therefore we have created our dataset from the "facial age" dataset, containing 9778 colored images [1] and the "UTK Face" dataset which includes 23,708 colored images [2]. 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.

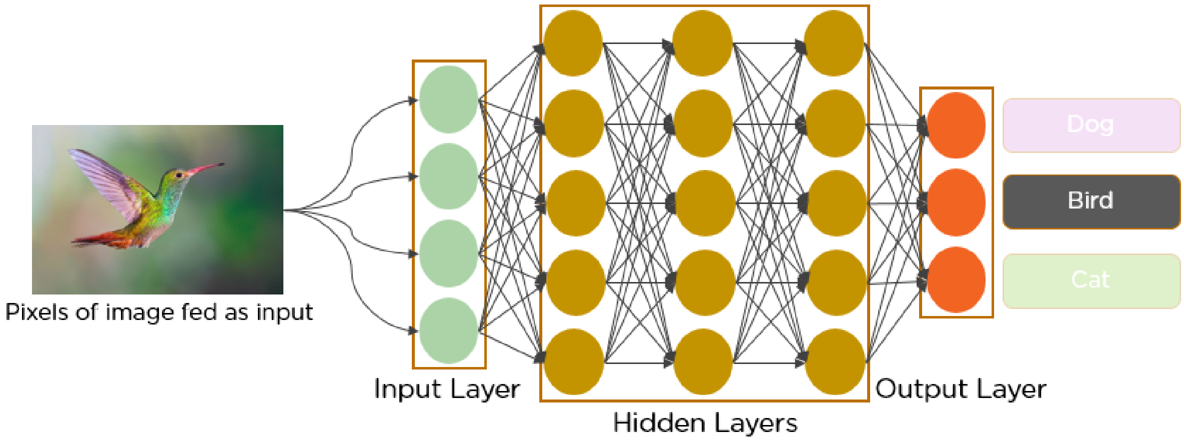

Figure 1: An example convolutional neural network used in a classification task [3].

The outlined problem was identified as a classification task rather than regression to achieve a better accuracy for two reasons. The first reason is that the dataset has some imbalances as mentioned. By combining ages into groups, a more balanced dataset can be obtained. The second reason is that facial appearance does not vary across ages close to each other. That is to say, for instance, the facial appearance of a person who is 27 would not differ much from their appearance when they were 25. Furthermore, people age at different rates. Therefore, facial patterns emerge in age groups more noticeable than at individual ages.

Consecutively, we ought to determine the intervals for our age classes. In order to set our class intervals appropriately, we have done research on the articles which are about age classification using convolutional neural networks. According to [4], 8 class intervals were defined for a dataset containing faces of individuals from age 2 to 80. Their intervals were 2-3, 4-6, 7-8, 9-11, 12-20, 21-35, 36-50 and 51-80, corresponding to infancy, early childhood, middle childhood, late childhood, adolescence, early adulthood, midlife and mature adulthood age groups respectively [4]. On the other hand, we possess datasets containing images of individuals in between age 1 to 116. Considering our resources, we have decided to add one more class interval as 80-116 and alter the lower limit of first class interval 2-3 to become 1-3.

the images in the AAF dataset are demonstrated in figure 4.

Table 2. Age group.

Age group	Years old	Number of data
Adolescence	12-20	1212
Early Adulthood	21-35	5198
Early Childhood	4-6	381
Infancy	2-3	277
Late Childhood	9-11	340
Mature Adulthood	51-80	2306
Middle Childhood	7-8	249
Midlife	36-50	3353

Figure 2: Age classification brackets used in a classification task with convolutional neural networks [4].

Overall, the given task can be described as an image classification task. Thus, a convolutional neural network (CNN) is the most suitable model to be used rather than a machine learning model.

II. Work Done

To start with, we have downloaded the "facial age" dataset. Following this step, we have to properly read this dataset which was in PNG format in order to be able to correctly proceed in the generation of the model. We first converted these images to arrays. After that, we splitted these images into training and test samples, where we enabled the shuffle functionality in the split function in order to ensure that the train and test set split will be always different in each and every training. After the train test split, we have converted the data from images to tensor format. The conversion to tensor format was not sufficient, we have to create dataset objects to pass into the DataLoader of our CNN model. The dataset objects contain both the “x” and “y” parts of the training data, where “x” part corresponds to the features of the subject sample and the y part corresponds to the relevant class label of the sample.

The step after the creation of the dataset objects and their preparation was the generation of the convolutional neural network. We have employed the functions of the PyTorch library in order to implement the structure of the neural network. 2D convolution layers form the backbone of our model, considering the fact that our subjects are 2D images.

Pooling is crucial for a convolutional neural network to summarize the output from feature maps. They can be applied in between convolution layers. We have proceeded similarly, too, after the application of two 2D convolution layers. For that pooling layer, we have used max pooling. We have repeated the employment of two 2D convolution layers and one max-pooling layer.

“Linear” function helped us to apply required linear transformations to eventually obtain the 7 defined class-intervals as our output. As our activation function, we have resorted to Rectified Linear Unit (ReLU).

Consecutively with the definition of a convolutional neural network, we have proceeded with the setting of loss function. We have decided to use cross-entropy as our loss function for our task. When it comes to the optimizer method, our choice was going with stochastic gradient descent.

We should also state that we have run our code on GPU, where going along with CPU was very inefficient considering our task. Possible alternatives such as utilizing GPUs or TPUs available from Google Colaboratory were also considered, nevertheless they do not appear to be long-term alternatives as subscription services of Google Colaboratory are not feasible in monetary terms.

We have selected our batch size to be 64 initially. The reason was to optimize our available work power as much as possible and to not assign intolerable computing load on the system.

For the beginning, we have taken the learning rate as 0.001. Notwithstanding, a learning rate on the order of one out of a thousand can accrue to be very large to properly enable our neural network to learn weights. Therefore also with the feedback of our teaching assistant, we may alter the learning rate to a smaller level. The number of epochs was set as 20 during training.

We have trained our convolutional neural network, nevertheless at the time of our project progress meeting we have detected an error in the compatibility of our loss function type and the elements of our output.

III. What Remains to be Done?

First of all, the problem in compatibility of loss function type and the elements of our output will be solved. In addition, image transforms will be applied to our images before feeding them into the neural network for training. Proceeding from that point, we will test our model. The results from the test phase will be utilized to obtain a confusion matrix, where we will inspect our performance metrics, which are accuracy, F1-score, recall and precision, accordingly. Moreover, we will expand our dataset to the “UTKFace” dataset alongside the “facial age” dataset. In succession, the convolutional neural network model will be optimized to predict age classes with higher accuracies. The hyperparameters will be tuned for this purpose. The number of layers in the CNN architecture might be modified as well.

We can also capitalize on the weights of previously trained neural networks such as ImageNet and ResNet. The results obtained through transfer learning will be compared to that of our model.

Eventually, we will complete our task of devising a model for age classification based on faces trained and tested using both the “facial age” and “UTKFace” datasets.

IV. Work Division

Kaan Hamurcu: 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.

Efser Efe Kantik: Took place in the writing phase of the report, conducted research on age classification.

Cemhan Kaan Özaltan: Took place in the writing phase of the report, provided hands-on experience regarding the multiple libraries of Python.

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, contributed to the writing of the report.

References

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

[2] "UTKFace," Utkface. [Online]. Available: https://susanqq.github.io/UTKFace/. [Accessed: 01-December-2022].

[3] Mandal, M. (2022, December 1). *CNN for deep learning: Convolutional Neural Networks*. Analytics Vidhya. Retrieved December 3, 2022, from https://www.analyticsvidhya.com/blog/2021/05/convolutional-neural-networks-cnn/

[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