A FACIAL AGE GROUP CLASSIFIER BASED ON DEEP LEARNING TECHNIQUES

Ahmad ALSALEH

MASTER OF SCIENCE THESIS

Department of Computer Engineering

Supervisor: Assist. Prof. Dr. Cahit PERKGÖZ

Eskişehir

Anadolu University

Institute of Graduate Programs

September 2022

FINAL APPROVAL FOR THESIS

This thesis titled A FACIAL AGE GROUP CLASSIFIER BASED ON DEEP LEARNING TECHNIQUES has been prepared and submitted by Ahmad ALSALEH in partial fulfillment of the requirements in "Anadolu University Directive on Graduate Education and Examination" for the Degree of Master of Science in Computer Engineering Department has been examined and approved on 08/09/2022.

Committee Members		**Title, Name and Surname**	**Signature**
Member	:	Assist. Prof. Dr. Cahit PERKGÖZ	
Member	:	Assoc. Prof. Dr. Kemal ÖZKAN	
Member	:	Assist. Prof. Dr. Burcu YILMAZEL	

Prof. Dr. Murat TANIŞLI
Director of the Institute of Graduate Programs

08/09/2022

SUPERVISOR APPROVAL

Master of Science student Ahmad ALSALEH, whom I supervise, has completed this thesis titled A FACIAL AGE GROUP CLASSIFIER BASED ON DEEP LEARNING TECHNIQUES. According to my inspections, the work is scientifically and ethically appropriate for the student to the thesis defense exam.

Supervisor
Assist. Prof. Dr. Cahit PERKGÖZ

ABSTRACT

A FACIAL AGE GROUP CLASSIFIER BASED ON DEEP LEARNING TECHNIQUES

Ahmad ALSALEH
Department of Computer Engineering
Anadolu University, Institute of Graduate Programs, September 2022
Supervisor: Assist. Prof. Dr. Cahit PERKGÖZ

Recently, researchers in the computer vision field have become increasingly interested in extracting important information from the human face, including estimating demographic characteristics such as age, gender, and race, due to their increasing applications in the real world. Models based on convolutional neural network (CNN) have proven that they are one of the most important techniques capable of extracting facial features automatically from images of human faces.

In this thesis, a novel CNN-based model to extract facial discriminative features from face images and classify those images into the corresponding age group is proposed. The overall performance of the proposed model is evaluated on both UTKFace and Facial-age datasets. The results showed that the model has a better performance compared with recent studies in terms of classification accuracy. The model achieved age group classification accuracy of 87.82% on the mentioned facial aging datasets.

Keywords: Facial age estimation, Age group classification, Deep learning, Convolutional neural network (CNN).

ÖZET

DERİN ÖĞRENME TEKNİKLERİNE DAYALI YÜZ YAŞ GRUBU SINIFLANDIRICISI

Ahmad ALSALEH
Bilgisayar Mühendisliği Anabilim Dalı
Bilgisayar Bilim Dalı
Anadolu Üniversitesi, Lisansüstü Eğitim Enstitüsü, Eylül 2022
Danışman: Dr. Öğretim Üyesi. Cahit PERKGÖZ

Son zamanlarda, bilgisayarla görü alanındaki araştırmacılar, gerçek dünyadaki artan uygulamaları nedeniyle, yaş, cinsiyet ve ırk gibi demografik özelliklerin tahmin edilmesi de dahil olmak üzere, insan yüzünden önemli bilgilerin çıkarılmasıyla giderek daha fazla ilgilenmeye başladılar. Bu alanda Evrişimsel Sinir Ağlarına (ESA) dayalı modeller, insan yüzlerinin görüntülerinden yüz özelliklerini otomatik olarak çıkarabilen en önemli tekniklerden biri olduklarını kanıtlamışlardır.

Bu tezde, yüz görüntülerinden yüz ayırt edici özellikleri çıkarmak ve bu görüntüleri ilgili yaş grubuna sınıflandırmak için CNN tabanlı yeni bir model önerilmiştir. Önerilen modelin genel performansı hem UTKFace hem de Facial-age veri kümelerinde değerlendirilmiştir. Sonuçlar, modelin sınıflandırma doğruluğu açısından son çalışmalara kıyasla daha iyi bir performansa sahip olduğunu göstermiştir. Önerilen model, bahsedilen veri kümelerinde %87,82'lik yaş grubu sınıflandırma doğruluğu elde etmiştir.

Anahtar Sözcükler: Yüz yaşı tahmini, Yaş grubu sınıflandırması, Derin öğrenme, Evrişimsel Sinir Ağlarına (ESA).

ACKNOWLEDGEMENTS

First and foremost, I would like to express my sincere gratitude to my research supervisor, Assist. Prof. Dr. Cahit PERKGÖZ for all the support, recommendation and involvement in every step and all stages of research project. By his contributions and advises this work has been completed. I would like to thank him very much for his understanding and I appreciate all the efforts made throughout my study duration.

I would like also to take this opportunity to express my profound gratitude to my to my family especially my wonderful parents for the continuous support and encouragement throughout my years of study. They stood by me in my worst and hardest moments, in the moments when I thought I would never continue this work. This research would not have been possible without their generous support and encouragement.

Also, I would like to offer my special thanks to my friends for their support and always being by my side to achieve my goals.

Huge gratitude to the whole personnel at Anadolu University, whose their support made my studies a pleasant and fascinating experience.

And most importantly, I owe it all to the Almighty God for blessing me with the wisdom and health to carry out this work and for allowing me to see it through to its completion.

Ahmad ALSALEH

08/09/2022

STATEMENT OF COMPLIANCE WITH ETHICAL PRINCIPLES AND RULES

I hereby truthfully declare that this thesis is an original work prepared by me; that I have behaved in accordance with the scientific ethical principles and rules throughout the stages of preparation, data collection, analysis and presentation of my work; that I have cited the sources of all the data and information that could be obtained within the scope of this study, and included these sources in the references section; and that this study has been scanned for plagiarism with "scientific plagiarism detection program" used by Eskişehir Technical University, and that "it does not have any plagiarism" whatsoever. I also declare that, if a case contrary to my declaration is detected in my work at any time, I hereby express my consent to all the ethical and legal consequences that are involved.

Ahmad ALSALEH

CONTENTS

Page

LIST OF TABLES

Page

LIST OF FIGURES

Page

GLOSSARY OF SYMBOLS AND ABBREVIATIONS

AE	:	Age Estimation
AFAD	:	Asian Face Age Dataset
AI	:	Artificial Intelligence
ASHCI	:	Age Specific Human Computer Interaction
BERC	:	Biometric Engineering Research Center
CACD	:	Cross-Age Celebrity Dataset
CNN	:	Convolutional Neural Network
DCNN	:	Deep Convolutional Neural Network
DEX	:	Deep EXpectation
DL	:	Deep Learning
ECRM	:	Electronic Customer Relationship Management
FC	:	Fully Connected
HOIP	:	Human Object Interaction Processing
IFDB	:	Iranian Face Database
ML	:	Machine Learning
VGGNET	:	Visual Geometry Group Network
Xception	:	Extreme Inception

1. INTRODUCTION

Social life and our presence in society are greatly affected by the ability of individuals to look at our faces and estimate our age, which is called facial age estimation, since the face carries essential information regarding individual traits and this information play a crucial role in face-to-face contact between individuals. Many of these social interactions are influenced by a variety of variables, including age, gender, expression, social status, and ethnicity, which in turn are considered manifestations of facial features. When it comes to applications in the business world, these facial features are very important; nevertheless, it is not yet feasible to accurately and consistently predict them based on a face image alone (Agbo-Ajala & Viriri, 2021).

Age estimation can be very useful in many real-time applications such as age simulation, access control and surveillance, electronic customer relationship management (ECRM), age specific human computer interaction (ASHCI), soft biometrics, finding missing persons and so on (Al-Shannaq & Elrefaei, 2019).

Although there is much research and continuing interest in the field of age estimation and both academia and industry dedicate their efforts from algorithm design, modeling, data collection, and system performance testing, this task is still very challenging due to its association with many factors. These factors, which relate to the aging process of the face, are divided into two types: internal factors such as the change in the size of the face, the shape of facial features, wrinkles, facial contour and the distribution of facial features on the face, and external factors as well, such as lifestyle and health, eating habits, social, environment and weather conditions (Agbo-Ajala & Viriri, 2021).

Much research has been done in the field of automatic age estimation to obtain the facial features needed to represent the face correctly and as a result obtaining these features was very difficult and complicated because most of these researchs were based on hand-crafted methods (Agbo-Ajala & Viriri, 2021). Hand-engineered foreknowledge was needed to correctly predict human age with this type of methods. In the most recent studies of the age estimation, the use of Convolutional Neural Network (CNN) approaches has been utilized, which has led to improved accuracy in age estimation performance. Discriminant features can be learned directly from raw pixels and obtained filters used to display face photos in another feature space using deep learning and Convolutional Neural Network CNN approaches (Liu et al., 2015). Because CNN-based

methods are designed to deal with images and video, they are capable of obtaining distinctive and powerful facial features from a facial image (Antipov et al., 2016).

Despite the age estimation nature which is very challenging task, there are still numerous continuing efforts to increase the accuracy of age estimate, and researchers continue to examine many relevant CNN methodologies in order to further improve the findings. Therefore, in this work we present a deep CNN-based model to improve the results found in the literature in this field. The following are the thesis main contributions and findings:

- A novel CNN-based classification model in order to address facial age estimation task from human face images where we consider this task as a classification task that encodes each group of ages as a class label is proposed.
- The model is compared with the most recent state-of-art related works that uses age group classification as a definition for age estimation task.
- Finally, the UTKFace and Facial-age datasets are used to evaluate the proposed model and as a result the model achieved a better performance comparing to state-of-art works in term of accuracy.

The remainder of the paper is as follows: Chapter 2 is background that introduces and explains the basic information of the techniques used in this work. Chapter 3 explains age estimation systems in detail and summarizes other work relevant to the study. Chapter 4 explains the proposed age group classification model in detail. Chapter 5 contains the experimental results as well as a comparative analysis. Chapter 6 presents our findings and discusses potential future work.

2. BACKGROUND

In this chapter the basics of the technology used in this study are presented starting from the artificial intelligence (AI), Machine learning (ML) and deep learning (DL) definitions. Then the types of these technologies are discussed. After that, we have explained the Convolutional Neural Network (CNN) fundamentals building blocks in detail.

2.1. Artificial Intelligence (AI): What is it?

Programming computers and machines to mimic human thought and behavior is the goal of the science known as artificial intelligence (AI).

Although it might seem straightforward, there is no existing computer on the market can equal the complexity of human intelligence. Although computers are excellent at applying rules and carrying out tasks, there are occasions when a very simple activity for a person may be quite complicated for a computer. For instance, carrying a tray of drinks through a packed bar and serve the right client every day requires complicated decision-making and relies on a large amount of data that is passed between neurons in the human brain. Machine learning (ML) and deep learning (DL) are steps towards a crucial component of this objective, which is to analyze vast amounts of data and make decisions/predictions based on it with the least amount of human interaction. Figure 2.1 shows the main differences between AI, ML, and DL.

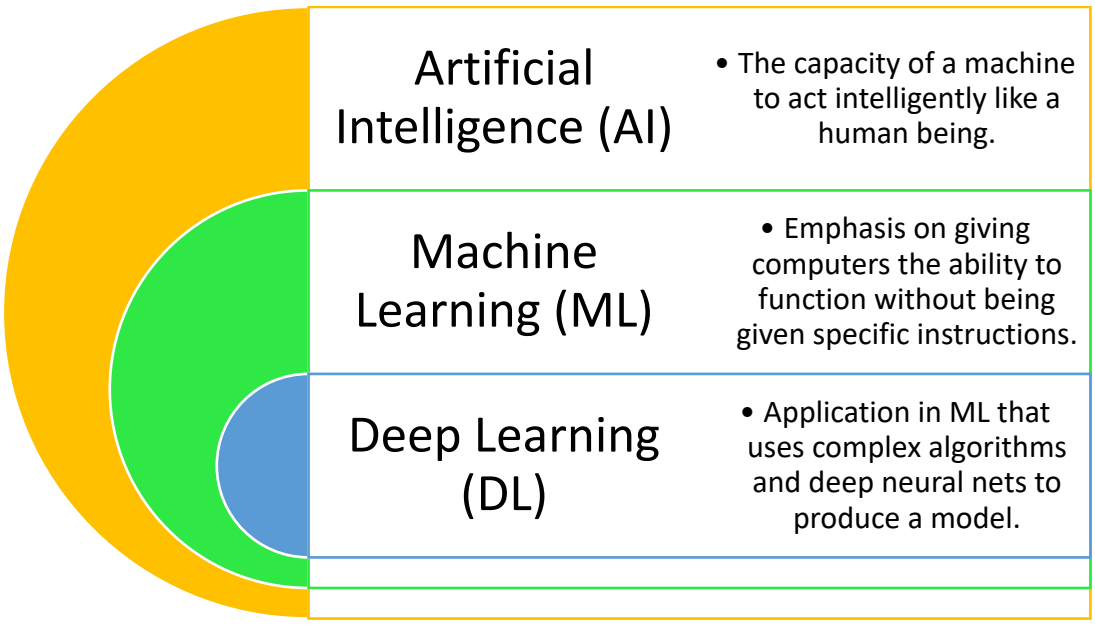


Figure 2.1. *Main differences between AI, ML, and DL*

2.2. What Is Machine Learning?

The field of study known as "machine learning" is a subfield of artificial intelligence that employs a wide variety of methods in order to train computers to do tasks without being specifically programmed to do so. Machine learning is capable of functioning in dynamic networks independently of the intervention of humans or the application of complex mathematical procedures (Jordan & Mitchell, 2015).

The ability of machine learning algorithms to solve many problems results in their implementation across a wide range of different real-world applications. These algorithms are used in the process of designing machines that are able to learn continually via experience. In recent years, learning approaches have seen widespread use in a variety of contexts. Recent developments in learning algorithms have been driven by the development of new algorithms as well as accessing to enormous datasets. These developments have also led to the creation of low-cost algorithms (Jordan & Mitchell, 2015).

Generally speaking, the goal of learning algorithms is to enhance the performance of a task by repeated practice, training, and learning from previously gained knowledge and experience. For instance, the objective of age estimation systems is to categorize a picture of an individual's face into one of the age group classes. An increase in performance may be achieved by improving the accuracy of the classification, as well as the experiences through which the algorithms train and learn from a collection of human face photos labeled with their ages.

There are mainly four categories into which machine learning algorithms may be placed: supervised, unsupervised, semi-supervised, and reinforcement learning algorithms.

Supervised learning A group of machine learning known as "supervised learning" needs the most constant human involvement, thus the name. In this type of learning, the computer is provided with a lot of examples of training data and we provide or 'teach' it the output of each example so that for the future examples it will know how to react. This output that is provided to the computer is called target data or labels.

The programmer or data scientist is fed more data into the computer after it has been trained on the training set of data to see how the computer will react to the new data and validate the accurate predictions or can make changes for any wrong ones. As an example, when a programmer attempts to teach a computer how to categorize photos

the example photos will be provided to the computer and instruct it to do classification for each image, then either validate or correct each computer prediction.

The model becomes increasingly accurate and able to handle new datasets that follow "learned" patterns as this process of supervision is continued. Nevertheless, it is not practical to keep checking computer performance and making modifications over time.

Semi-supervised learning Sometimes the existing data may not be all labeled, but a combination of this unlabeled data with properly labeled data is provided, and have the computer look for patterns within this data. ' Semi-supervised learning' is the name given to this type of learning.

Because it requires the knowledge of trained human experts, the cost of labeling the data is rather high in many practical circumstances. This is because the task requires human specialists to be taught. Therefore, in situations where the majority of observations do not have labels but some do, the most effective methods for developing models are those that are semi-supervised.

Unsupervised learning In this type of learning, as it is called "unsupervised learning", the computer works with data that has no labels and is allowed to identify any pattern or combination of patterns that it deems appropriate to represent the data. This type of learning sometimes yields findings that a human data analyst would not be able to see.

One of the most uses of this type of learning is called "clustering", in which the computer rearranges the data according to a pattern into groups. This technique is often used by shopping/e-commerce websites to determine what suggestions to provide to individual customers based on their prior purchases.

Reinforcement learning So far, we have mentioned three types of learning in which if the machine misunderstands or misclassifies the data there are no "consequences or penalties". In reinforcement learning, again, as it is called, there is reinforcement of the machine's behavior and it receives positive feedback when it does the right thing, and negative feedback when it does the wrong thing. Thus, the computer or machine begins to learn how to accomplish specific tasks through trial and error, knowing that a method is set up by the programmer or data scientist to award the machine a reward (for instance, a score) to reinforce its "good behavior".

It is essential for computers to be able to learn from vast, highly flexible, and unpredictable datasets via reinforcement learning. In future automated systems might be used in a wide variety of ways, from driving cars to scanning bags for harmful materials.

As a comparison between all these types, it is advised to employ reinforcement learning to solve comparison and decision-making difficulties rather than supervised or unsupervised learning techniques, which are often used for the majority of solving data analysis problems.

The type of data that is accessible serves as a determinant for both this classification and the associated selection of the machine learning approach. When both the nature of the data that is being input and the outputs (labels) that are anticipated are already known, supervised learning may be carried out. In this situation, the system has to be taught to simply map the inputs to the outputs that are given. Both regression and classification are examples of supervised learning approaches; however, the main distinction between the two is that regression may deal with continuous data, whilst classification works with discrete data. The Support Vector Regression (SVR) technique, the Polynomial Regression Method, and the Linear Regression Method are three of the most used types of regression methods. By contrast, classification employs discrete data values (class-labeled data). A few of the classification methods that are used most often include the Support Vector Machine, Logistic Regression, and K-Nearest Neighbor. Some algorithms, such as neural networks, are capable of doing both classification and regression when given the appropriate inputs. Unsupervised learning approaches are used in the training of systems for which the outputs are not clearly specified and the system is tasked with locating the pattern contained within the raw data. Clustering is a kind of unsupervised learning in which objects are grouped in accordance with given similarity criteria. K-means clustering is an example of this type. Clustering may be used to learn about both structured and unstructured data.

The success of predictive analytics is dependent on the efficiency with which machine learning technology utilizes historical data to build models and predicts future values. As an example of predictive modeling algorithms, we have Naive Bayes, neural networks and SVR.

2.3. What Is Deep Learning?

The ultimate objective of machine learning is to enable computers to execute some of the same tasks that humans do automatically without explicitly telling them how to do it. But the ability of computers and the way they interact with data such as understanding and collecting information from an image or video is still far less than what humans can do because it still behaves and think like machine.

As a step forward and to meet these challenges, deep learning models are designed with a very complex approach to machine learning as they are specifically modeled on the human brain. Deep Learning (DL) is a machine learning approach derived from artificial neural networks (ANN). The neural network is made up of neurons (which can be seen as variables) that are linked using weighted connections (can be considered as parameters). Therefore, complex and multi-layered deep neural networks (DNN) are built which in turn allow data to be passed between nodes (such as neurons) in highly interconnected ways. As a result, we get a non-linear transformation of the data which is getting more abstract.

In order to get the required set of outputs from the network, a learning algorithm, either unsupervised or supervised, is used in combination with the network. Unlabeled and labeled data, collected by unsupervised or supervised learning algorithms, respectively, are used to start off the learning process, which is then followed by iterative updates of the weights between all pairs of neurons. Because of this, when we discuss deep learning, we make reference to deep massive neural networks that have a high number of layers, where the word "deep" refers to the number of layers that are included inside a network. Although building, feeding, and learning such systems of this type require massive amounts of data, but once the system is in place, it may start producing results immediately, and the need for human involvement is relatively small.

Voice recognition, computer vision, and natural language processing are just a few examples of the many machine learning applications that benefit from deep learning models. Due to their efficiency in extracting features and patterns from massive amounts of data, these models also improve data compression and restoration in the spatial and temporal domains.

There are various deep learning architectures available, including Convolutional Neural Networks (CNN), Generative Adversarial Networks (GAN), Recurrent Neural Networks (RNN), Long-Short Term Memory (LSTM), Networks Boltzmann Machines

(BM), Feed forward Deep Networks (FDN), and Deep Belief Networks (DBN). The most commonly used DL architectures are CNN (used in spatially-distributed data) and RNN (for time series data). One of the most important types of deep learning that will be used in our proposed system is Convolutional Neural Networks (CNN) which will be explained in detail in the next section.

2.4. Convolutional Neural Network (CNN)

2.4.1. Basics of CNN

A feed-forward artificial neural network where the individual neurons are structured in such a manner that they react to overlapping areas in the visual field is known as a Convolutional Neural Network (CNN). CNN is a type of neural network inspired from biological principles. It was discovered in 1968 by Hubel and Wiesel (Hubel & Wiesel, 1968) working on the visual cortex of cats that the visual cortex comprises a complicated structure of neurons. This information was obtained through their early research. These neurons have a sensitivity to certain receptive fields, which are smaller sub-regions of the overall visual field. To cover the whole of the visual field, the subregions are tiled together. These neurons function as local filters across the input space, and they are well adapted to take advantage of the spatially local correlation that is present in real pictures.

The research carried out by Fukushima (Fukushima, 1980) begins by computing models predicated on the local connectivities that exist between neurons and on the transformations of the image that are structured hierarchically. In this body of work, he discovered that a type of translational invariance may be produced when neurons with the same parameters are put to patches of the preceding layer at various positions. According to this theory, LeCun constructed and trained CNNs utilizing the error gradient with back-propagation algorithm, achieving state-of-the-art performance on a wide variety of pattern recognition tasks (LeCun et al., 1989; Lecun et al., 1998). The implementation of weights sharing, which involves applying the same weights on the same feature map, is one of the most significant contributions that CNN has made to the field of neural networks. This improves the learning efficiency of the network by significantly reducing the number of trainable parameters.

Recently, CNN has included different active functions and regularization techniques, which boosts the non-linearity and independence of feature maps, resulting

in better comprehension and more stable learned features. In addition to functioning as an end-to-end structure, the deep-learning features contained within the structure have the potential to also be used as features to feed into classifiers in order to carry out computer vision tasks. This is similar to the way that traditional hand-craft features function. Convolutional neural networks need much less preparation compared to more conventional pattern recognition and computer vision techniques. To put it another way, the network is in charge of the learning filters that are traditionally hand-engineered in conventional algorithms. One of the most significant benefits of CNN is its independence on prior knowledge, which is a significant advantage when compared to the complexity that currently exists in designing hand-engineered features.

2.4.2. Structure of CNN

A Convolutional Neural Network (CNN) is comprised of two primary kinds of layers: convolutional layers and pooling layers. And as CNN continues to evolve, new features like new active functions and dropout are being introduced in order to improve its overall performance. Also, at the very last layer of the CNN, the loss layer that is used is determined by the kind of task being performed. A representation of a CNN's overall structure may be seen in Figure 2.2. Each and every component of CNN will be broken down in good depth.

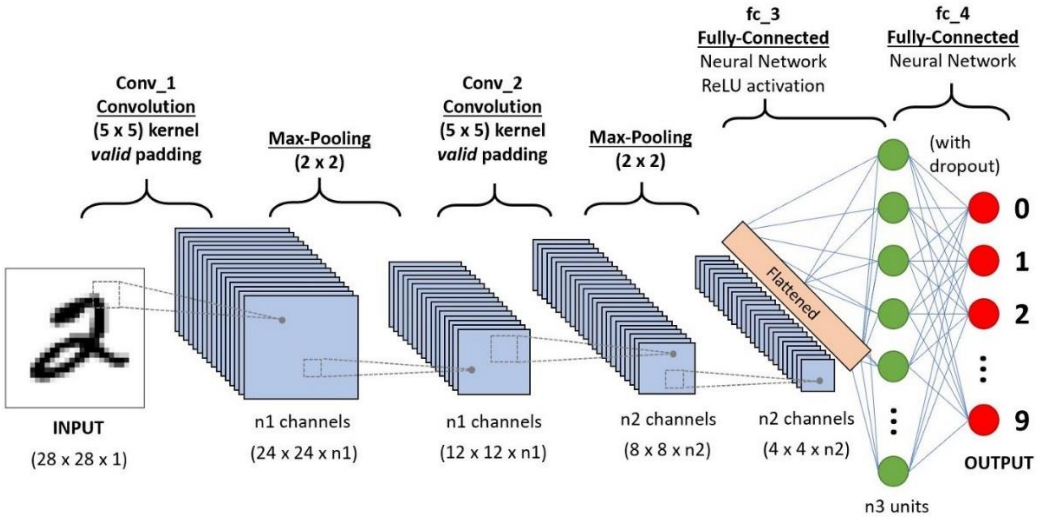


Figure 2.2. *Overall structure of CNN (Saha Sumit, 2018)*

2.4.2.1. Convolutional Layer

The convolutional layer is a fundamental component of a convolutional neural network (CNN), which distinguishes CNN from more conventional artificial neural networks. Convolutional operations on tiny areas have been proposed as a solution to the problem of having to learn billions of parameters (assuming all layers are completely connected), as a result of which this solution has been developed.

One of the most significant benefits of convolutional networks is that the weights in the convolutional layers are shared. This allows the same filters to be applied to the same feature map, which improves efficiency. By sharing weights, CNNs are able to increase their performance on computer vision tasks while also reducing the amount of necessary processing RAM. The impact of the weights sharing on the parameter reduction is seen in Figure 2.3. A comparison is made between convolutional layers that share their weights and those that are completely linked. The number of trainable parameters with the application of weights sharing is clearly less than the total number of trainable parameters without weights sharing, despite the fact that the same neurons are used in the hidden layer, which is the next adjacent layer. Therefore, the over-fitting issue of conventional neural networks was subsequently reduced by decreasing the number of trainable parameters in the network (Qiu Jiayan, 2016).

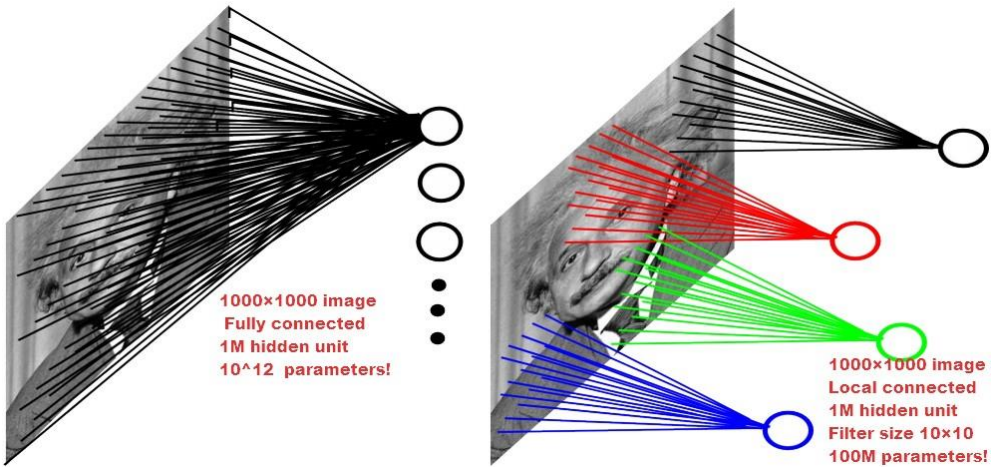


Figure 2.3. *The weights sharing's effect on parameter reduction (Qiu Jiayan, 2016)*

The convolutional layer's parameters are comprised of a series of learnable filters that are restricted in terms of their spatial size. The forward pass involves convolving each filter over the two dimensions of the input volume to generate a two-dimensional

activation map of that filter. The network adjusts the filters that will be activated by particular kinds of features taken from the input at certain positions. This is the same as the convolutional operation that is used in the traditional feature designed algorithms for the purpose of extracting basic features from input images. After that, the total output volume is created by stacking the activation maps for all of the filters along the depth dimension. The number of learned filters in the convolutional neural network has risen thanks to the aid of weights sharing, which allows the extraction of additional information from the input data.

When using a convolutional layer, a feature map is generated by repeatedly applying a function to different sub-regions of the whole picture which is in another way, a feature map can be generated by performing a convolution on the input image using a filter with adding a bias term. The Equation 2.1. shows this operation.

$$h^{k+1} = W^k \otimes h^k + b^k \tag{2.1}$$

Where $\otimes$ the convolution operation, *k-th* input feature map of given convolution layer which is referred as h^k, W^k is the weights of the filters, b^k the bias, and h^{k+1} is the output feature map.

A fully connected (FC) layer can be considered as special case of convolutional layer which takes all neurons in the previous layer and connects them upon itself. This means a FC layer is convolutional layer without the weights sharing feature and each filter of this layer has a size of 1 × 1. Reducing the spatially located of neurons and shaping more abstract and meaningful features are two of the primary roles of a fully connected layer.

The goal of the convolution operation is to extract high-level features, such as edges, from the input image. Since a Convolutional Neural Network (CNN) consists of multiple convolutional layers, Low-Level features like edges, colors, directions, etc. are captured by the first convolutional layer, while the more depth layers adapted to the High-Level features, giving us a network with a comprehensive understanding of the images in the dataset.

There are two types of results of the convolution operation: the result with the **valid padding** and the result with the **same padding**. In case of valid padding, the convolved feature dimensionality is reduced compared to the input. While in the same padding the dimensionality remains the same.

2.4.2.2. Pooling Layer

In a similar manner to that of the Convolutional Layer, the main goal of the pooling layer is to reduce the spatial size of the Convolved Feature which, in turn, helps bring a decrease in the total of computational power that is required to process the data through dimensionality reduction. In addition to this, it is helpful for extracting dominating characteristics that are rotationally and positionally invariant, which in turn helps to sustain the process of efficiently training the model (Saha Sumit, 2018).

Pooling may be broken down into two main types: **Max pooling** and **Average pooling**. The Max Pooling provides the greatest value that may be derived from the portion of the image that is being processed by the filter. In contrast, the Average Pooling calculates the mean value of all the values in the portion of the image that are affected by the filter and delivers that result.

In addition to dimension reduction, the max pooling does denoising functions where it eliminates noisy activations entirely and preform as noise suppressant, while the average pooling mechanism does carry out dimensionality reduction as a technique for eliminating noise. As a result, max pooling achieves far better results than average pooling does.

The difference between the max pooling and the average pooling is seen in Figure 2.4 Both of these pooling methods use filters with a size of 2 × 2 and a stride of 2. Putting the patch, which is also referred to as the window, through the pooling layer allowed us to down-sample it from 4 by 4 to 2 by 2, as seen in the Figure 2.4. In the case of max pooling, the maximum value is taken across all four values of a 2×2 patch (given that the size of the filter is 2×2), and then the filter moves by 2 (the stride size) before computing the next value in the same manner as before. In this manner, the filter, which is also referred to as the kernel, travels down the path from the top left to the bottom right across the input pitch.

Combining a network's convolutional layer with its pooling layer results in the i-th layer of a CNN. Adding more of these layers to capture low-level details would need more computational resources. Depending on the sophistication of the photos, the number of layers may grow.

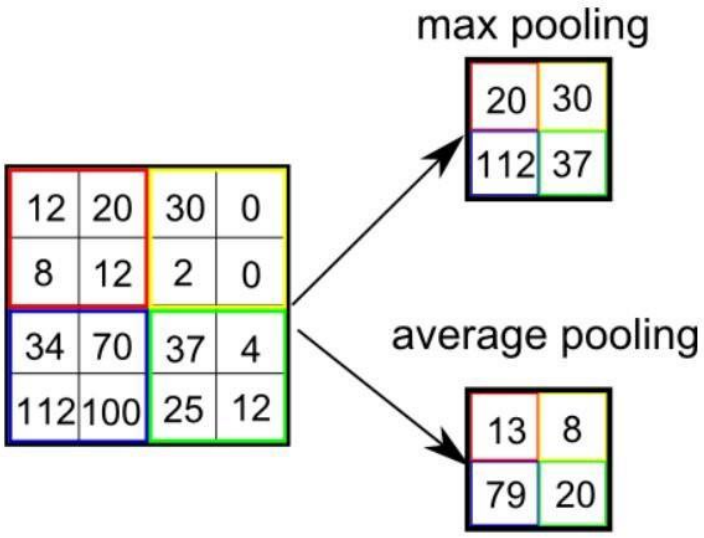


Figure 2.4. *The difference between Max pooling and Average pooling (Saha Sumit, 2018)*

2.4.2.3. Activation function

The incorporation of an activation function is one of the aspects of CNN that is considered to be among the most important. Activation functions raise the level of non-linearity in networks, which ultimately results in a comprehensive understanding of the input batch. The activation function, in addition to having a non-linear structure, produces a feature map that is free of data values that are very extreme which raises the level of neuronal independence in the subsequent layer, which, in turn, raises the level of network stability.

The three activation features that are used most often in CNN will be discussed in this section next. One crucial thing that everyone agrees on is that all activation functions should be differentiable which guarantees that the back-propagation process can be applied throughout the training phase.

Sigmoid Function: The Equation 2.2 is a definition of the sigmoid function.

$$sigm(x) = \frac{1}{1+e^{-x}} \tag{2.2}$$

Also, the Figure 2.5 shows the sigmoid function.

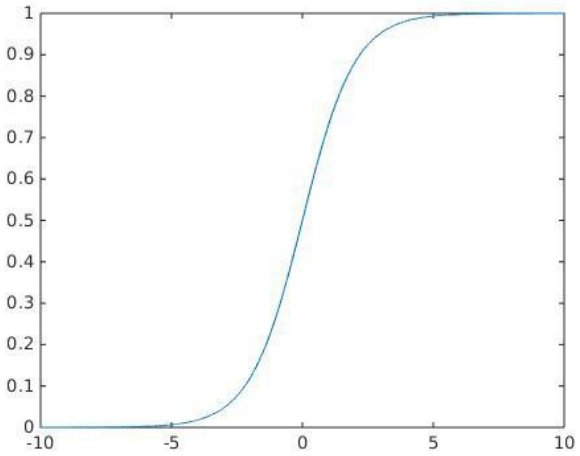


Figure 2.5. *Sigmoid function curve*

Figure 2.5 illustrates the sigmoid function's transformation of a real number into a value between zero and one. In instance, zero and one are the most common big values for negative and positive numbers, respectively. The sigmoid function has seen widespread use due to its intuitive representation of a neuron's firing rate, which ranges from 0 (no firing) to 1 (maximal firing).

Recent study demonstrates that the non-linearity of the sigmoid function performs poorly in specific actual situations due to the fact that it has two key drawbacks (Qiu Jiayan, 2016). Despite the fact that the sigmoid function has been utilized extensively, this finding is interesting.

The first disadvantage is that the Sigmoid function may quickly reach saturation, which eliminates gradients throughout the training phase. When the activation of a sigmoid neuron reaches its maximum at either the tail of 0 or 1, the gradients in these areas are virtually zero, which is a characteristic of the sigmoid neuron that is highly unpleasant. The gradient will be multiplied by the gradient of this gate's output for the whole objective while back-propagation is taking place therefore, if the local gradient is extremely modest, it will effectively kills the gradient and nearly has no signal flow from the neuron to its weights and then to its data, which is a recursive process. Additionally, in order to avoid saturation while initializing the weights of sigmoid neurons, one must pay special attention to the process. For instance, if the initial weights are set too high, the majority of the network's neurons would get saturated, and the network will almost completely lose its capacity for learning.

The other disadvantage is that the sigmoid function's outputs are not centered at zero. Since neurons in the subsequent layer will be exposed to off-center information, which is a not desired. If the input to a neuron is always positive, the resulting gradient during back-propagation will be either positive or negative, which affects the dynamics of gradient descent. This may cause the process of updating the weights to become unstable and noisy. When these gradients are aggregated over a data batch, the final update for the weights may have variable signs that reduce the caused error (Qiu Jiayan, 2016) which is an inconvenience, however it pales in contrast to the issue of saturated activation.

Tanh function: The Equation 2.3 is a definition of the hyperbolic tangent function.

$$\tanh(x) = \frac{1- e^{-2x}}{1+e^{-2x}} \tag{2.3}$$

Also, the Figure 2.6 shows the tanh function.

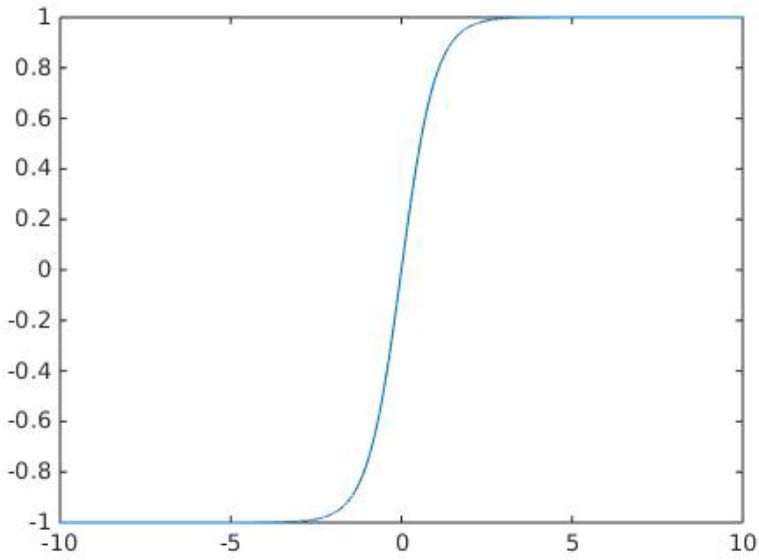


Figure 2.6. *Tanh function curve*

Non-linearly squishing a real-valued number to the to the range between -1 and 1 is what the tanh function does, as seen in Figure 2.6, making it similar to the sigmoid function. Despite the fact that it features zero-centered outputs, which makes it more stable in weight-updating dynamics, it nevertheless suffers from the saturated activation

issue that we saw in sigmoid function, so the tanh function has one advantage against the sigmoid function but it is still does not performing well in reality.

Rectified Linear Unit (ReLU): This function can be defined in the Equation 2.4.

$$ReLU(x) = max(0, x) \tag{2.4}$$

Also, the Figure 2.7 shows the ReLU function. Currently, in the field of CNN, ReLU is the most popular activation function. The ReLU function has two primary benefits. In the first place, implementing ReLU is as easy as setting a threshold to zero, in contrast to the more complex sigmoid function and tanh function, which require operations involving exponentials. Consequently, CNN models with ReLU can be trained much quicker than their tanh and sigmoid equivalents (Qiu Jiayan, 2016). The benefit of the CNN is that it does not need a lot of preprocessing, and ReLU's resilience to saturation can also be added to this advantage.

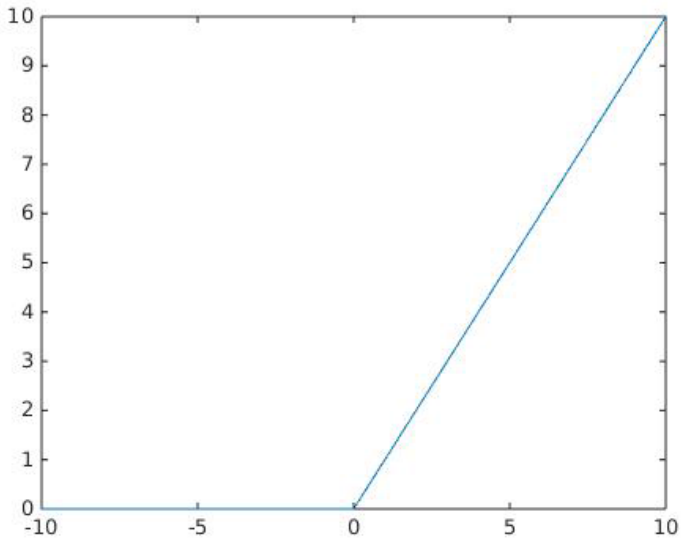


Figure 2.7. *ReLU function curve*

The ReLU function has certain benefits, but it also has some drawbacks, such as the fact that ReLU units might be die during training. If the learning rate is too high, for instance, ReLU will be knocked off the data manifold, which means it will permanently die and refuse to activate any data point during training. However, with the learning rate appropriately adjusted, this becomes less of a concern.

Rectified Linear Unit Family is a group of ReLU functions such as Leaky ReLU and Parametric ReLU that seeks to solve this issue (Qiu Jiayan, 2016). These functions will have a negligible negative slope, for example 0.01, when x less than zero, nd the definition might be modified in the Equation 2.5:

$$f(x) = \begin{cases} x & x \geq 0 \\ \alpha x & x < 0 \end{cases} \quad (2.5)$$

Where α is a small constant (of 0.01, or so). Here is the primary goal when using this leaky ReLU is to determine a suitable α.

2.4.2.4. Dropout

A CNN is considered to be an descriptive model because its design consists of numerous non-linear hidden layers; which enables it to understand the complex relationships that exist between the inputs and outputs of the network. However, if training data is limited, many of these complex non-linear relationships will be the result of sampling noise which only exists in the training set and does not exist in the testing dataset, even if both datasets are drawn from the same distribution (Qiu Jiayan, 2016). Then, throughout the process of training, overfitting would take place. There are a number of approaches that have been developed to address this problem, some of which include terminating training as soon as performance on a validation set begins to deteriorate, called early stopping, another approach is implementing a number of different types of weights penalties, and using a method known as soft weight sharing (Qiu Jiayan, 2016). Dropout is considered as a strong technique that was developed to overcome the overfitting issue. Therefore, it helps to decrease the generalization error that large neural networks experience. Because with the dropout method a single neuron is unable to depend on the existence of other neurons, the complexity of neuronal co-adaptations is greatly reduced. Because of this, dropout improves CNN's ability to learn features that are more robust and stable. The process of eliminating neurons (both hidden and visible) from a neural network is referred to as dropout. When we talk about "dropping out" a unit, we refer to temporarily disconnecting it from the network together with all of its connections (Qiu Jiayan, 2016).

Dropout, which can be applied during supervised training using end-to-end backpropagation, may be seen as a mechanism to regularize a CNN by introducing noise

to its hidden units and can be employed in all layers except the loss layer. Dropping neurons is completely at random. As we can see in Figure 2.8 the neural network with dropout vastly decreases the number of trainable parameters in comparison to a regular neural network, which in turn reduces the number of computations and improves computing efficiency. When the CNN is considerably bigger and more in-depth than this sample model, the impact of the dropout technique becomes more apparent.

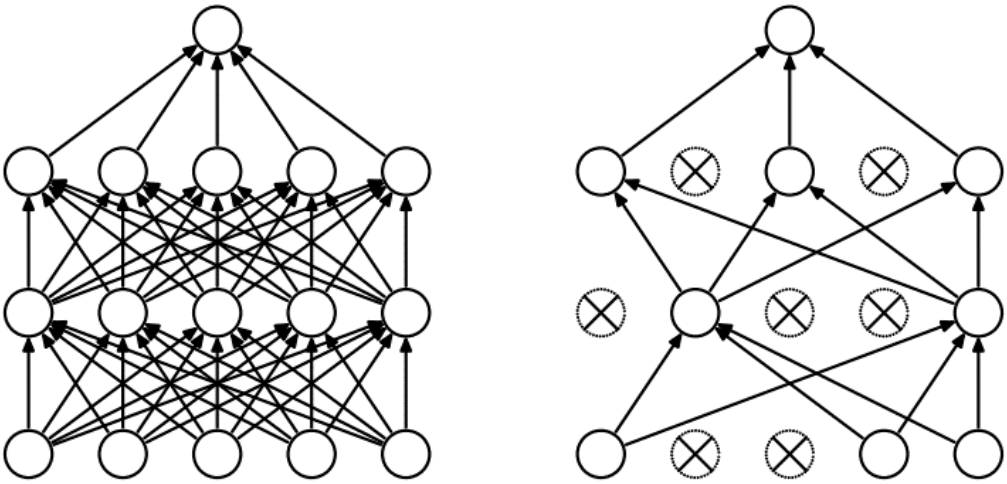


Figure 2.8. *Illustration of CNN structure before and after dropout is applied*

2.4.2.5. Loss Layer

CNN provides researchers with a wide variety of loss functions from which they may choose the appropriate one for the specific images processing task at hand. In this study, the Softmax loss function, which is the most often used loss function for classification tasks, will be implemented. Therefore, our primary focus will be to present this function.

Softmax: In the case of a classification task, softmax loss is used to predict a single class out of K different classes that are mutually exclusive and produces a possibility vector with the size of 1 × k. This vector's total value must be equal to one at all times. The Softmax loss may be determined using the formula in the Equation 2.6:

$$L = -\sum_i y_i \log p_i \tag{2.6}$$

where y_i is the ground truth class and if the target class is a belongs to the i-th class, then y_i is equal to one, and if it is not a member of that class, then y_j is equal to zero. The notation p_i specifies the probability of the input which belongs to class i-th. The

definition of softmax function is used when the predicted possibility vector is calculated. Equation 2.7 is the definition of softmax:

$$p_i = \frac{e^{z_i}}{\sum_j e^{z_j}} \tag{2.7}$$

where z_i indicates the output at i-th position of CNN's last layer.

2.5. Summary

This chapter provided an introduction to topics related to artificial intelligence, machine learning, and deep learning. We defined and described all of the different types of machine learning and deep learning. In addition to this, we included a background on CNN, which details both an introduction to CNN as well as the structure of CNN, along with an explanation of the components that make up this structure.

In the next chapter, we will present age estimation systems based on face images, and we will also discuss a large number of studies that are connected to the framework that we have proposed for the age estimation from facial image task.

3. AGE ESTIMATION SYSTEMS AND RELATED WORK

This chapter is meant to review age estimations systems, where first the typical age estimation system is introduced then the real applications for age estimation system are discussed. After that, an overview of Facial aging datasets is presented. And then, we reviewed some of the state-of-art CNN architectures existing in the literature. Also, the age estimation systems are classified according to their algorithm in the final stage of protection. Finally survey of related works is presented to show the last steps taken in the literature regarding the age estimation task.

3.1. Introduction

Human age is regarded as a key personal attribute that may be directly deduced from the appearance of distinct face patterns. Identifying the age of a person's face is a difficult assignment that faces many of the same difficulties as other facial recognition tasks. Initially, they need to identify the face, then discover the most relevant facial characteristics or features, and then classify the picture based on these features. Automatic labeling of face images with an individual's age (year) or age group (year range) is the technique of Age Estimation using facial image (Al-Shannaq & Elrefaei, 2019).

Additionally, age estimate is considered a supplementary soft biometric since it provides additional information about the user's identification. Biometric traits such as face, fingerprint, and iris might be enhanced and supplemented using this kind of data.

Many real-world applications may be identified for age determination. Age-based access control is one of these uses (Al-Shannaq & Elrefaei, 2019). Age-related limits, for example, are required for a website or mobile app's virtual entryway as well as its physical entrance. More applications will be discussed later in this chapter.

In general, age estimate models are divided into two main categories: hand-crafted algorithms and deep learning. In hand-crafted models, feature extraction and selection is done manually, which is the fundamental difference between the two categories. However, in the case of supervised learning, this process is carried out automatically and with minimum human intervention. The age estimate models may employ two types of datasets: constrained and unconstrained (Al-Shannaq & Elrefaei, 2019).

The constrained datasets can be thought of as the perfect picture that was taken in ideal conditions. Unconstrained datasets, on the other hand, are those whose photographs were taken under uncontrolled environments and which include a wide range of

occlusion, face expression, and head postures, etc. Face recognition from unconstrained photos and videos has recently seen an increase in awareness due to issues such as face detection, facial landmark localization, and age estimation. As a result, age estimation will be more robust to real-world applications when it has been tested on unconstrained datasets (Al-Shannaq & Elrefaei, 2019). A typical age estimation system based on CNN is shown in Figure 3.1.

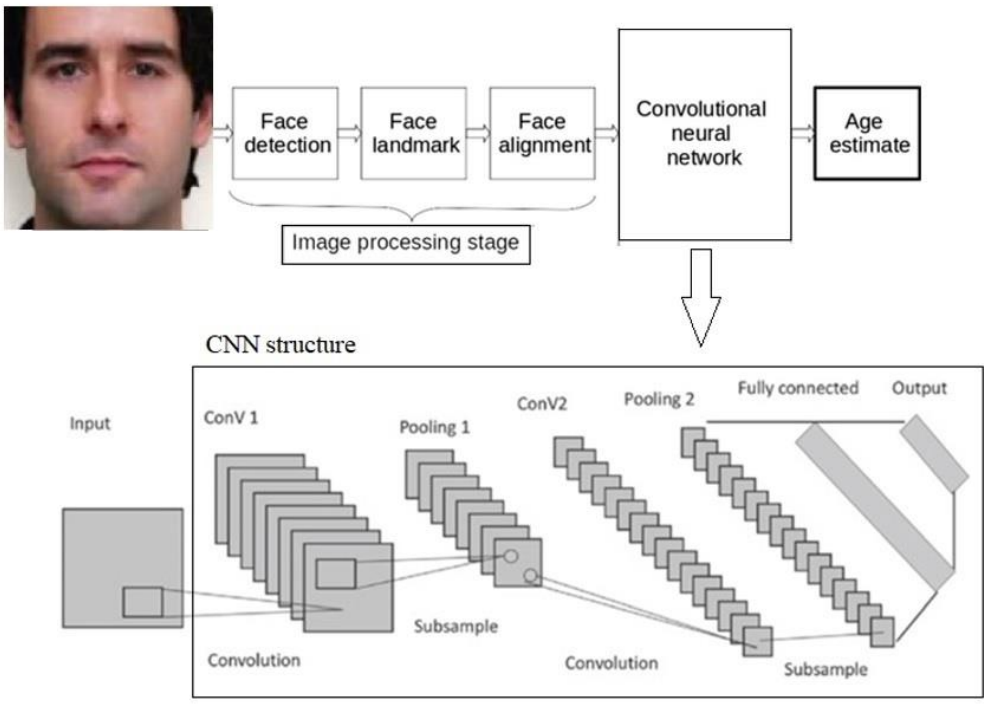


Figure 3.1. *A typical CNN age estimation system (Agbo-Ajala & Viriri, 2021)*

3.2. Real Applications for Age Estimation

A variety of real-time applications might benefit from age estimate system. Facial features might be seen as supplementary to a number of other key features used for identity verification or identification. The following uses are examples of the many contexts in which age estimate may be used as a beneficial technique.

3.2.1. Age simulation

The facial features of the same individual at various ages are necessary for age simulation. This features that have been learned are used to model the face at a certain

point in time. In this way, the age estimate helps this system understand the diverse aging patterns of a person. A lot more information found in Al-Shannaq & Elrefaei (2019).

3.2.2. Electronic customer relationship management (ECRM)

In online commercial systems, clients are contacted virtually. To manage connections and understand the expectations for each client, thus, age estimate may be used to determine the customer's age. In this system, services will be adjusted and customized based on the age of the user (Al-Shannaq & Elrefaei, 2019).

3.2.3. Access control and surveillance

As data becomes more easily obtainable, the need for solutions to access control and surveillance monitoring issues will become more important in our daily lives. An age estimate technique may be useful in stopping children from buying cigars or excluding them from some bars. In addition, age estimate may be used to restrict access to certain websites or movies that are not appropriate for minors. The nurse robot or the intelligent care unit might raise the question of age estimate as a futuristic usage in the health care system. A wide range of services may be provided by these intelligent units that can be adjusted to the patient's age (Al-Shannaq & Elrefaei, 2019).

3.2.4. Age specific human computer interaction (ASHCI)

People in various age groups have distinct needs. Automatic age estimate is a great tool for identifying the demands of individual users based on their facial age group, which may be used to tune certain age-related interface settings (Al-Shannaq & Elrefaei, 2019).

3.2.5. Biometrics

One of the secondary soft biometrics that provides additional information about a person's identification is age estimate (Al-Shannaq & Elrefaei, 2019). Main biometric data like face, fingerprint, and iris may be supported with age estimate data, which can increase performance of a system.

3.2.6. Missing persons

In identification system, age estimate using age simulation may aid in identifying a person who is lost for a period. By determining the person's age based on his or her most recent photos and comparing them to older ones (Al-Shannaq & Elrefaei, 2019).

3.3. An Overview of Facial Aging Datasets

A comprehensive overview of the most prevalent and accessible datasets with age annotations is provided. For accurate age estimate, a big, balanced, and labeled database is required. In actuality, it is incredibly difficult to gather such aging datasets, and it is much more difficult to collect a sequence of photos for a single human at various ages. Next, a lot of datasets are presented to get a good idea of what dataset we have for the age estimation task.

FG-NET is a publicly-available aging database that was collected in 2004 to facilitate studies on the aging of individual's face. In this dataset, there are 1,002 photos of 82 distinct people, ranging in age from 0 to 69 years with average number of photos for each person is 12. Because these face photos were scanned from personal photographs, they show a wide range of illumination, image quality, expression among others. Additionally, the dataset contains 68 landmark points that may be used to model face features. Occlusions of various shapes and sizes may also be seen in the photos. The dataset can be found on the internet (Panis et al., 2016a).

MORPH (Ricanek & Tesafaye, n.d.) is a publicly accessible aging dataset gathered by the face aging group at the University of North Carolina at Wilmington. It has labels such as gender, ethnicity and weight and height are included in the data collection. The entire dataset has two main albums; album-I and album-II which in album-I has a total of 1,724 face images of 515 individuals aged between 27 and 68 years with 1430 images of males and 294 images of females, while in album-II includes 55,134 facial images gathered from 13,618 individual collected over four years.

Iranian Face Database (IFDB) (Azam Bastanfard et al., 2007) is a publicly-available dataset that was gathered and utilized by the department of engineering in "The University of Karaj" to study different research activities related to face such as race detection, age estimation and facial surgery. The dataset comprises a total of 3600 color images belonging to 616 individuals which are divided into 487 for males and 129 for female and with age labels ranging from 2 to 85. The facial images were captured in a controlled environment including variations of head pose, expression, without glasses with a 640×480 pixel resolution images.

Human & Object Interaction Processing (HOIP) (Yun Fu et al., 2010) dataset has 300 individuals' photos with total number of images about 300k whose age range is between 15 and 64 years. The ages are separated into 10 age groups and individuals are

distributed equally between the age groups with 5 years interval in the total age, which results in 30 individuals in each group, 15 male images and 15 female images.

Biometric Engineering Research Center (BERC) (Choi et al., 2011) dataset is a face aging dataset gathered and published by "Biometric Engineering Research Center" (BERC). The collection includes of facial photos of 390 individuals or subjects having ages spanned from 3 and 83 years. The dataset has very high-resolution photos without changes in facial expression and illumination and these photos of the individuals are uniformly distributed with their age and gender.

OIU-Adience (Eidinger et al., 2013) dataset was constructed to produce a classification based on the age group and gender and obtained from "Fliker.com" albums. It has 26,580 images belong to 2,284 subjects and with eight age-group labels distributed as follows: 0–2, 4–6, 8–13, 15–20, 25–32, 38–43, 48–53, 60+. it is considered an unconstrained dataset because it is collected from an ideal real-life and unconstrained environment with its poor resolution, excessive blurring and occlusions, as well as a variety of head positions and emotions, this dataset provides significant hurdles. It is available on the internet.

Cross-Age Celebrity Database (CACD) (B.-C. Chen et al., 2015) is a large-scale publicly-available dataset that was developed for face recognition and age retrieval tasks and collected from the Internet. This database has a collection of over 160,000 face images belonging to 2,000 celebrities with a range of ages from 16 to 62. The dataset is available online.

Asian Face Age Database (AFAD) (Niu et al., n.d.) is a big and publicly-available dataset was gathered by (Niu et al., n.d.) which comprises over 160K face photos. The dataset was designed to assess the age estimate performance and obtained from selfies shot by Asian students on a social network which resulted in the photos featuring solely Asian faces. The ages span a range from 15 to 40 years and the total 164,432 photos distributed to 63,680 photos belong to female and 100,752 belong to male faces.

IMDB-WIKI (Rothe et al., 2018a) dataset is regarded as the biggest dataset that is publicly available and was gathered and applied to handle the age estimation task in the wild. It comprises more than 500k images with accurate age labels between 0 and 100 years. This dataset has face photos labeled with gender and age annotations with total 523,051 images of 20,284 subjects. The dataset is collected by crawling images from IMDB and Wikipedia websites and since the images collected from websites, they will

had low-quality images, full body images, multi-person images, blank images, and without metadata about the date when it was taken.

APPA-REAL (Agustsson E et al., 2017) dataset is the first dataset including both real and apparent age labels which gathered using labeling application from many resources such as “crowd-sourcing” data collection and “AgeGuess platform”, and with the assistance of “Amazon Mechanical Turk” (AMT) workers. The real and apparent age labels are provided for 7591 images in age range between 0 and 95 and the samples were taken under different conditions. This makes the dataset more challenging for recognition purposes.

AgeDB dataset was gathered to study different task related to face such as age estimation, age-invariant face verification and face age progression. Images of human faces were gathered by hand, and as a result, there is some degree of variation that is to be anticipated when a picture is obtained in an uncontrolled environment, such as variations in expressions, pose, occlusions etc. AgeDB comprises 16,488 images belonging to 568 individuals with their identity, age and gender annotations and age range between 1 and 101 years (Moschoglou et al., 2017).

UTKFace (Berg et al., 2020) dataset was gathered to study different task related to face such as age estimation, face detection, landmark localization and age progression/regression. The dataset is a large dataset with over 20k face photos with age, gender, and ethnicity age range between 0 and 116 years covering large variation in facial expression, pose, illumination, resolution, occlusion, etc.

Facial-age (WIKI_ART, 2019) dataset is gathered by WIKI_ART company and comprises 9,778 colorful images of faces in PNG format with a resolution of 200x200 pixels each. The photos are split into folders and the folder names match to the age labels of images within those folders. The dataset could be useful for studies related to human face such as age estimation.

Table 3.1 summarizes the previously mentioned databases, arranged according to their publication date. For each dataset, we identify whether the images are acquired under controlled conditions (constrained) or uncontrolled conditions (unconstrained). A portion of these datasets are labeled with actual ages, while the rest have age group labels. FG-NET and MORPH-II (Al-Shannaq & Elrefaei, 2019) are the most often used datasets for evaluating the performance of various age estimation models.

Table 3.1. *Summary of the presented facial aging databases*

Dataset	Size	Subject number	Age range	Age type	Year	Uncons-trained
FG-NET (Panis et al., 2016b)	1002	82	0-69	Real	2004	Yes
MORPH-II (Ricanek & Tesafaye, n.d.)	55,134	13,618	16–77	Real	2006	No
Iranian face (Azam Bastanfard et al., 2007)	3600	616	2–85	Real	2007	Yes
HOIP face (Yun Fu et al., 2010)	306,600	300	15–64	Age group	2010	No
BERC (Choi et al., 2011)	5910	95	3–83	Real	2011	No
OIU-Adience (Eidinger et al., 2013)	26,580	2284	0–60+	Age group	2014	Yes
CACD (B.-C. Chen et al., 2015)	160,000+	2000	16–62	Real	2015	Yes
AFAD (Niu et al., n.d.)	164,432	N/A	15–40	Real	2016	Yes
IMDB-WIKI (Rothe et al., 2018a)	523,051	20,284	0–100	Real	2016	Yes
APPA-REAL (Agustsson E et al., 2017)	7591	7000+	0–95	Real & apparent	2017	Yes
AgeDB (Moschoglou et al., 2017)	16,488	568	1–101	Real	2017	Yes
UTKface (Berg et al., 2020)	20,000+	N/A	0–116	Real	2017	Yes
Facial-age (WIKI_ART, 2019)	9,778	N/A	1-110	Real	2019	No

3.4. A review of state-of-the-art CNN architectures

When it comes to face recognition and age estimation, the most recent studies has used CNN models, which is a type of deep learning model that works with image and video data, known as convolutional neural networks (CNN or ConvNet). The capability of CNN to learn facial discriminative features directly from photo pixels has recently proven a promising performance in feature learning, age estimation and face recognition (Agbo-Ajala & Viriri, 2021). To appropriately determine the age of persons from facial images these features are required.

As seen in Figure 3.1, the fundamental structure of CNNs comprises essentially of three layers: the convolution layers, the subsampling layers, and the classification layers. The convolutional layers extract the local patterns and may improve the original signal while reducing the noise. In the subsequent subsampling step, the features are re-extracted from the convolutional layers. The motivation behind this layer can be divided into three

objectives: First, the subsampling operation helps into disregard non-maximal values, second it helps in reducing the previous layer calculation and preventing overfitting and finally additional enhancements may be made to the distortion tolerance of this layer while offering robustness to position. In classification layers, the received features from the preceding layers will be encoded into a one-dimension vectors and then a trainable classifier puts these vectors into classes (Al-Shannaq & Elrefaei, 2019).

Recent publications in the area of age estimation used CNN as one of their model components, either during the feature extraction or classification / regression phases. In this section, we have presented the most common architectures which achieved good results on different tasks including age estimation task.

3.4.1. AlexNet architecture

For age estimation, one of the early CNN architectures is AlexNet, which is named after its designer Alex Krizhevsky (Krizhevsky et al., 2017). This network was the winner with the first-place in the 2012 edition of ImageNet Large Scale Visual Recognition Challenge (ILSVRC) (Agbo-Ajala & Viriri, 2021). As shown in Figure 3.2, the structure has in total eight layers, five of them are convolutional layers and the other three are fully connected layers. Max-Pooling layers follow some of the convolutional layers while ReLU layer follow every layer in the structure. Most often, it's used to tackle a variety of difficult face analysis problems, such as age estimation. Nevertheless, AlexNet is surpassed by more complex models like ResNet and GoogleNet, although it is computationally costly to use. AlexNet reports a top 5 error rate of 16.4% (Punyani et al., 2020).

3.4.2. Visual geometry group (VGGNet-16) architecture

Simonyan and Zisserman develop the VGG-16 architecture (Simonyan & Zisserman, 2014). The VGG-16 architecture consists of 16 convolutional layers, 5 Max-pooling layers, and 3 fully connected layers each of these hidden layers followed by a ReLU layer and an output of this structure is a SoftMax layer as shown in Figure 3.3. This architecture's simplicity is one of its most attractive features. This very massive network design employs a total of 138M parameters, yet is nonetheless commonly utilized for research due to its simplicity and uniformity. VGG-Net's top five error rate is 7.32%, which is much lower than Alex-Net (Punyani et al., 2020).

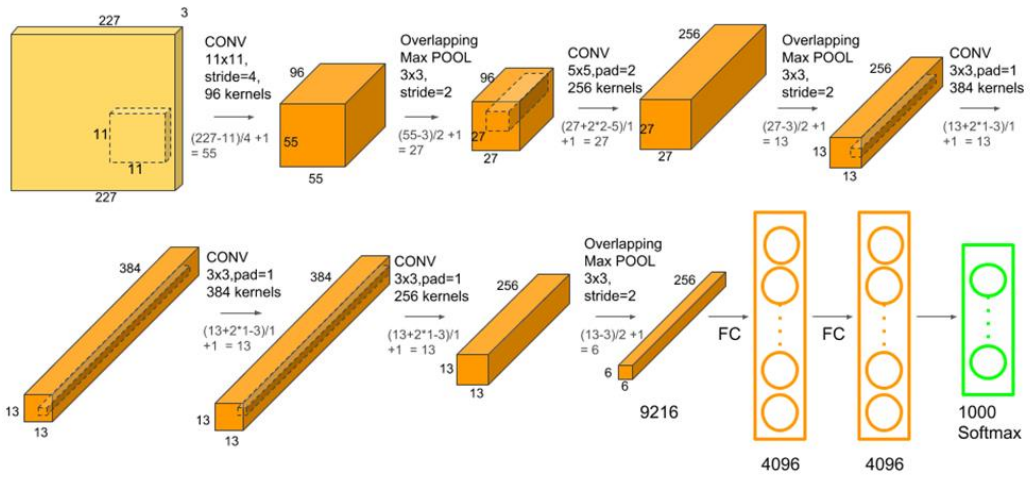


Figure 3.2. *The AlexNet structure (Krizhevsky et al., 2017)*

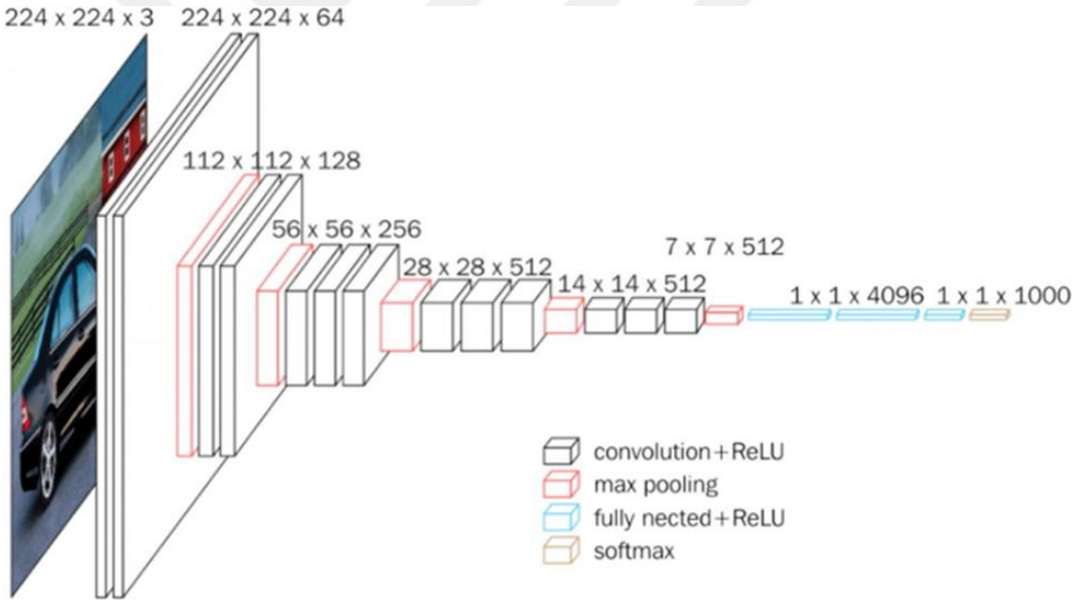


Figure 3.3. *VGG-16 architecture (Simonyan & Zisserman, 2014)*

3.4.3. GoogLeNet architecture

The authors (Szegedy et al., 2015) made a compromise between the Alex network and the VGG network, designing a structure that is both deeper and wider than the AlexNet yet lighter than the VGG architecture which led to a structure that outperformed both AlexNet and VGG network in the 2014 edition of the “ILSVRC” competition (Agbo-Ajala & Viriri, 2021).

The authors took advantage of a concept called "network in network" or "micro-architecture" and designed a micro-architecture, which they then called the inception layer. By repeating this inception layer nine times as shown in Figure 3.4, the overall model is formed with the addition of some Max-pool layers after some of the Inception layers to reduce the parameter size. In GoogLeNet architecture shown in Figure 3.5, each inception block consists of different convolutional filters of sizes 1×1, 3×3, and 5×5. As the network gets deeper, the 1×1 convolution layers perform the function of reducing feature dimensionality (Punyani et al., 2020). Both Alex-Net and VGG-top Net's top 5 error rate are decreased to 6.67% using GoogLeNet architecture with 22 convolutional layers.

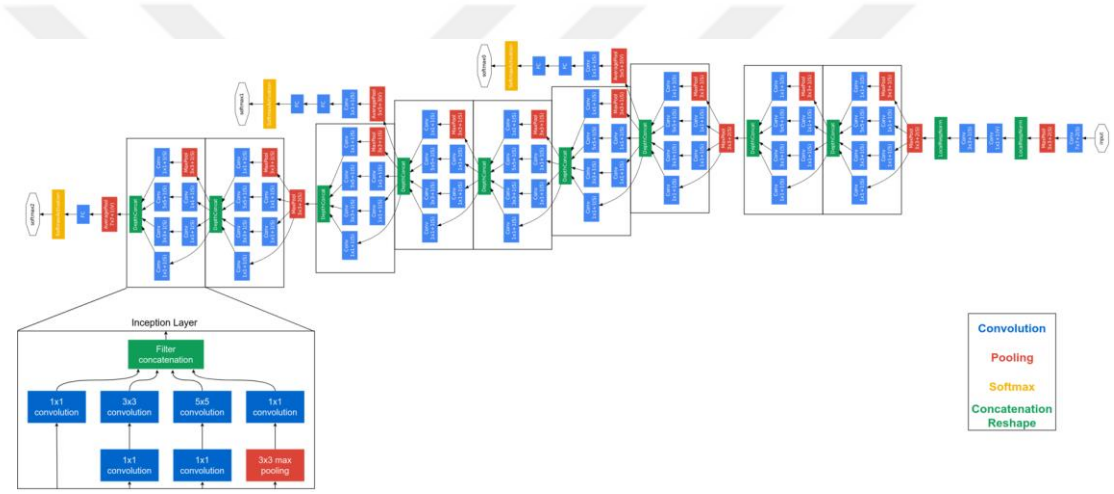


Figure 3.4. *Overfiew of GoogLeNet architecture (Szegedy et al., 2015)*

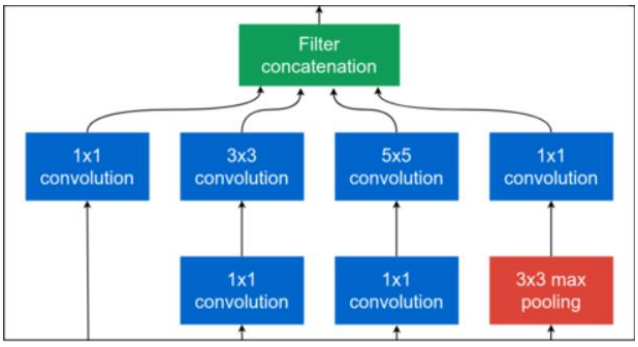


Figure 3.5. *The inception layer (GoogLeNet Architecture, n.d.)*

3.4.4. ResNet architecture

There are many problems that researchers face when working on very deep neural networks like vanishing gradient and higher training error. These problems motivated researchers (He et al., 2016) to work on overcoming these difficulties and inventing new neural networks with greater depth, where they used the concept of "skip connections" in ResNet architecture that allow access to deeper neural networks without any issues. Using identity mapping to generate a route between the input and output layers and skipping certain levels in between, a ResNet architecture has been developed with a variety of ResNet models such as ResNet-18, ResNet-34, ResNet-50, ResNet-101, and ResNet-152. The architecture of ResNet-18 is shown in Figure 3.6, so that the framework may be comprehended with ease. The top 5 error rate for ResNet-34 is 5.71%, which is much better than Inception and VGG-16 architectures while the ResNet-50 architecture achieves the best performance on top 5 error rate with a value of 4.49% (Punyani et al., 2020).

3.4.5. Xception architecture

Xception is a deep convolutional neural network architecture introduced by Francois Chollet (Chollet, 2016) that takes the ideas of Inception layer to an extreme level that is why it is called Extreme inception or Xception.

Basically, Xception architecture utilizes a special type of convolution layer called "Depth-wise Separable Convolutions" with “residual or skip connections” throughout the entire structure. As shown in Figure 3.7, it comprises 36 Depth-wise Separable convolutional layers structured into 14 blocks or modules, all of which include linear residual connections, or skip connections, around them, except for the first and last blocks.

The Xception architecture showed up a performance that outperformed not only the "Inception V3" architecture but also the “ResNet-50”, “ResNet-101” and “ResNet-152” architectures on the “ImageNet” dataset (Agbo-Ajala & Viriri, 2021), but it suffers from a noticeable slowness compared to the Inception architecture.

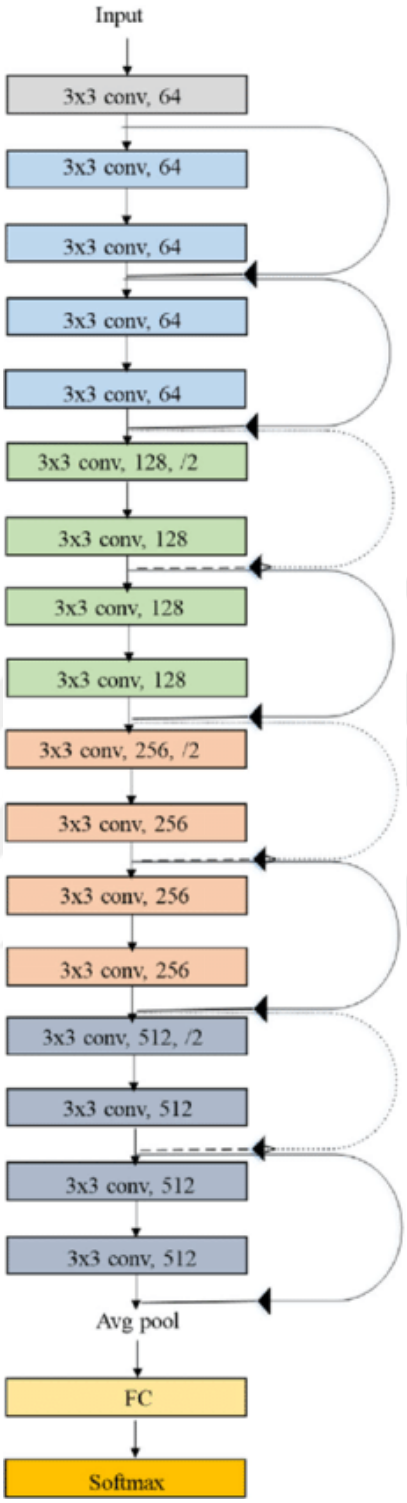


Figure 3.6. *Original ResNet-18 Architecture (Original ResNet-18 Architecture, n.d.)*

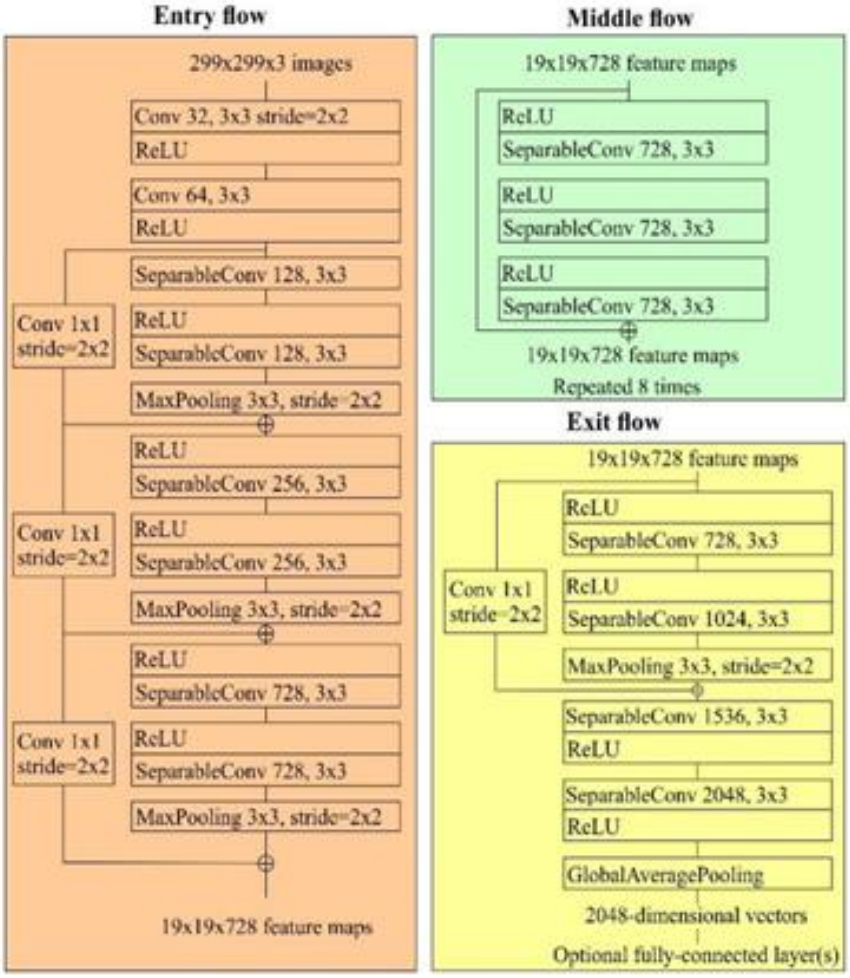


Figure 3.7. *Xception modules and architecture (Chollet, 2016)*

3.5. Age Estimation Algorithms

Facial age estimate methods and algorithms can be categorized into five categories which will be discussed in this section.

3.5.1. Classification-based methods

In this category, the total age range is divided into independent multi-classes (as an example age range from 0 to 99 can be viewed as 100 independent class) or a group of ages can be considered as independent classes or multi-classes where the individuals can be classified by an age classifier into a real age class or an age group class. Multi-class classification methods do maximize the probability of certain class label without considering other classes which in insufficient and imbalance datasets can lead to overfitting problem (Agbo-Ajala & Viriri, 2021).

3.5.2. Regression-based methods

In this approach, the age can be seen as numerical values rather than as a multiclass problem. The regressor is trained to use the proper regularization approach to map the age-value space to the feature space. Although it is very typical to consider age estimation as a regression task, which reduces the Mean Absolute Error (MAE) result and improves the accuracy of the estimate performance this approach leads to unstable training mode resulting in significant error terms, which negatively impacts accuracy (Agbo-Ajala & Viriri, 2021).

3.5.3. Ranking-based methods

The ordinal ranking idea has recently gained substantial traction in the field of age estimation, given the fact that human aging manifests itself differently at various ages. For example, age of a person can be classified to one of the adjacent ages rather than ages which are far from the neighboring. The relative order of the age range is not resolved by the multi-class and regression techniques, which treat it as uncorrelated and independent information. While in the case of ranking-based approach, it employs the relative order of the age and using relative age ranks instead of real age labels and ranks age class labels in the descending order using their relevance to the presented face images (Agbo-Ajala & Viriri, 2021; Al-Shannaq & Elrefaei, 2019).

3.5.4. Hybrid-based methods

The hybrid-based approach, takes the advantages of both classification and regression methods and builds a model by using a parallel or hierarchical arrangement of many algorithms, to improve the performance and obtain more robust system. Unfortunately, this combination may be inapplicable in resource-constrained machines due to large storage and computational cost.

3.5.5. Soft classification-based methods

A soft classification approach also referred as "deep label distribution learning" (Agbo-Ajala & Viriri, 2021) in other papers, can be viewed as a middle ground between classification and regression. It works by converting real-value age to a discrete-age distribution to fit the entire age range (Al-Shannaq & Elrefaei, 2019).

This approach solves the problem of insufficient training data experienced in most age estimation task while suffers from a lack of consistency between the employed evaluation metric and the training goals, therefore, produce an undesirable result (Agbo-Ajala & Viriri, 2021). Figure 3.8 shows the difference between classification, regression and soft classification as an algorithm in term of age encoding.

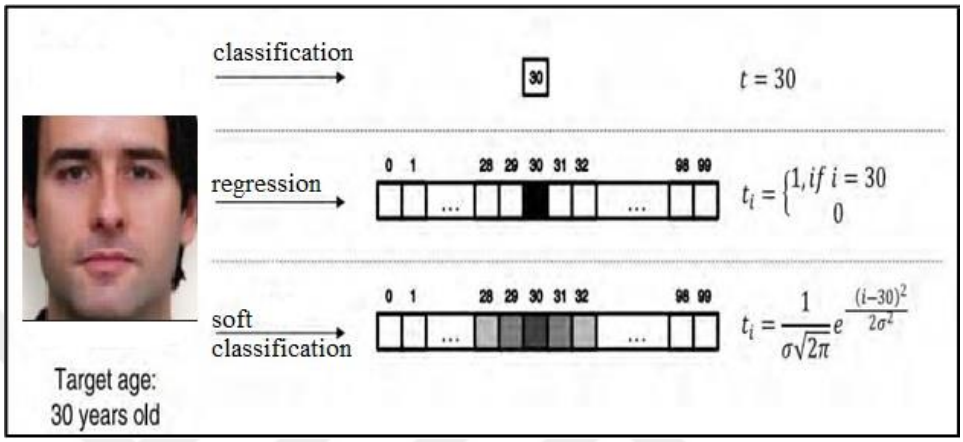


Figure 3.8. *The difference between classification, regression and soft classification as an algorithm in term of age encoding (Al-Shannaq & Elrefaei, 2019)*

3.6. Related Works

In 2015, a structure of three convolutional layers and two fully connected layers were designed by Levi & Hassncer (2015), to learn the representations of facial feature from facial images with the aim of reducing overfitting which may occur when training data is limited. At that time the availability of large datasets on face aging doesn't readily exist. A "dropout regularization" approach and "data augmentation" techniques were also used in an effort to prevent overfitting. When they evaluated their work on the difficult adience benchmark for age estimation, their architecture surpassed state-of-the-the-art approaches at their time with an exact accuracy of 50.7 ± 5.1 and 1-off accuracy of 84.7 ± 2.2 despite the simplicity of their architecture.

"AgeNet" is an "end-to-end learning method" presented for apparent age estimation by Liu et al. (2015). The developed network solved the issue of apparent age estimation by combining a classification model based on "Gaussian label distribution" with a regression model based on "actual value". To learn the representation of facial features for the two models, a large-scale deep CNN was used. As a solution to the issue of overfitting, they proposed a deep transfer learning strategy and as a result the

experiment result was awarded second place at the "ChaLearn 2015" competition with MAE is 3.3345 and MNE is 0.2872, proving that the (AgeNet) architecture had achieved the state-of-the-art performance in apparent age estimation task.

Deep convolutional neural networks were used (Ranjan et al., 2015) to perform age estimation task from unconstrained facial photos with multi-stage process consist of face detection, face alignment, deep facial features extraction, and a three-layer neural network regression. For the age estimation challenge, the authors used a "gaussian loss function" with a "3-layer neural network regression" model and a "hierarchical learning" strategy to acquire features from the pre-trained DCNN model and as a result the model achieved a better performance than the conventional “linear” model for age estimation.

Deep convolutional neural networks (CNN) are used in the approach that was developed in Huo et al. (2016), which uses "distribution-based (KL divergence) loss functions". The architecture is made up of two deep CNNs with distinctive architectural designs running in parallel as two streams. These CNNs are VGG-16 and a new architecture. The VGG-16 was fine-tuned on three distinct datasets, and the second model, which was used to train the innovative CNN model, made use of a variety of inputs and a variety of various augmentation techniques. Before performing the final tuning on the competition dataset, each deep CNN model performed preliminary training on a variety of other datasets. They need to combine the results from two different models in order to arrive at the final predicted ages. As a consequence of this, their strategy got an error of 0.3057, which was good enough to win them the fourth place finish in the "ChaLearn 2015 apparent age competition." Because it trained on an additional 119,539 face photos in addition to existing public facial datasets, the model was able to make a more accurate prediction as a result.

Antipov et al. (2016), came up with a solution by using a VGG-16 CNN that had been pretrained. They began by training the network using the massive IMDB-WIKI dataset and then proceeded to fine-tune it using the little dataset that was supplied during the competition. They demonstrated that the most important aspect of the competition is providing an accurate age estimate for children. In light of this, researchers created a distinct VGG-16 network for children between the ages of 0 and 12 and trained it for apparent age estimate. This "children network" was then isolated from the "general network." For the training of the "children" network and the "general" network, they used two distinct "age encoding" strategies: the strict one for the "children" network and "label

distribution encoding" for the "general" network. The result of the experiment was successful enough to gain first place in the ChaLearn LAP apparent age estimation competition that took conducted in 2016.

In their study Aydogdu & Demirci (2017), the authors presented a "optimized deep CNN" architecture for the problem of age estimation. The suggested design for a CNN includes four convolutional layers and two fully connected layers. The performance of the architecture was evaluated using a MORPH-II database, and the researchers found that it surpassed existing CNN designs when the "exact success," "top-3," and "1-off" criteria as well as the "standard deviation" values were applied.

S. Chen et al. (2017), created a unique CNN-based architecture called "ranking-CNN" for age estimation. This design was based on the ranking technique. A series of fundamental CNNs have been trained on "ordinal age labels" and the final age estimation is based on the collection of the binary outputs that are generated by the CNNs. They proved that their suggested approach resulted in fewer estimation errors when compared with "multi-class" classification approaches by conducting a extensive empirical experiments to back up their claims. As a consequence of this, the "ranking-CNN" technique performed noticeably better than other age estimation models when applied to benchmark datasets.

In addition, the authors of the study referred in (Qawaqneh et al., 2017) used a VGG-Face network model and fine-tuned it on a dataset designed for the task of identifying faces. Eight convolutional blocks followed by three fully connected layers is forming a total 11 layers which make the structure of the model of the deep CNN architecture. The convolutional blocks are made up of convolutional layers followed by rectification layer, and then "max-pool" layer is included in each convolutional block. The research also evaluated a "GoogLeNet" architecture. The "GoogLeNet" architecture was not able to compete with the suggested VGG model despite the fact that it was trained on a much larger dataset that included millions of training pictures. The VGG-Net CNN received further finetuning and improvement so that it could fulfill the task of age estimation.

Zhang et al. (2017), introduced a novel CNN-based technique called "Residual Networks of Residual Networks (RoR)" for estimating age groups and genders from images taken in uncontrolled environments. In order to improve the RoR model's capacity for learning from face photos, it was first pretrained using the ImageNet dataset.

Subsequently, it was fine-tuned using the IMDB-WIKI-101 and adience datasets. They examined the effectiveness of the suggested RoR approach for estimating ages and genders using a widely used benchmarks.

Moreover, (Rothe et al., 2018b) employed a deep learning approach called "Deep EXpectation" (DEX) to tackle the problem of actual and apparent age estimation from a single face picture without the usage of facial landmarks. DEX is built on the VGG-16 architecture. In addition to that, they presented the IMDb-WIKI dataset, which is the biggest public collection of face images along with annotations of age and gender. In order to gain better performance, the DEX model was first pre-trained on both the "ImageNet" dataset as well as the "IMDB-WIKI" dataset. Before calculating an expected value for age regression, DEX applied a robust face alignment on substantial amount of data. They used common benchmarks such as MORPH-II, FG-NET, and LAP2015 to verify the DEX approach, and the results showed that it attained a state-of-the-art result for both the real and apparent age estimation.

A deep CNN-based model that can rebuild low-resolution faces as high-resolution faces was used to tackle the challenge of age estimation from low-resolution facial pictures (Nam et al., 2020). This model was able to solve the problem of low-resolution images. Before being put to use as input, low-resolution face images were run through a conditional generative adversarial network (GAN) that was part of the solution that was based on CNN. After that in order to determine an individual's age based on their rebuilt face pictures, the model relied on most successful CNN network architectures such as ResNet, VGG, and DEX. The usefulness of the suggested approach in high-resolution reconstruction was tested on PAL, MORPH, and FG-NET databases, and the experimental results demonstrated this effectiveness. The results it gets in age estimation from low-resolution photos are considered to be state-of-the-art.

Agbo-Ajala & Viriri (2020), have recently created a CNN-based model that can classify unconstrained real-life face photos into classes of age groups and gender. The method included of a preprocessing method for images, which was responsible for preparing the input images, and a CNN architecture, which was responsible for performing feature extraction as well as the classification of the images according to their age and gender group. The experimental results were assessed on the OIU-Adience dataset, and the evaluation shows that their method is effective. In fact, their method outperformed previous studies that were done on the same dataset.

4. PROPOSED AGE GROUP CLASSIFICATION MODEL

The purpose of this chapter is to illustrate and discuss the framework that we have created in order to deal with the age estimation task where we will first show in detail the framework of the suggested age estimation system. Then, the motivation and challenges behind the proposed model are discussed. After that, the proposed architecture for a deep CNN is presented and discussed with its details.

4.1. Introduction

The Figure 4.1 illustrates the framework of the proposed age group classification model. It can essentially be divided into four stages: **(1) Combining datasets** which includes merging UTKFace dataset with Facial-age dataset to produce a combined dataset that can be used in the later stages of the process. **(2) Dataset preparation** during this stage, we will be performing three primary processes, which include assigning age group labels, splitting the combined dataset into a Train, Validation, and Test dataset, and finally performing data augmentation on the combined dataset. **(3) Image preparation** here, we have four main processes will be done on each image in the combined dataset these processes are firstly implementing one-hot encoding on each image, after that we are resizing images to the size we need, then we are performing normalization operation, and finally we are collecting a group of images forming a mini-batch which in turn will be fed into the next stage. **(4) Age group prediction** here we have trained our proposed deep convolutional neural network (DCNN) to extract facial features from facial images and we have utilized it to predict the actual age group for each sample of the provided image dataset. All these stages and processes will be discussed in detail in this chapter.

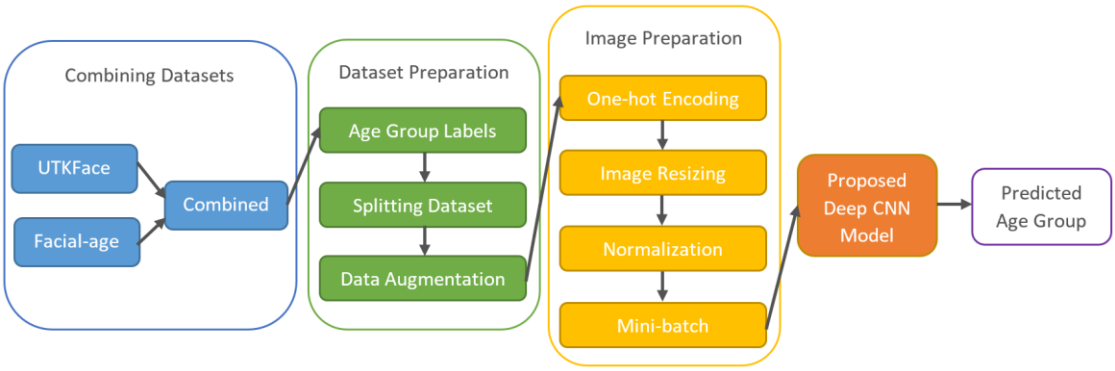


Figure 4.1. *The framework of the proposed age group classification model*

4.2. Combining Datasets

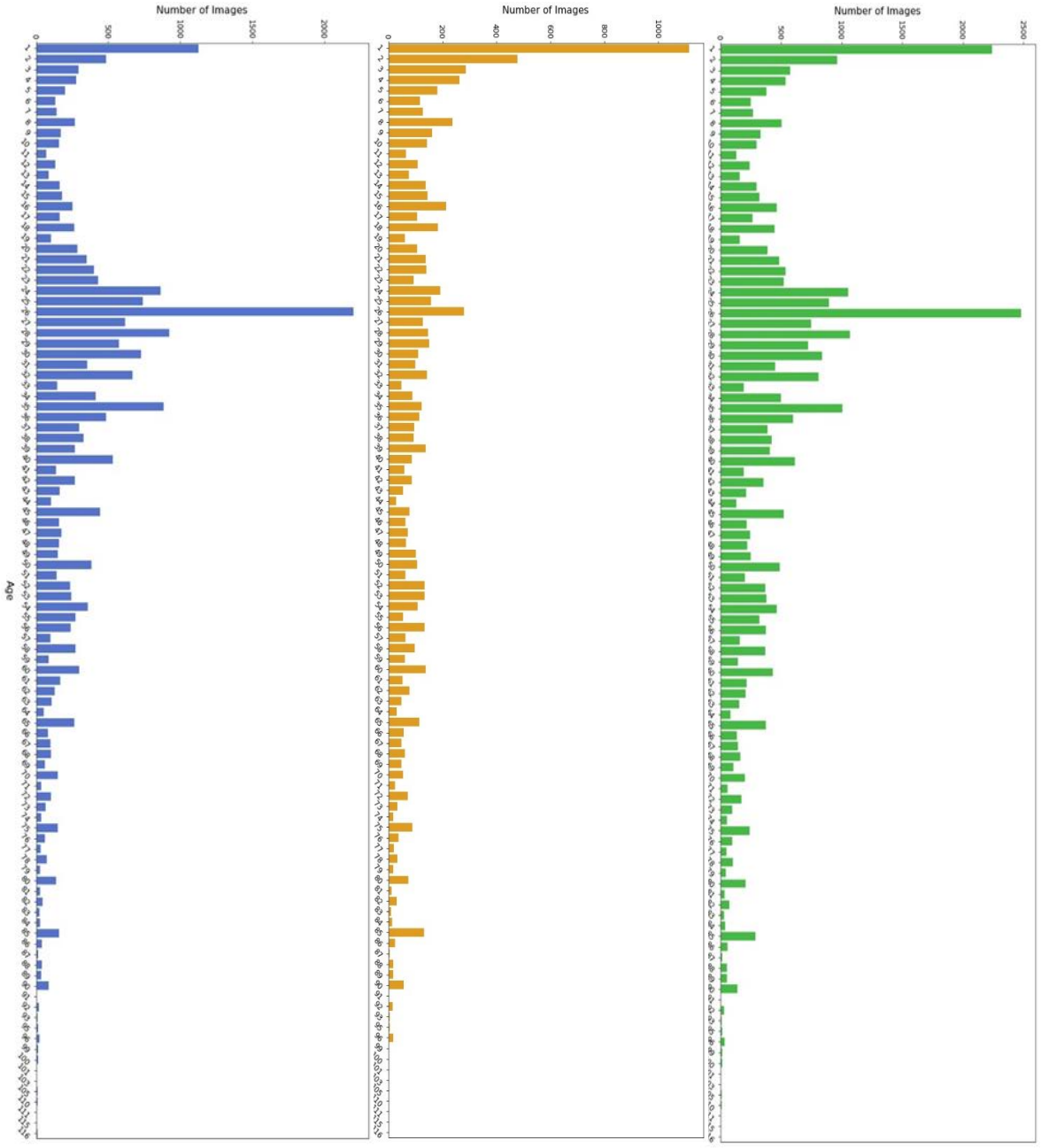
A large number of the most important facial aging datasets that contain age labels can be found on the internet which have been discussed in the third chapter of this thesis. Facial-age and UTKFace were the datasets we chose to use in our research (UTKFace-Cropped album of UTKFace dataset we utilized). Both datasets save a lot of time allowing us to skip the most time-consuming data cleaning stage becuase they contain already cleaned and properly labeled images. After we have chosen them, we have combined these two different datasets in order to produce a larger data set that is capable of satisfying the requirements of the model that we have proposed. The following are the details of both UTKFace and Facial-age dataset and the resulting combined dataset:

UTKFace dataset was gathered to study different tasks related to face such as age estimation, face detection, landmark localization and age progression. The dataset is a large dataset with 23,708 RGB face images with long age range between 0 and 116 covering large variation in facial expression, pose, illumination, resolution, occlusion, etc. All the images are formatted in JPG format and have a resolution of 200x200 pixels with age, gender, and ethnicity labels. These labels are embedded in the file name, and they are formatted like this, [age]_[gender]_[race]_[date&time].jpg, where the meanings of the labels are as follows: **age** label is a number from 0 to 116, which specifies the individual age, **gender** label is either 0 (male) or 1 (female), **race** label is a number between 0 and 4, determining White, Black, Asian, Indian, and Other races (like Hispanic, Latino, Middle Eastern), and finally **date&time** label is in the format of yyyymmddHHMMSSFFF, showing the date and time an image was collected to UTKFace dataset.

Figure 4.2a shows the number of images distributed within the range of ages in UTKFace dataset. This dataset has two albums: “In-the-wild Faces” Album and “Aligned & Cropped Faces” Album. The “In-the-wild Faces” album has the original images before it is used to produce the “Aligned & Cropped Faces” Album after a cleaning and cropping process. The “Aligned & Cropped Faces” Album is utilized to train the proposed model. Figure 4.3a shows an example face images from “In-the-wild Faces” album while Figure 4.3b shows example images from “Aligned & Cropped Faces” album.

Facial-age dataset is gathered by WIKI_ART company which comprises 9,778 colorful face images in PNG format with a resolution of 200x200 pixels each. The photos are organized into folders where each folder name matches the age label for the images

within it. Figure 4.2b shows the number of images distributed within the range of ages in Facial-age dataset. Figure 4.4 shows example images from Facial-age dataset.



(a) *(b)* *(c)*

Figure 4.2. *The number of images in each age year (a) UTKFace dataset (b) Facial-age dataset (c) combined dataset*

(a)

(b)

Figure 4.3. *Example images from UTKFace dataset (a) "In-the-wild Faces" album (b) "Aligned & Cropped Faces" album*

Figure 4.4. *Example images from Facial-age dataset*

Combined dataset is a result of combining both UTKFace and Facial-age dataset which comprises over 33k images with same resolution of the mention datasets. Figure

4.2c shows the number of images distributed within the age range for each age year in combined dataset.

4.3. Dataset Preparation

At this stage, and after we have combined the two datasets into one, this stage includes three basic operations, which are defining age groups and assigning it to each image within the dataset, splitting the combined dataset into three datasets which are training, validation, and testing datasets and increasing the size of the training dataset to meet the requirements of our model. All previous operations will be explained in detail.

4.3.1. Age groups labels

In our work, we have treated the age estimation task as a classification task because the available resources to us were very limited which will be specified in chapter 5. So, in our age group classification, we have divided the age range into eight basic groups, taking into our consideration the most common dataset benchmarks available in the literature of age group classification. These age groups are 0-2,3-7,8-14,15-20,21-27,28-45,46-65,66+ and the corresponding labels or classes respectively are 0,1,2,3,4,5,6,7. The Table 4.1 shows each age group with the corresponding class (label) and number of images in each dataset. Figure 4.5 shows the percentage of number of images for each age group in combined dataset.

Table 4.1. *Each age group with the corresponding class (label) and number of images in each dataset*

Age group	Class (label)	UTKFace	Facial-age	Combined
0-2	0	1605	1587	3192
3-7	1	1028	964	1992
8-14	2	1018	916	1934
15-20	3	1226	800	2026
21-27	4	5572	1119	6691
28-45	5	7646	1710	9356
46-65	6	3914	1684	5598
66+	7	1699	998	2697

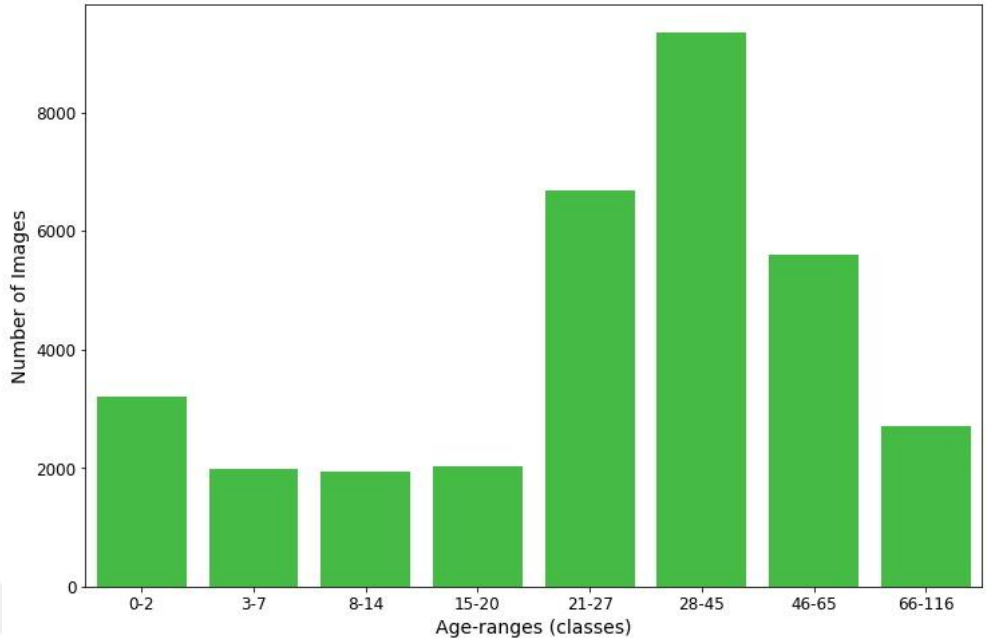


Figure 4.5. *The percentage of number of images for each age group in Combined dataset*

4.3.2. Splitting dataset

In the process of developing any artificial intelligence system, it is necessary to divide the dataset into more than one dataset. As a standard approved in the literature of developing artificial intelligence models, the dataset must be divided into three parts: training dataset, validation dataset (or development dataset), and testing dataset. In our project, we divided our combined dataset as follows: 70% for the training dataset, 22.5% for validation dataset, and 7.5% for the testing dataset. Also, we maintained the classes balance of the combined dataset so that each divided dataset carried the same class balance within the original dataset. This process of dividing dataset with maintaining the classes balance is called stratified splitting. Table 4.2 shows the number of images and the percentage of each dataset.

Table 4.2. *Number of images and the percentage of each dataset*

Dataset	Size (%)	Number of images
Training dataset	70	23440
Validation dataset	22.5	7534
Testing dataset	7.5	2512
Combined dataset	100	33486

4.3.3. Data augmentation

The training dataset is relatively small because we will be training our model from scratch, so increasing the number of images in training dataset is essential. In addition, increasing the training dataset benefits the model training process by increasing the variance and reducing the risk of overfitting. The way to increase the number of images depends on doing some operations on the images such as rotating, flipping, shearing etc. In our project, we used every image in the training dataset and applied five basic operations which are: 20° rotation, -20° rotation, 40° rotation, and -40° rotation, image flipping, and re-applying all the previous rotations to the flipped image. By doing these operations we have managed to get from one original image additional nine images that can be used in the model training process. Figure 4.6 shows an example of one image and the resulting nine images after doing the mentioned operations.

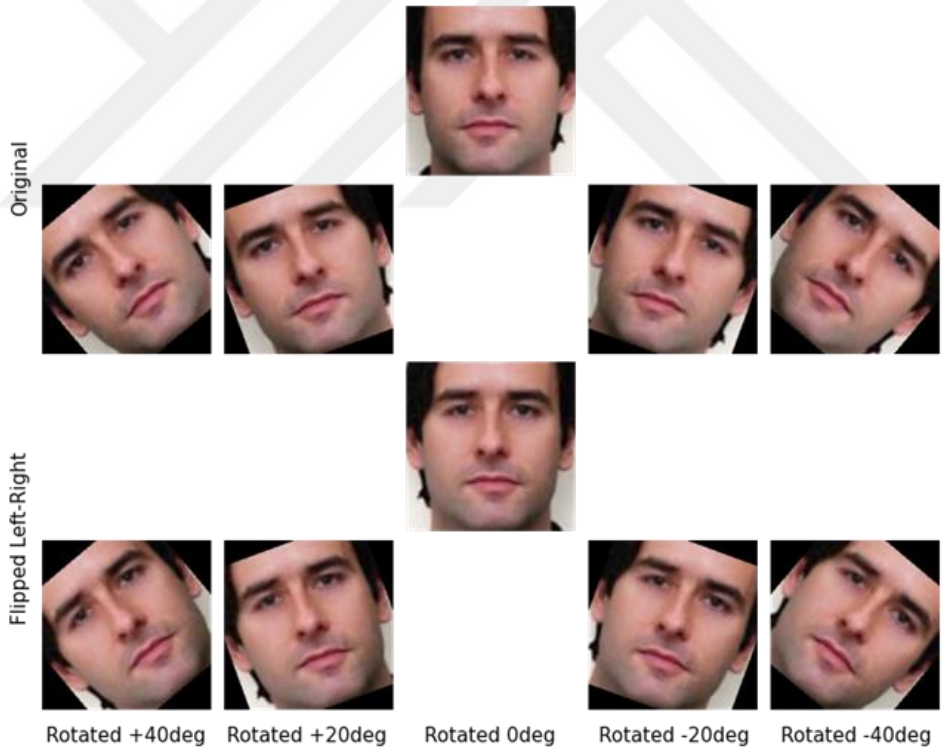


Figure 4.6. *The data augmentation on training dataset*

After performing these operations on the images in the training dataset, we have increased the number of images in the training dataset by ten times, bringing the total to more than 230k images. Table 4.3 shows the number of images and class balance for all datasets that will be used later in our work.

Table 4.3. *The number of images and class balance for each dataset*

Age group	Class (label)	Class balance (%)	Training dataset + augmentation	Training dataset	Validation dataset	Testing dataset	Combined dataset
0-2	0	9.53	22340	2234	718	240	3192
3-7	1	5.94	14020	1390	452	150	1992
8-14	2	5.8	13360	1361	430	143	1934
15-20	3	6.05	14280	1415	458	153	2026
21-27	4	19.98	46840	4684	1505	502	6691
28-45	5	27.94	65490	6549	2105	702	9356
46-65	6	16.71	39190	3919	1259	420	5598
66+	7	8.05	18880	1888	607	202	2697

4.4. Image Preparation

In this stage, and after we have done all the dataset preparation, we have to do some preparations on images before we take them as input to our model. We have at this stage Four basic operations, which are encoding the image labels with one-hot encoding, image resizing, normalizing the images, and grouping the images into mini-batches of images so we can handle it with our limited resources. All these operations will be discussed next.

4.4.1. One-hot encoding

Most people utilize one-hot encoding to deal with categorical feature numeralization since it is straightforward and effective. To represent each category or class attribute's value as a binary vector, it only has to have a single element with a value of 1, the rest being zeros. The element with the value 1 indicates the existence of a category property with a matching potential value. The categorical feature color has three possible values, say blue, red and green. If one-hot coding is used, the blue value may be encoded as (1, 0, 0), the red value as (0, 1, 0), and the green value could be encoded as (0, 0, 1).

In our work, we have 8 classes ranges from 0 to 7 so if we want to represent our classes using one-hot encoding approach, we need a vector with dimension of 1x8 and we put the value 1 in the position corresponding to the value of the class, and the rest of the vector elements will be zeros. For example, to represent Class 5, the corresponding

vector would be as follows: [0 0 0 0 0 1 0 0]. All the images in our datasets must be encoded with one-hot encoding approach before getting into our model.

4.4.2. Image resizing

In this work, four models were built, based on four image resolutions. Each model works on a specific resolution in order to obtain the best result based on the change in the resolution of the image. These resolutions are 200×200 pixels, 100×100 pixels, 50×50 pixels, and 25×25 pixels. The results of each model will be discussed in chapter 5.

4.4.3. Image normalization

Data points from multiple dimensions are compared using machine learning algorithms to uncover patterns in the dataset. When trying to use machine learning, it might be difficult since there are dimensions with wildly varying sizes. Images are stored as three RGB channels in our datasets, thus they must be normalized before being fed into our model.

Min–max normalization is utilized to reduce the dimension scales in this study. A linear transformation is used to bring the original data into the range [0, 1]. The min-max algorithm uses the Equation 4.1 to scale the data:

$$\widetilde{x_{ij}} = \frac{x_{ij} - min(x_i)}{max(x_i) - min\,(x_i)} \tag{4.1}$$

where $min\ (x_i)$ and $max(x_i)$ are the minimum and maximum values of the i^{th} numeric feature x_i, respectively, and $\widetilde{x_{ij}}$ denotes that the normalized feature value is between [0,1].

4.4.4. Mini-batch

So far, we have large datasets for example the training dataset consists of more than 230,000 images, and this number of images cannot be fit into any modern memory. So instead of feeding all the images of the dataset at once, we divide them according to our resource's memory capacity into group of images or batches and each batch of images will be processed and fed into our model. This process collecting number of images and grouping it is called a mini-batch and the mini-batch size is the number of images in one group. There are many benefits to using the mini-batch approach which are: the computational efficiency to fit into limited resources, helping model optimizer to converge in a stable way towards the global minimum, and faster learning since the model

has to update its weights after each mini-batch. The disadvantage of this approach is that the size of the mini-batch is considered as a hyperparameter, which increases the number of hyperparameters of the model, which is already too many. In our work and after a lot of experiments, we found that choosing the mini-batch size as 50 of images is the best. After we use mini-batch approach the number of resulting subsets from the original dataset is calculated using the Equation 4.2:

$$Number\ of\ subsets = \frac{N}{minibatch\ size} \tag{4.2}$$

where *N* is the total number of images in dataset.

4.5. Motivation and Challenges behind the Proposed Model

In Chapter 3, we reviewed some of the most successful CNN models, as well as many studies related to age groups classification task. Most CNN structures contain a very high number of parameters, at least 2.5M, therefore optimizing the weights of such a model requires a significant amount of time during the model training process, which may take a few days to a week or more. The greater the number of parameters employed, the longer the training time required to discover their optimal values. In order to create a model with sufficient precision, a high level of computing competence is required. Table 4.4, shows the number of parameters required by the most common and successful CNN architectures.

Table 4.4. *Number of parameters needed for the most common and successful CNN architectures (Keras Models, n.d.)*

Model	Number of parameters
Xception	22.9M
VGG16	138.4M
VGG19	143.7M
ResNet50V2	25.6M
ResNet101V2	44.7M
ResNet152V2	60.4M
InceptionV3	23.9M
MoblieNetV2	3.5M

The computational resources and computing power at our disposal make it challenging to develop a model capable of keeping up with the work seen in the age-

classification literature using unconditional face images. Using one of the standard structures is also challenging for the same reason. In light of the above, our motive was to design our CNN model and train it from scratch to have the ability to achieve the following requirements:

- To be able to capture face features properly for all age groups.
- To be able to provide a result comparable with the literature for this task.
- The fact that our computing power is limited motivates us to minimize the number of parameters as much as possible in an attempt to develop an inexpensive model.

In chapter 5 we showed that the model we developed meets all of these requirements, as it was able to capture the facial features well for each age group, achieving a better result than all the works in the literature for this task, and outperforming them all with a much smaller number of parameters than what was previously reviewed of the available architectures. The number of parameters we utilized enabled us to reduce the requirement for computational power, which is appropriate with our current computational resources. Next, we will discuss the proposed deep CNN architecture in details.

4.6. Architecture Of Our Deep CNN Model

After all preparations have done, we describe in detail the design of our novel deep CNN structure and summarize the description in Table 4.5. Our model architecture has two main stages: extracting facial features which we call it feature extraction and predict input age group which is a classification operation.

The feature extraction stage contains the convolutional layer, activation function layer (ReLU), batch normalization, max-pooling layer, and at the end of this stage there is a global max-pooling layer. This stage has seven convolutional layers with their corresponding parameters, including the number of each filter, the kernel size of each filter, the stride, etc. All convolutional layer has kernel size of 3×3, without any strides and with padding to make the outputs of the convolutional layers in the same resolution of its input and the weights of the kernel are initialized with a kernel initializer called "Glorot_Uniform". Since we have 7 convolutional layers, the number of filters for each convolutional layer is ordered as follows: 32, 32, 64, 64, 128, 128, and 256 filters for last convolutional layer. All the convolutional layers are followed by Batch normalization

layers in order to unify the dimensions of the next convolutional layer input and after every two layers of both convolutional and batch normalization layers a Max-Pool layer has been implemented to reduce the input image resolution and consequently reduce the feature map. At the end of this stage a global max-pooling layer is added for the preparation to next stage.

The next stage is the classification stage, where we have two FC layers to predict the age group. The first FC layer has 128 neurons with a ReLU activation layer, and followed by batch normalization layer. Finally, the last fully-connected layer which is a softmax output layer, it outputs the previous 128 features which are densely mapped to 8 neurons for the age group classification task. The output layer is adopted by a softmax with cross-entropy loss function to obtain the probability of each class.

Cross-Entropy: The performance of classification models is evaluated using cross-entropy loss, which produces an output value between 0 and 1. Cross-entropy loss diminishes as the predicted probability approaches the right label; the lower the cross-entropy result, the more generalizable the classification model.

In binary classification, when N equals to 2, the cross entropy is consequently defined by the Equation 4.3:

$$-(z\log(p)+(1-z)\log(1-p)) \tag{4.3}$$

while in multi-class classification case where N >2, we calculate a separate loss for each label of observation and then sum the outcome using the Equation 4.4:

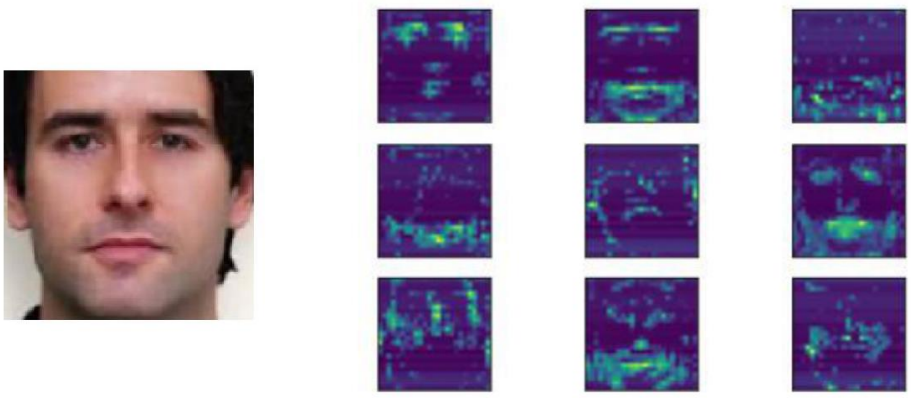
$$-\sum_{c=1}^{N} z_{o,c}\log\left(p_{o,c}\right) \tag{4.4}$$

where N is the number of classes and z represents the binary indicator (0 or 1) If observation o's actual class label is c, then log is the natural log and p is the predicted probability observation o of class label c.

Example of an input image and the corresponding output feature map of last max-pool layer, which is max-pool 4, is shown in the Figure 4.7. From Figure 4.7 we can see that the output of last layer of feature extraction stage has done its task and it detects the facial details very well. These features will be used later in the classification stage to estimate the age group for an input image.

Table 4.5. *Proposed deep CNN architecture*

Layer (type)	Output size	Number of parameters
Input Image	200×200×1	-
Conv 1	200×200×32	320
BatchNormalization 1	200×200×32	800
Conv 2	200×200×32	9248
BatchNormalization 2	200×200×32	800
Max-pool 1	100×100×32	-
Conv 3	100×100×64	18496
BatchNormalization 3	100×100×64	400
Conv 4	100×100×64	36928
BatchNormalization 4	100×100×64	400
Max-pool 2	50×50×64	-
Conv 5	50×50×128	73856
BatchNormalization 5	50×50×128	200
Conv 6	50×50×128	147584
BatchNormalization 6	50×50×128	200
Max-pool 3	25×25×128	-
Conv 7	25×25×256	295168
BatchNormalization 7	25×25×256	100
Max-pool 4	12×12×256	-
Global Max-pool 1	256	-
FC 1	128	32896
BatchNormalization 8	128	512
FC 2	8	1032
Total params: 618,940, Trainable params: 617,234, Non-trainable params: 1,706		



(a) *(b)*

Figure 4.7. *Output of feature extraction stage (a) input image example (b) output feature map*

It is worth mentioning that the input layer, which is the first layer, whose parameters are changed according to the resolution of the images that is provided into the model, as we mentioned that there are four image resolutions that were used in this work which are 200×200 pixels, 100×100 pixels, 50×50 pixels, and 25×25 pixels. Therefore, we have four models with the following inputs respectively 200×200×1, 100×100×1, 50×50×1, and 25×25×1. The four models in our experiments were named after the image resolution that they deal with. So, we have model-200, model-100, model-50, and model-25. For the model-200 and model-100 the structure of the models remains the same as explained but for model-50 and model-25 some changes are done. In case of model-50 and mode-25 the input image is relatively small and after the images are passed we have noticed from the output of last max-pooling layer that nothing remains from the image so we have to reduce the number of layers in these two models in order to have some features that can the models work with. So, in model-50 the max-pooling layer is excluded from the architecture and the layers sequence is as the following:

Input - Conv1 - BatchNormalization1 - Conv2 - BatchNormalization2 - Max-pool1 - Conv3 - BatchNormalization3 - Conv4 - BatchNormalization4 - Max-pool 2 - Conv5 - BatchNormalization5 - Conv6 - BatchNormalization6 - Max-pool3 - Conv7 - BatchNormalization7 - Global Max-pool1 - FC1 - BatchNormalization8 - FC2.

In case of model-25 more than one layer are excluded from the architecture because of the very small size of input images. The layers sequence of model-25 is modified as the following:

Input - Conv1 - BatchNormalization1 - Conv2 - BatchNormalization2 - Max-pool1 - Conv3 - BatchNormalization3 - Conv4 - BatchNormalization4 - Max-pool 2 - Conv3 - BatchNormalization3 - Global Max-pool1 - FC1 - BatchNormalization8 - FC2.

The results of all four models' performance will be discussed in detail later in the next chapter.

4.7. Summary

So far, we have presented the general framework of this chapter and explained all the preparation processes for datasets and images, and we have also explained the structure of the models that we proposed. In the next chapter, we will evaluate these models, present and discuss the results to complete the picture of the work completely.

5. EVALUATIONS AND RESULTS

We arrange this chapter as follows: First, we introduce the development environment, which includes the programming language and APIs, programming libraries, frameworks used to conduct this work and experiments, in addition to specifying the computer resources that were used to implement, experiment and examine the work. Second, we present the evaluation metrics used to measure the performance of the model. Third, we present the models' training experiments and the used parameters. Fourth, we present and discuss the results of the four models testing on the test dataset. Finally, we choose the best model and compare it with the last state-of-art model which achieved the best results in the literature.

5.1. Development Environment

To develop this work, we used the Python programming language, which is one of the most important and most common programming languages in the field of artificial intelligence applications. Also, with this programming language, we have utilized many important programming libraries and APIs to complete this work. These libraries are: TensorFlow, Keras, NumPy, Scikit-Learn, Pandas, Matplotlib, Seaborn and OpenCV. We present a brief overview of these libraries as follows:

TensorFlow is a free and open-source Python library that is used in Artificial intelligence and machine learning application. TensorFlow provides access to a variety of tools for use by academics and programmers. These tools assist with making the building of DL and ML models approachable to both novices and experts. TensorFlow is able to operate on several computational platforms, like CPU and GPU, because of its adaptable design and foundation. However, it functions most optimally when run on a tensor processing unit (TPU). We have used the GPU version of Tensor Flow, it’s called "TensorFlow-GPU", to implement our model which gives us 10x faster performance than the CPU one.

Keras is an open-source Python library and it is also considered as TensorFlow API built specifically for constructing and testing neural networks in DL and ML models. Using a little amount of code, it is feasible to begin training neural networks with Keras, making the development of ML models quite straightforward. The modular, flexible, extendable, and user-friendly Keras library enables developers to do rapid and effective

prototyping, research, data modeling, and visualization. Similar to TensorFlow, Keras can operate on several computational platforms like CPUs and GPUs.

NumPy Python's popular numerical library, can perform a wide range of mathematical operations on arrays and matrices, and it's available for free. NumPy helps in improve the performance of ML models thanks to its ability to do fast vector calculations and is therefore suited for machine learning and AI projects.

Pandas is a Python package for data science and data analysis, which is based on NumPy, it is used to clean and organize data before machine learning is applied.

Matplotlib is a package that may be used to generate many types of plots and graphs from data. It's capable of processing NumPy data structures and advanced Pandas data models.

OpenCV is a library for real-time computer vision applications that can analyze a range of visual inputs from both images and videos. The library's real-time data processing applications are intended to take full use of the multicore processing capabilities of the processor.

With the aid of the aforementioned libraries and the computer resources listed in Table 5.1, we were able to develop an age group classification model. The graphics card and the size of its RAM memory play a crucial role in this study, since they utilized to do all computations in parallel, resulting in fast model training and development.

Table 5.1. *Computer resources utilized for this work*

Component name	Description
CPU	Intel(R) Core (TM) i7-6700HQ
GPU	Nvidia RTX 2060 with 6GB graphic memory
RAM	12GB
Hard disk	Samsung Evo SSD 500GB

5.2. Evaluation Metrics

A novel DCNN model has been presented which classify unconstrained facial images according to their actual age group. Using the combined dataset, the performance of this model to classify an individual into the correct age group was evaluated via a different number of empirical experiments. The classifier performance is measured by five measures two of them are standard measures in the literature, The confusion matrix

and accuracy are the standard ones, and precision, recall, and F1-score are additional evaluation measures to further evaluate the performance of the proposed model.

Confusion Matrix: This metric evaluates the performance of a multi-class age group classification model on a particular testing dataset. Table of predicted and actual classes are shown in a confusion matrix metric to illustrate the classification model's performance in various combinations. Since in our model we have 8 classes which are (0–2, 3-7, 8–14, 15–20, 21-27, 28-45, 46–65, 66+), the confusion matrix shows the results of age groups for the eight classes. The metric results of the previously mentioned model are generated on validation dataset, testing dataset, and on both combined for age group classification.

Accuracy: Accuracy is a commonly utilized performance measure for classifiers and classification tasks. From an age group classification perspective, it measures how accurately a classifier can predict an age group for an individual. In other words, it provides the ratio of correctly classified facial images to the total number of the dataset images. It is calculated using the Equation 5.1:

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \times 100 \quad (5.1)$$

where TP means true positive which is the number of true positive predictions, TN means true negative which is the number of true negative predictions, FP is the number of false positive predictions, and FN is the number of false negative predictions.

Precision: While accuracy is a solid measure to tell how the model is performing, but it is not really the only metric used to draw the overall conclusions. A model can provide a high accuracy score with a class-balanced input dataset, but if the input dataset is unbalanced according to the classes, it may fail to correctly perceive the data. Precision is the percentage of predicted images that is actually accurately classified. A greater precision score indicates that the model is recognizing the age group properly. It gives more importance to increasing the number of correct predictions for a certain class that we are predicting It is calculated using the Equation 5.2:

$$Precision = \frac{TP}{TP+FP} \quad (5.2)$$

Recall: The recall metric is related to sensitivity and the level of confidence that all positive occurrences or actual age groups were predicted positively. It gives more

importance to increasing the number of examples in a certain class that we accurately predicted. It is calculated using the Equation 5.3:

$$Recall = \frac{TP}{TP+FN} \tag{5.3}$$

F1-Score: This metric combines precision and recall to measure the overall accuracy and performance of a model. A high F1-score implies that a model effectively predicted age group while minimizing false positives and false negatives. It is calculated using the Equation 5.4:

$$F1 - score = \frac{precision \times recall}{precision+recall} \times 2 \tag{5.4}$$

5.3. DCNN Model Training

At this stage, we will present the training details of our model which will predict the age group of unconditional face images on the combined dataset. The age group classifier is responsible for predicting the age group of eight different classes which are 0-2,3-7,8-14,15-20,21-27,28-45,46-65,66+. We have trained the network on a large combined dataset containing unconstrained real-life faces with age labels. The size of our training dataset was so crucial to build a good classifier able to learn the bias from a large dataset of image samples and generalize well on the validation and test datasets. As we mentioned in the previous chapter, the combined dataset was divided into three datasets: 70% for training, 22.5% for validation, and 7.5% for testing and then the augmentation process is applied only to the training dataset to increase the training dataset size which reduces the risk of overfitting and enable the model to generalize well on validation and testing datasets. The resolution of the input images was resized to the following resolutions 200×200 pixels, 100×100 pixels, 50×50 pixels, and 25×25 pixels. Then four models were experimented according to the resolution of the provided images. Then the images are converted to the grayscale colors and the labels of these images were also converted to one-hot encoding before passing the image and its label into the model. This conversion helped the model to detect the edges very well and avoid the confusion of processing the three-channel of RGB colors. We have specified the batch size as 50 for training, validation, and testing datasets and the network has trained on this batch size using the training dataset. We have also utilized Adam optimizer, which is considered one of the adaptive optimizers family, in our training process to optimize the loss function with a learning rate of 0.001 and mini-batches of size 50. The training starts with initial

weights initialized with Glorot_Uniform initializer and whenever there are improvements in the accuracy of the validation dataset, a checkpoint had taken, and the model with its weights had saved in a file of h5 type. The network has trained for 100 epochs.

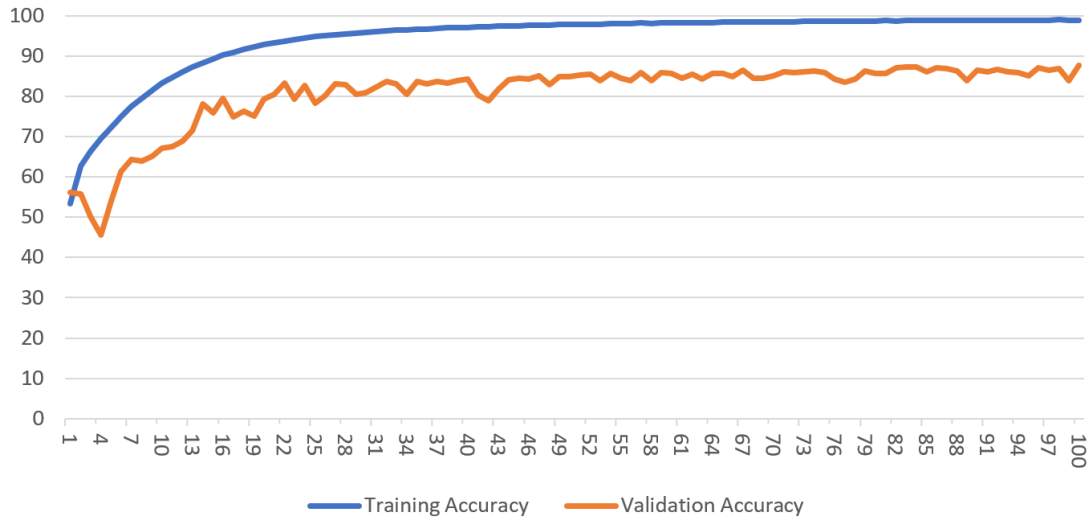
The Figure 5.1 shows the model's accuracy and loss with image resolution 200×200 pixels for both the training and validation datasets. Although, we have noticed that the model slightly overfits the training dataset, which may still be acceptable as long as it generalizes well on validation dataset and gives good accuracy result on a large testing dataset.

The four models in our experiments were named after the image resolution that they deal with. So, we have model-200, model-100, model-50, and model-25. Figure 5.2, Figure 5.3, and Figure 5.4 show the model's accuracy and loss for the following models: model-100, model-50, and model-25 respectively. We found the same observations in the performance of model-100, model-50, and model-25 that were found in model-200 except the training time was reduced and the accuracy was becoming decreasing as a result of lowering the image resolution which is a reasonable result.

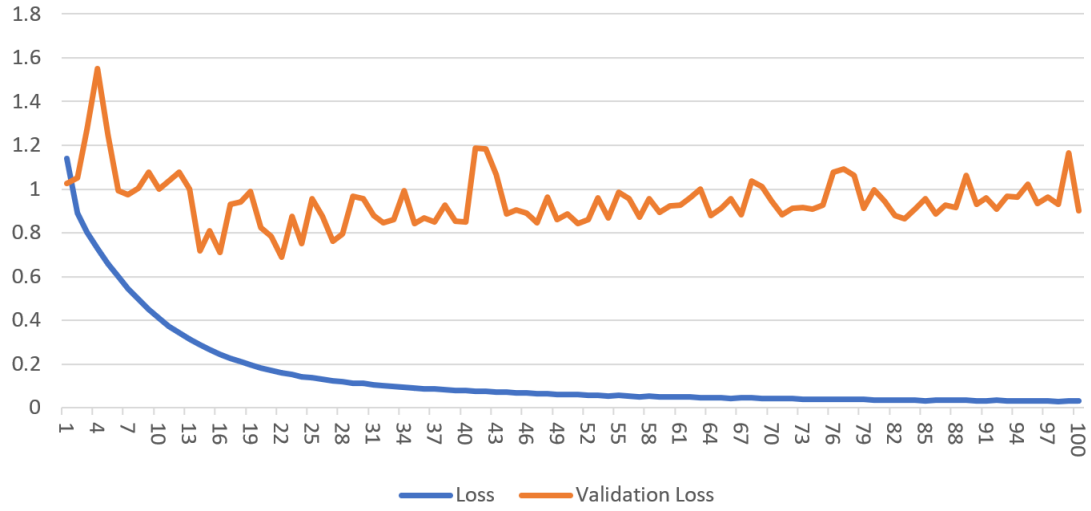
Table 5.2 shows the training time and the average for each epoch among 100 epochs for the following models: model-200, model-100, model-50, and model-25 where it takes 100000 seconds, 31000 seconds, 20400 seconds, and 18600 seconds as the total time of training and 1000 seconds, 310 seconds, 204 seconds, and 186 seconds as the average time for each epoch respectively.

Table 5.2. *Total Training time and average time for each epoch in all models*

Model	Number of epochs	Average Time (sec)	Total Training time (sec)
Model-200	100	1000	100000
Model-100	100	310	31000
Model-50	100	204	20400
Model-25	100	186	18600

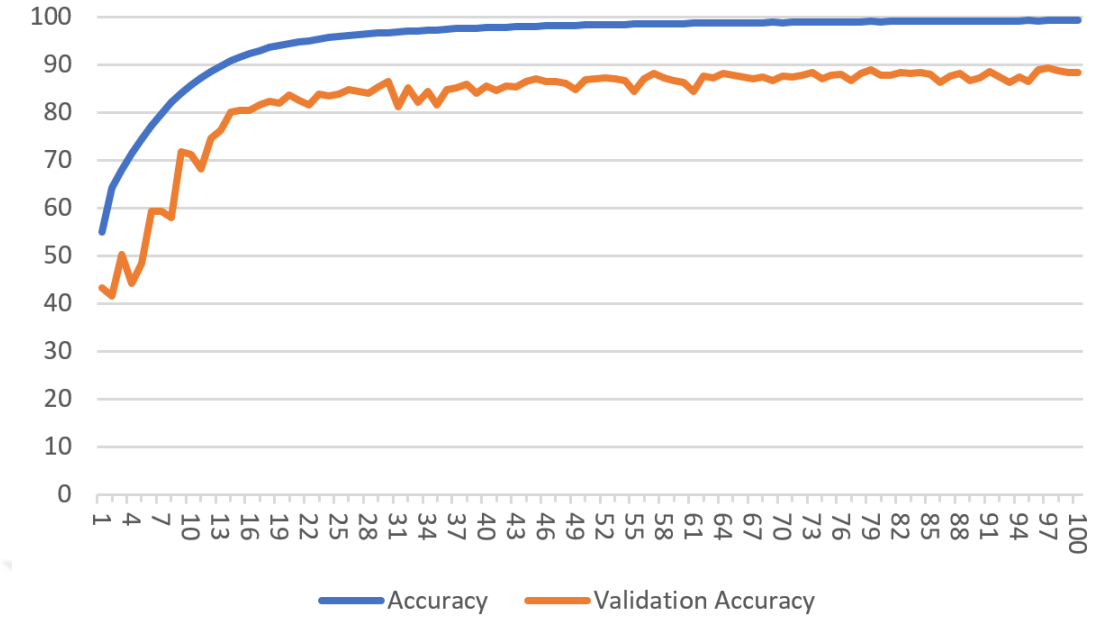


(a)

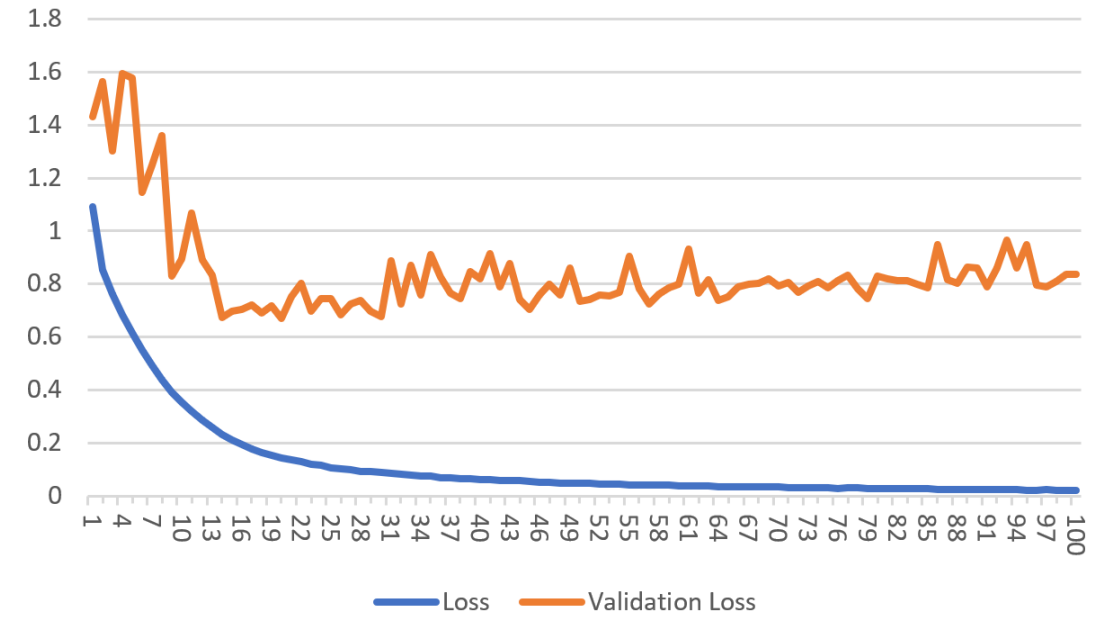


(b)

Figure 5.1. *Model-200's accuracy and loss for both the training and validation datasets in model training (a) accuracy and validation accuracy (b) loss and validation loss*

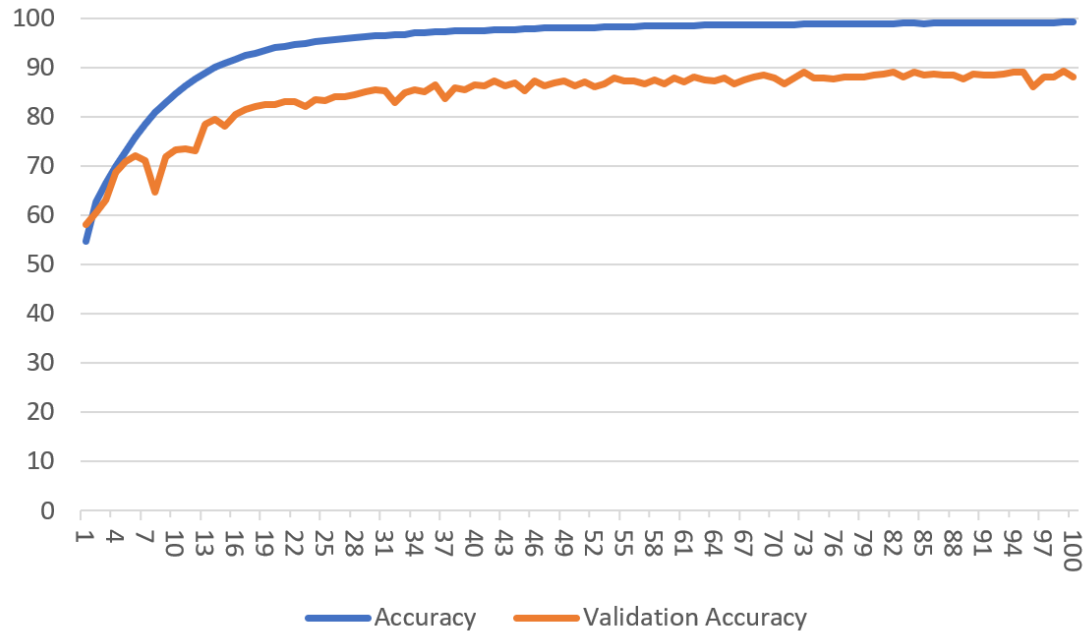


(a)

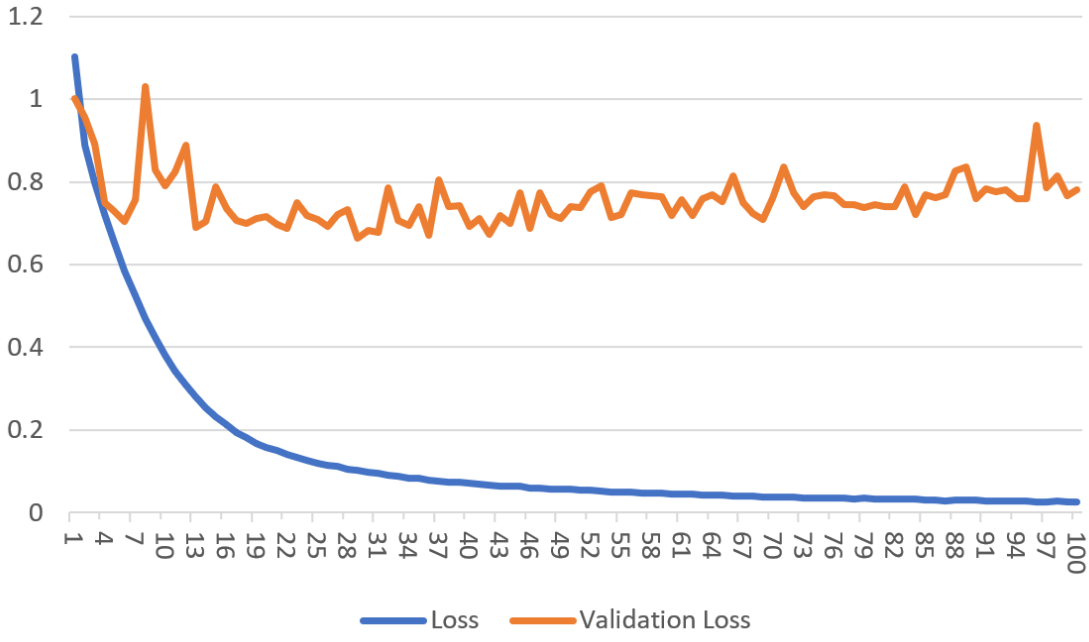


(b)

Figure 5.2. *Model-100's accuracy and loss for both the training and validation datasets in model training (a) accuracy and validation accuracy (b) loss and validation loss*

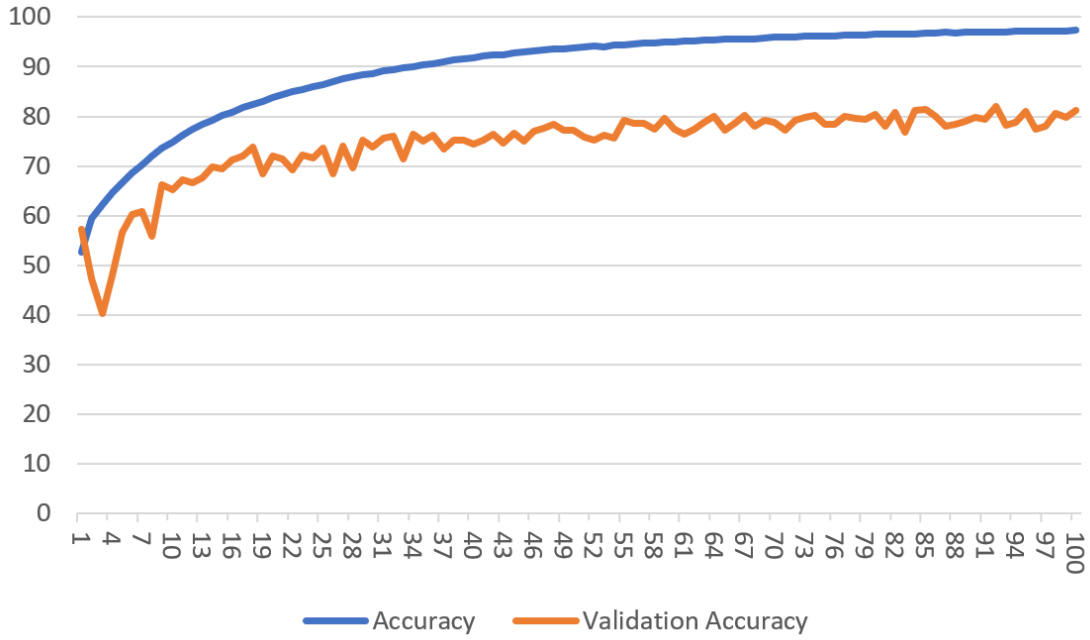


(a)

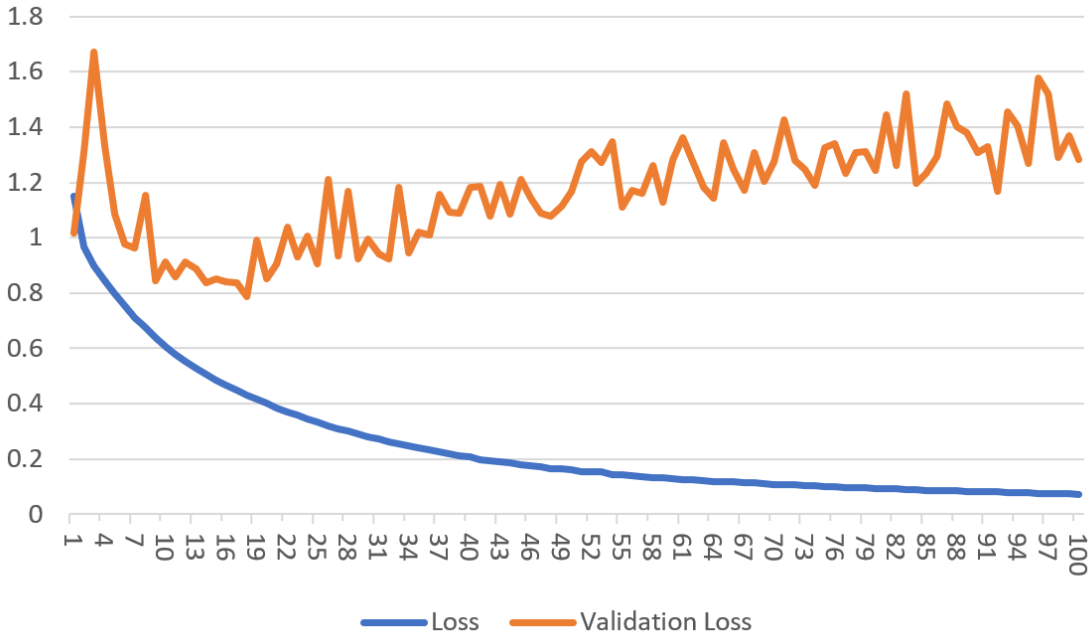


(b)

Figure 5.3. *Model-50's accuracy and loss for both the training and validation datasets in model training (a) accuracy and validation accuracy (b) loss and validation loss*



(a)



(b)

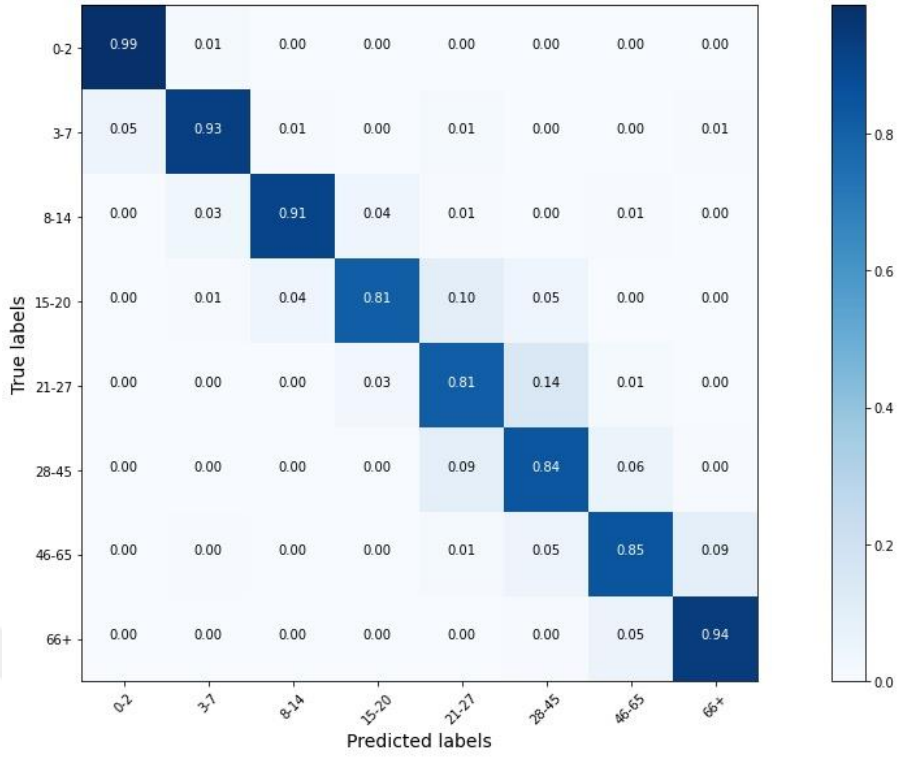
Figure 5.4. *Model-25's accuracy and loss for both the training and validation datasets in model training (a) accuracy and validation accuracy (b) loss and validation loss*

5.4. Results and Discussion

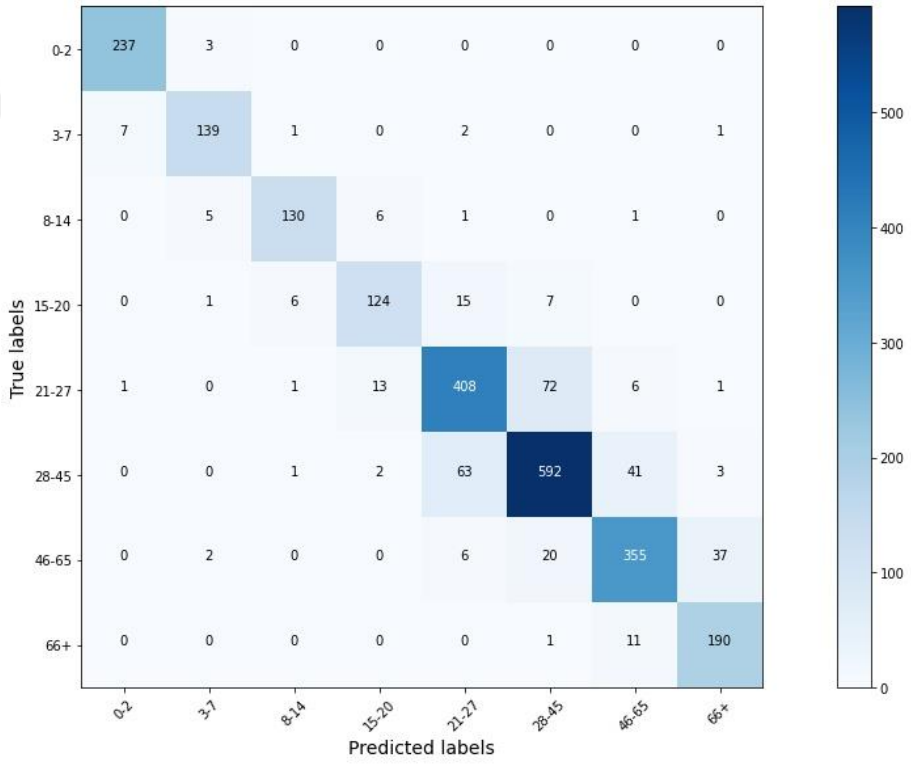
In this section, and as we mentioned in chapter 4, we have used our combined dataset to evaluate the four model which are model-200, model-100, model-50, and model-25. A test dataset has utilized to satisfy this purpose. The test dataset has 2512 facial images with its corresponding label encoded in one-hot encoding. All classes in test dataset are imbalanced, and this has an effect on the model, which may be manifested by the model's bias towards certain classes rather than others. Next, we will present the results of evaluation for the four models on test dataset using confusion matrix, accuracy, precision, recall, and f1-score metrics so we have a very good idea about the model performance.

5.4.1. Test dataset results for model-200

The model-200 achieved a very good result on the test dataset, which contains 2,512 face images, with an accuracy of 86.58% as an overall accuracy for all classes. Figure 5.5 shows the result of the confusion matrix measure of the performance of our model on the test dataset. We have noticed from Figure 5.5 that the age groups 0-2, 3-7, 8-14, 66+ were classified very well with an accuracy almost 90% and more than that. This is reasonable because the person's face in these age groups has distinctive features that enabled the model to learn them and distinguish these classes among other. Whereas in the age groups 15-20, 21-27, 28-45, 46-65 the appearance of the distinctive features is lower which results in less accuracy than the previous age groups. We have also noticed that when the model misclassifies a particular class, it misclassifies in favor of the adjacent class, and this is also reasonable, since the human age range is a continuous range in terms of value and there are no separator values can separate the age groups clearly.



(a)



(b)

Figure 5.5. *Confusion matrix's result for model-200's performance on the test dataset (a) matrix result in percentage (b) matrix result with the actual number of images*

Figure 5.6 shows the result of precision, recall, f1-score measures for the performance of our model on the test dataset. Although we used several measures and the fact that the dataset classes are unbalanced, we have noticed from Figure 5.6 that the results of the model's performance on all these measures are very good and so close to each other in terms of overall model accuracy and also the accuracy for each class separately. We have also noticed that in the 15-20 group, the accuracy of the precision metric is a little bit higher than the recall metric accuracy, which means that if the model predicts on this class, 85.52% of model predictions is correct, but there are some samples in actual age group of this class that the model could not correctly recognized it. We have also noticed in the 66+ class that the result of recall metric is noticeably increased with accuracy of 94.06%, which means that the model is very capable of recognizing and classifying the actual images of this class.

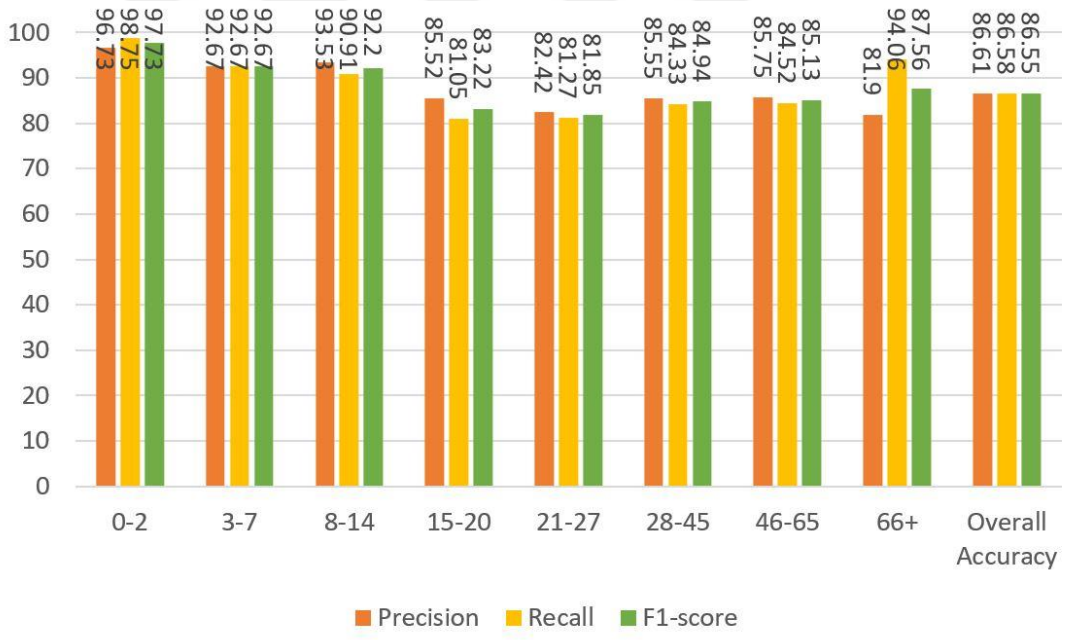


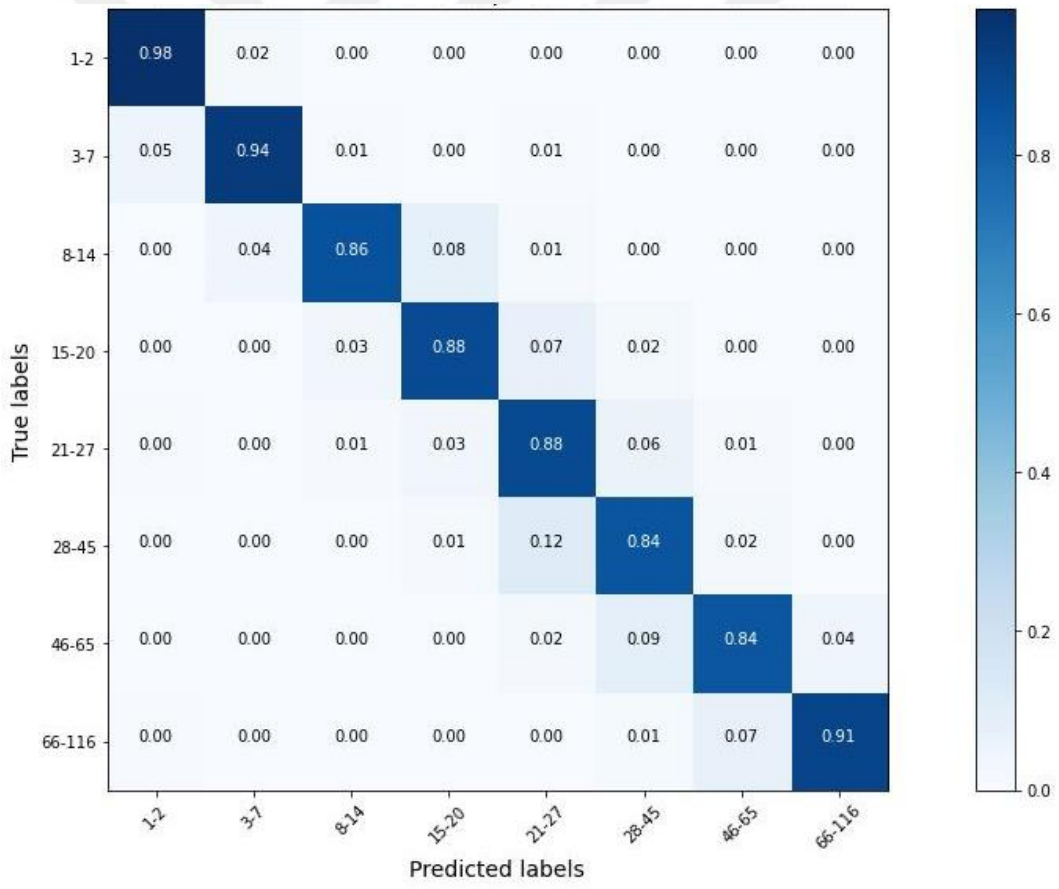
Figure 5.6. *Result of precision, recall, and f1-score metrics for model-200 on test dataset*

The overall accuracy of the model on the test dataset is 86.55% which is the result of f1-score metric. using the result of f1-score metric, the accuracy for 0-2, 3-7, 8-14,15-20, 21-27, 28-45, 46-65, and 66+ classes is respectively 97.73%, 92.67%, 92.2%, 83.22%, 81.85%, 84.94%, 85.13%, and 87.56%.

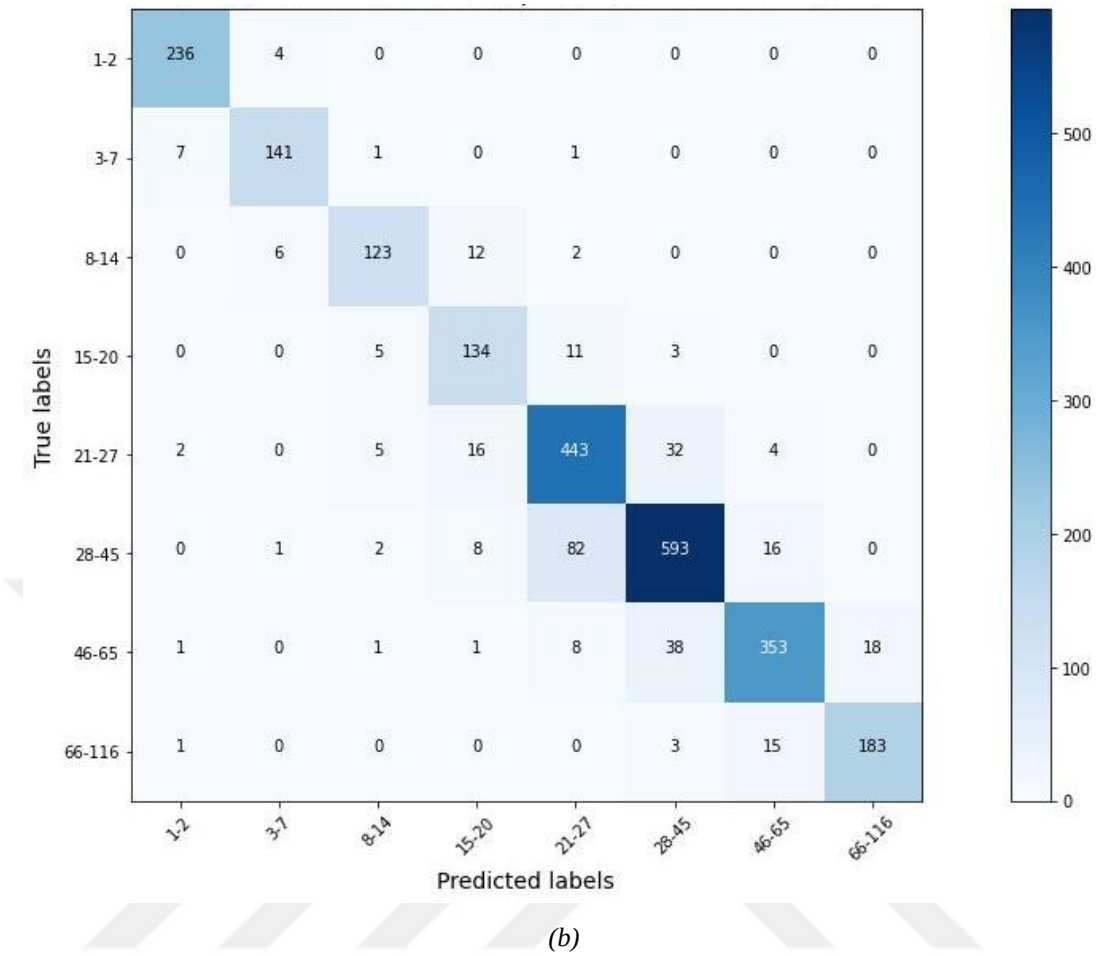
5.4.2. Test dataset results for model-100

The model-100 achieved a very good result even better result than model-200, on the test dataset, with an accuracy of 87.82% as an overall accuracy for all classes. Figure 5.7 shows the result of the confusion matrix measure of the performance of model-100 on the test dataset. Figure 5.8 shows the result of precision, recall, f1-score measures for the performance of our model on the test dataset. From the Figure 5.7 and Figure 5.8, we have also noticed the same observations that were discussed in the results of model-200.

The overall accuracy of the model-100 on the test dataset is 87.84% which is the result of f1-score metric. using the result of f1-score metric, the accuracy for 0-2, 3-7, 8-14,15-20, 21-27, 28-45, 46-65, and 66+ classes is respectively 96.92%, 93.38%, 87.86%, 82.72%, 84.46%, 86.51%, 87.38%, and 90.82%.



(a)



(b)

Figure 5.7. *Confusion matrix's result for model-100's performance on the test dataset (a) matrix result in percentage (b) matrix result with the actual number of images*

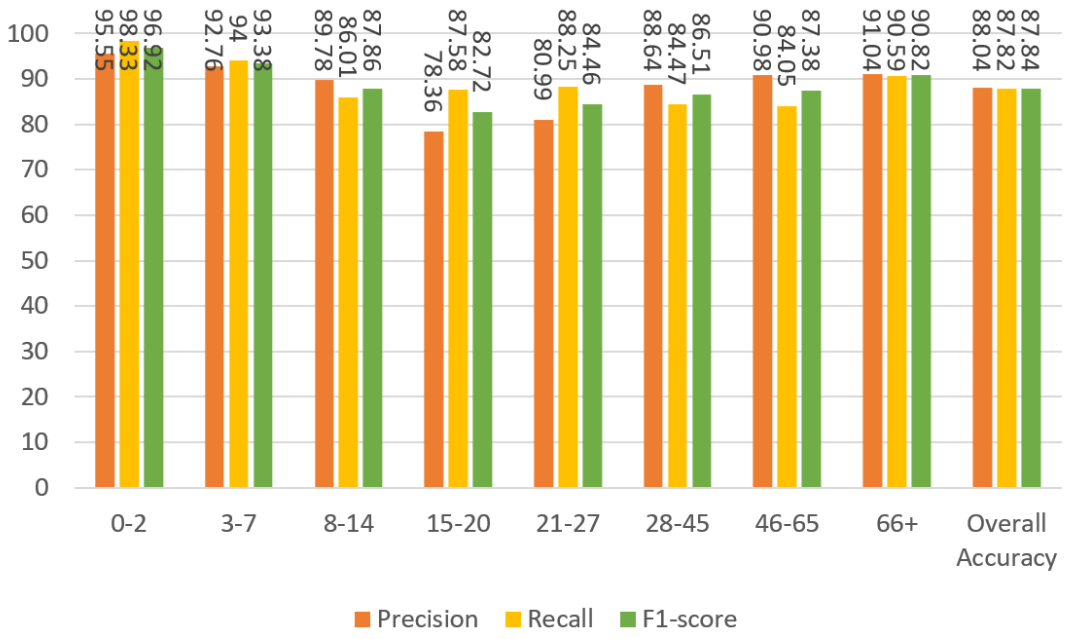
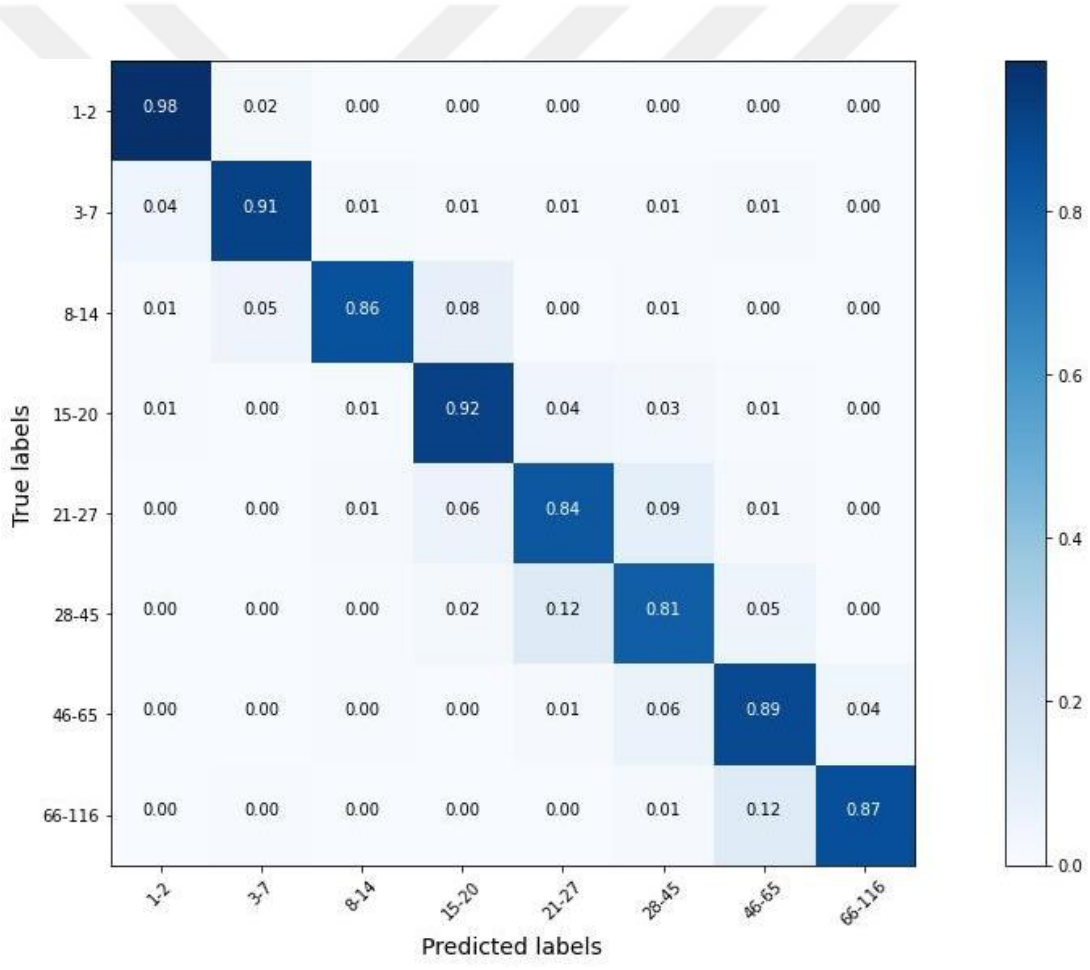


Figure 5.8. *Result of precision, recall, and f1-score metrics for model-100 on test dataset*

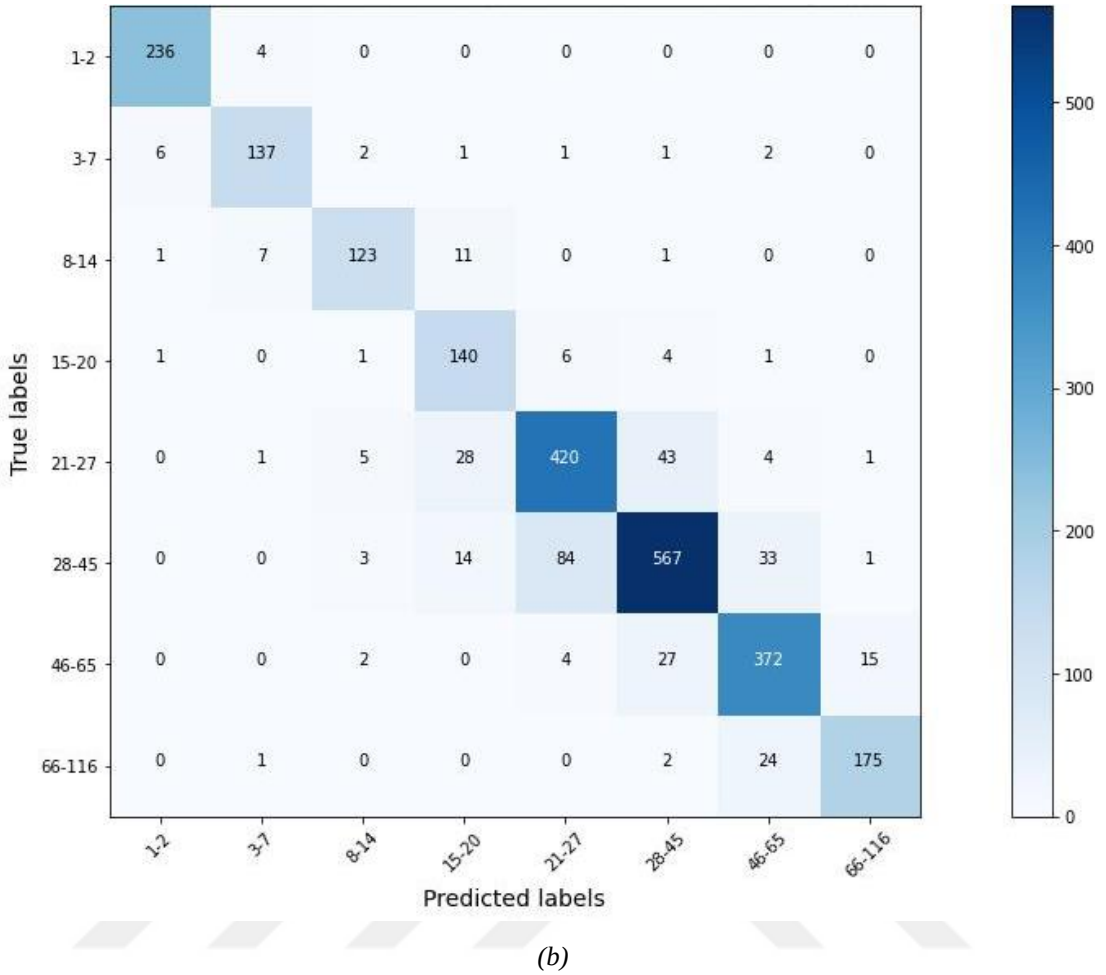
5.4.3. Test dataset results for model-50

The model-50 achieved a good result but worse than the model-200 and model-100, on the test dataset, with an accuracy of 86.4% as an overall accuracy for all classes. Figure 5.9 shows the result of the confusion matrix measure of the performance of model-50 on the test dataset. Figure 5.10 shows the result of precision, recall, f1-score measures for the performance of our model on the test dataset. From the Figure 5.9 and Figure 5.10, we have also noticed the same observations that were discussed in the results of model-200.

The overall accuracy of the model-50 on the test dataset is 87.51% which is the result of f1-score metric. using the result of f1-score metric, the accuracy for 0-2, 3-7, 8-14,15-20, 21-27, 28-45, 46-65, and 66+ classes is respectively 97.52%, 91.33%, 88.17%, 80.69%, 82.6%, 84.19%, 86.92%, and 88.83%.



(a)



(b)

Figure 5.9. *Confusion matrix's result for model-50's performance on the test dataset (a) matrix result in percentage (b) matrix result with the actual number of images*

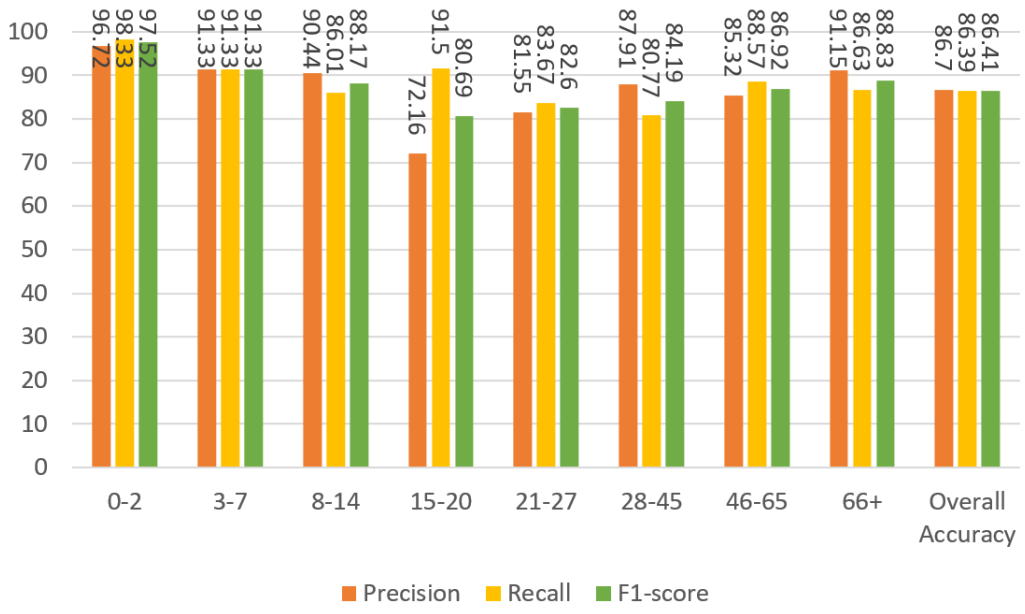
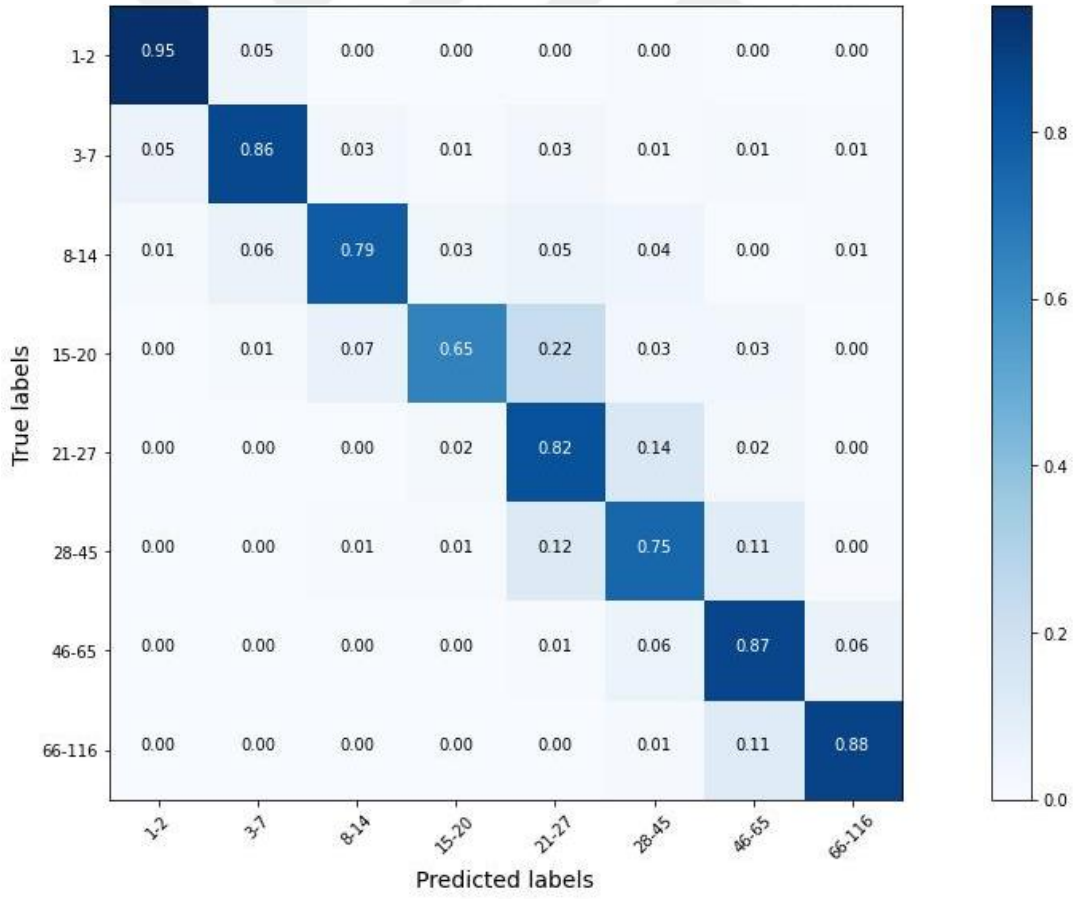


Figure 5.10. *Result of precision, recall, and f1-score metrics for model-50 on test dataset*

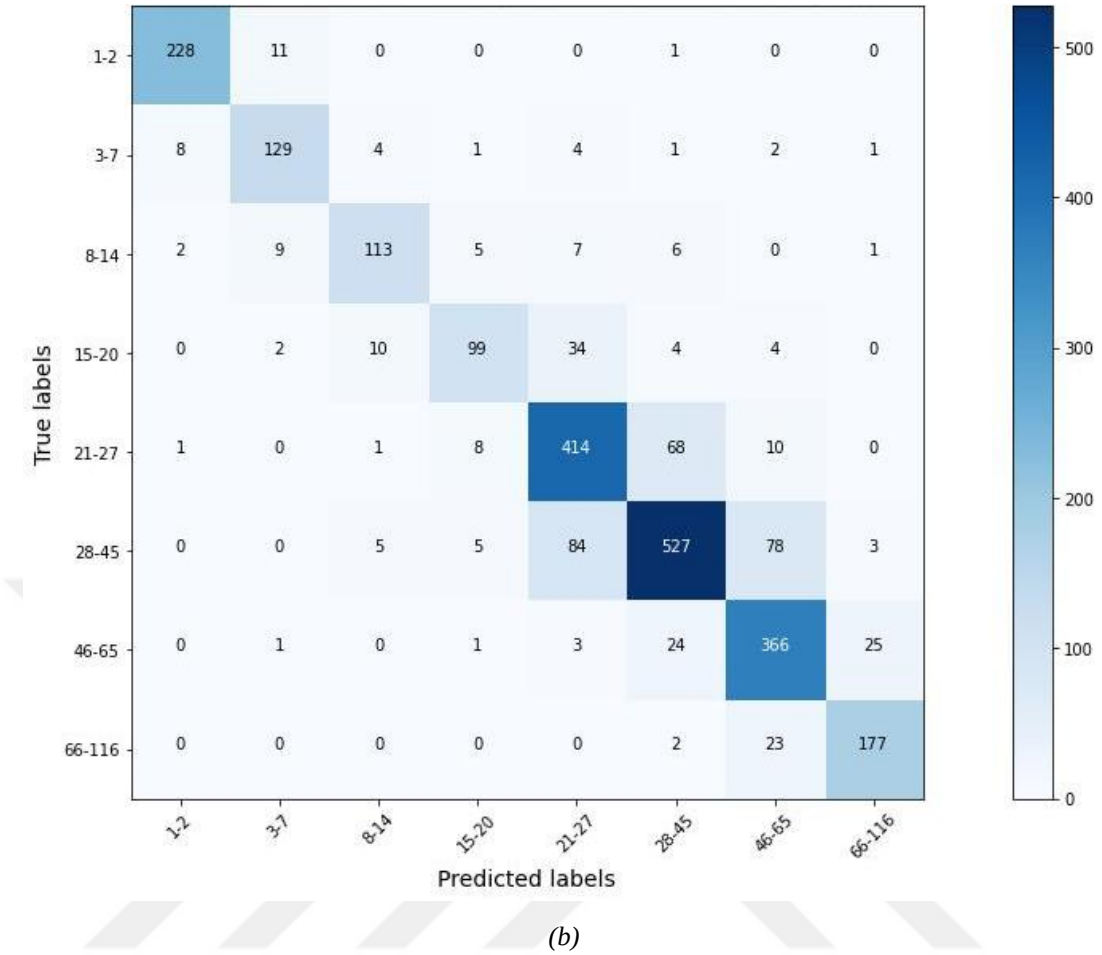
5.4.4. Test dataset results for model-25

The model-25 achieved a good result but worse than the model-200, model-100, and model-50, on the test dataset with an accuracy of 81.7% as an overall accuracy for all classes. Figure 5.11 shows the result of the confusion matrix measure of the performance of model-25 on the test dataset. Figure 5.12 shows the result of precision, recall, f1-score measures for the performance of our model on the test dataset. From the Figure 5.11 and Figure 5.12, we have also noticed the same observations that were discussed in the results of model-200.

The overall accuracy of the model-25 on the test dataset is 86.55% which is the result of f1-score metric. using the result of f1-score metric, the accuracy for 0-2, 3-7, 8-14,15-20, 21-27, 28-45, 46-65, and 66+ classes is respectively 95.2%, 85.43%, 81.88%, 72.79%, 79.01%, 78.95%, 81.06%, and 86.55%.



(a)



(b)

Figure 5.11. *Confusion matrix's result for model-25's performance on the test dataset (a) matrix result in percentage (b) matrix result with the actual number of images*

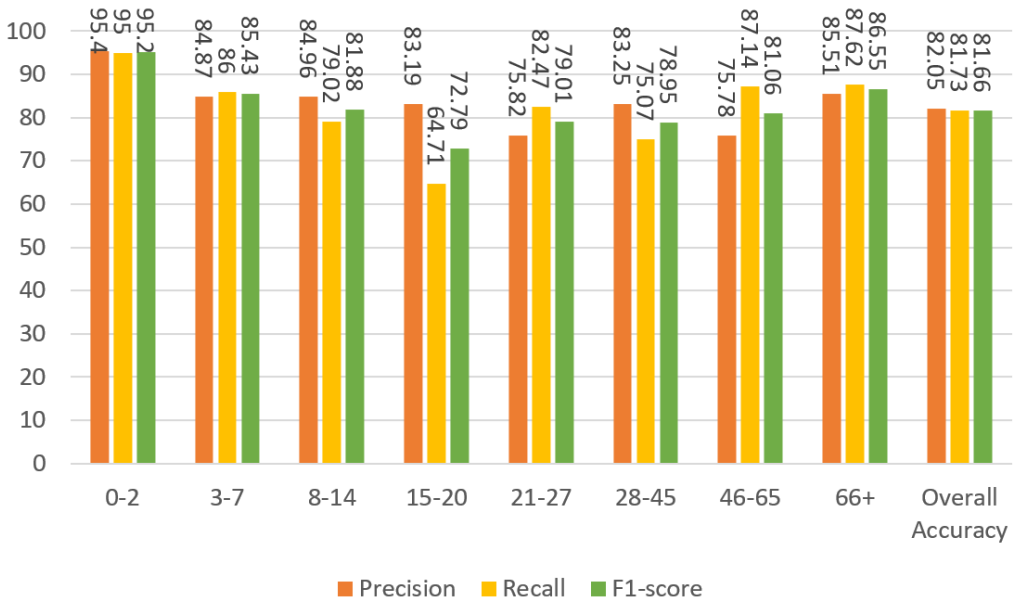


Figure 5.12. *Result of precision, recall, and f1-score metrics for model-25 on test dataset*

5.5. Comparison of Four Models

As we have seen from the results of the previous sections which include the following models: model-200, model-100, model-50, and model-25, the best result was achieved by the model-100 with an accuracy of 87.82% on test dataset, as shown in the Figure 5.13. Although the model-200 has a higher image resolution as an input, but the model-100 achieved a better result, because the original images of the dataset have a resolution less than 200×200, and when they are enlarged to the resolution of 200×200, they give a blurry image which affects the result of the model. The Figure 5.13 shows the accuracy of all the models mentioned with precision, recall, f1-score metrics. Since the model-100 has the best result, we have considered it as our best proposed model and next we have used it to compare our model with best state-of-are model in the literature.

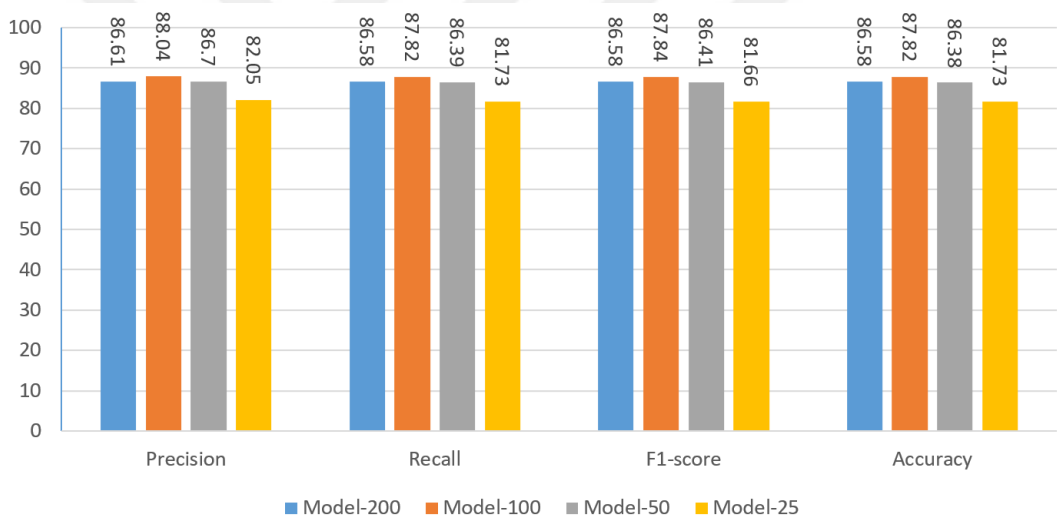


Figure 5.13. *Accuracy, precision, recall, and f1-score metrics result for all models*

5.6. Comparison of Our Best Model with Other Works

The work done by Agbo-Ajala & Viriri (2020) achieved the best state-of-art result found in the literature for classifying facial images into age groups. The authors defined age estimation task as a classification into eight age groups very close to the groups we have chosen. Figure 5.14 shows the confusion matrix for their model on testing dataset of the OIU-Adience dataset. Their testing dataset includes 992 facial images while our testing dataset is much bigger including 2515 images.

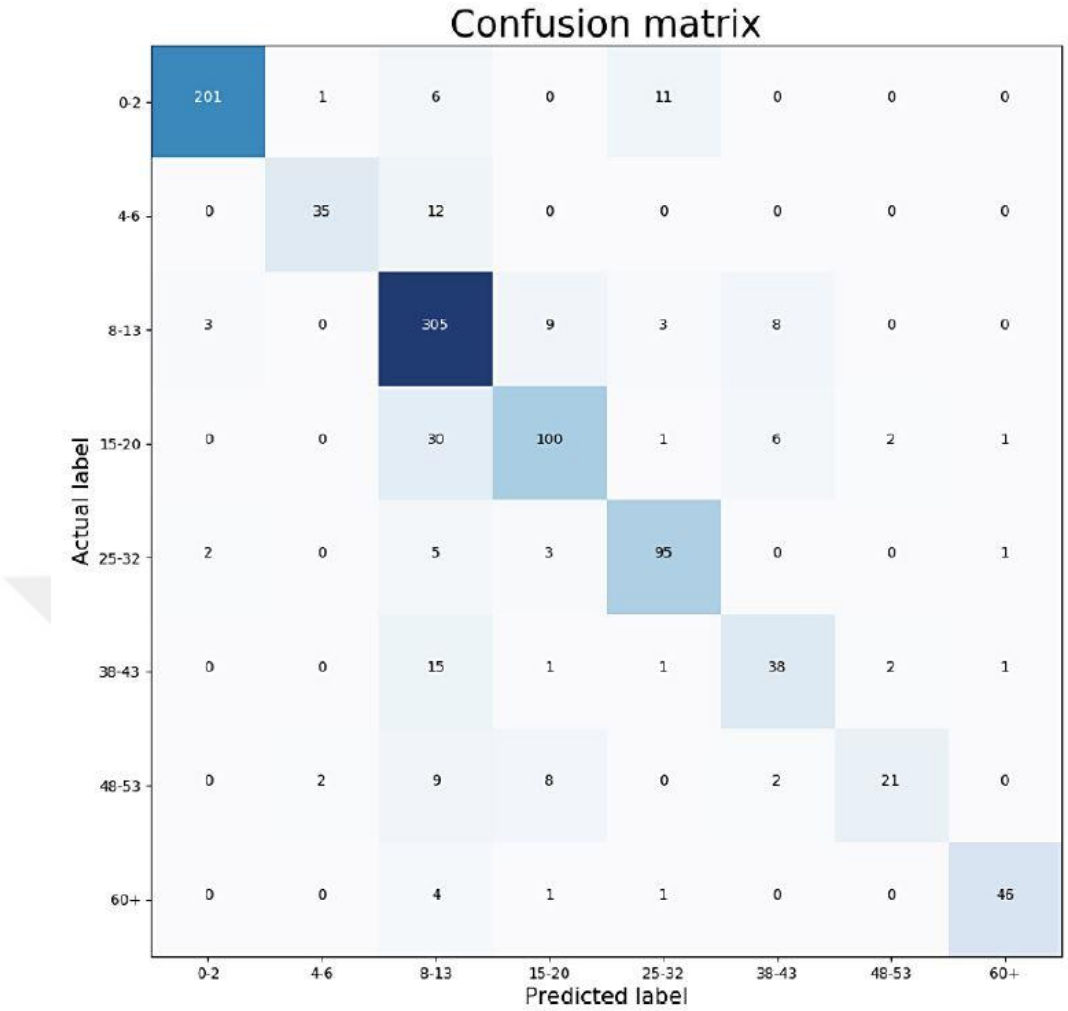


Figure 5.14. *Result of confusion matrix for (Agbo-Ajala & Viriri, 2020) work*

Relying on their confusion matrix we were able to calculate the results of precision, recall, and f1-score measures as shown in Figure 5.15. According to the F1-score metric their work has an overall accuracy 81.58% which is much less from the accuracy of our proposed model.

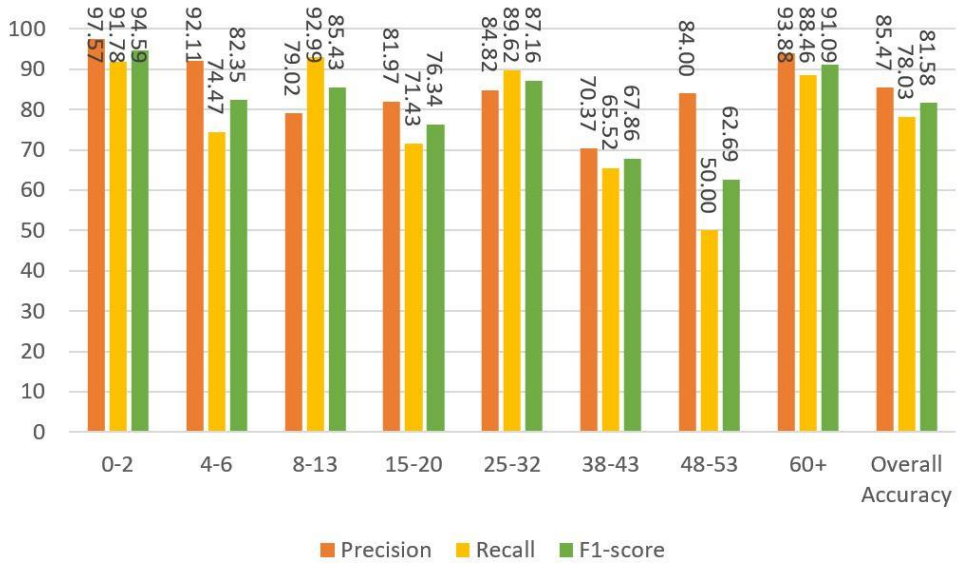


Figure 5.15. *The result of precision, recall, and f1-score metrics for (Agbo-Ajala & Viriri, 2020) work*

The Table 5.3, shows a comparison between our model and their model in terms of the overall accuracy and the accuracy for each class separately using the f1-score metric. From Table 5.3, we have noticed that the results of our model are better in almost all of the accuracies. Also, we have used the number of parameters of the model to compare the size of our model with theirs. Our model utilizes approximately 615k parameters, while their model uses 2.5M parameters, meaning that our model is much smaller than theirs, which mean saving a lot of computational resources and much faster to train.

Table 5.3. *Comparison between our model and Agbo-Ajala & Viriri (2020) model in term of F1-score*

	0-2	**3-7**	**8-14**	**15-20**	**21-27**	**28-45**	**46-65**	**66+**	**Overall**
Proposed Model	96.92	93.38	87.86	82.72	84.46	86.51	87.38	90.82	86.55
	0-2	**4-6**	**8-13**	**15-20**	**25-32**	**38-43**	**48-53**	**60+**	**Overall**
Agbo-Ajala & Viriri (2020) Model	94.58	82.35	85.43	76.33	87.15	67.85	62.68	91.08	81.58

In Table 5.4, we compare our work with some other works that follows the same methodology as age estimation task. Our work outperforms all other works in term of accuracy and number of parameter.

Table 5.4. *Comparison between proposed work and other works with same methodology*

Reference	Year	Approach	Accuracy	F1-score	Number of parameters
Levi & Hassncer (2015)	2015	CNN + dropout	50.7%	-	-
Aydogdu & Demirci (2017)	2017	4C2FC	46.39%	-	-
Qawaqneh et al. (2017)	2017	VGG + dropout	59.9%	-	-
Zhang et al. (2017)	2017	RoR	67.3%	-	-
Rothe et al. (2018b)	2018	DEX	64.0%	-	-
Agbo-Ajala & Viriri (2020)	2019	4C2FC	84.8%	81.58%	2.5M
Proposed Work	**2022**	**7C2FC**	**85.55%**	**86.55%**	**615K**

6. CONCLUSION AND FUTURE WORK

The process of estimating someone's age from his facial image is considered to be difficult due to the fact that people age in a variety of diverse ways depending on the circumstances of their lives. Age estimation systems are developed to address the age estimation task. The purpose and applications of these systems are discussed in details within this work. Also, the most recent facial age datasets, CNN architectures and related works on age estimation task are presented in this work. The challenge of age estimation has been approached by us as if it were a classification task, with eight distinct age groups being used. For the classification of unconstrained facial images, the proposed DCNN-based classification model is used. Four models are presented depending on four different image resolutions and the best model, which is model-100, is chosen to be compared with other recent related works. This proposed model outperformed all other related works in every aspect. The proposed model depended on the DCNN architecture's capability of performing feature extraction and classification well. The satisfactory performance of the classification model can be attributed, in great portion, to the novel developed DCNN architecture. This architecture was trained using a combination of two small and clean datasets (the UTKFace dataset and the Facial-age dataset), and an augmentation process was used to increase the number of images. A comprehensive evaluation of the proposed model's performance on the augmented and combined dataset demonstrates that our approach is applicable to unconstrained facial images and achieves very good results compared to other related works.

From the perspective of where this research can be headed in the future, a reliable face detection method might be used at the preprocessing stage to crop the face image from the original image. This would be done in order to remove unwanted background elements, thus avoiding the confusion that may happen to the model with elements that have no use in the classification. The process of training a model might also benefit from having access to a clean, large, and well-balanced dataset. As time goes on, the exact age classification of a human face will become an interesting area of research due to the numerous commercial applications that it will have.

REFERENCES

Agbo-Ajala, O., & Viriri, S. (2020). *Face-Based Age and Gender Classification Using Deep Learning Model* (pp. 125–137). https://doi.org/10.1007/978-3-030-39770-8_10

Agbo-Ajala, O., & Viriri, S. (2021). Deep learning approach for facial age classification: a survey of the state-of-the-art. *Artificial Intelligence Review*, *54*(1), 179–213. https://doi.org/10.1007/s10462-020-09855-0

Agustsson E, Timofte R, Escalera S, Baro X, Guyon I, & Rothe R. (2017). *12th IEEE International Conference and Workshops on Automatic Face and Gesture Recognition : FG 2017 : proceedings : 30 May - 3 June 2017, Washington, D.C.*

Al-Shannaq, A. S., & Elrefaei, L. A. (2019). Comprehensive Analysis of the Literature for Age Estimation from Facial Images. *IEEE Access*, *7*, 93229–93249. https://doi.org/10.1109/ACCESS.2019.2927825

Antipov, G., Baccouche, M., Berrani, S.-A., & Dugelay, J.-L. (2016). Apparent Age Estimation from Face Images Combining General and Children-Specialized Deep Learning Models. *2016 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 801–809. https://doi.org/10.1109/CVPRW.2016.105

Aydogdu, M. F., & Demirci, M. F. (2017). Age Classification Using an Optimized CNN Architecture. *Proceedings of the International Conference on Compute and Data Analysis - ICCDA '17*, 233–239. https://doi.org/10.1145/3093241.3093277

Azam Bastanfard, Melika Abbasian Nik, & Mohammad Mahdi Dehshibi. (2007). Iranian Face Database with age, pose and expression. *2007 International Conference on Machine Vision*, 50–55. https://doi.org/10.1109/ICMV.2007.4469272

Berg, A., Oskarsson, M., & O'Connor, M. (2020). *Deep Ordinal Regression with Label Diversity*. https://doi.org/10.1109/ICPR48806.2021.9412608

Chen, B.-C., Chen, C.-S., & Hsu, W. H. (2015). Face Recognition and Retrieval Using Cross-Age Reference Coding With Cross-Age Celebrity Dataset. *IEEE Transactions on Multimedia*, *17*(6), 804–815. https://doi.org/10.1109/TMM.2015.2420374

Chen, S., Zhang, C., Dong, M., Le, J., & Rao, M. (2017). Using Ranking-CNN for Age Estimation. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 742–751. https://doi.org/10.1109/CVPR.2017.86

Choi, S. E., Lee, Y. J., Lee, S. J., Park, K. R., & Kim, J. (2011). Age estimation using a hierarchical classifier based on global and local facial features. *Pattern Recognition*, *44*(6), 1262–1281. https://doi.org/10.1016/j.patcog.2010.12.005

Chollet, F. (2016). *Xception: Deep Learning with Depthwise Separable Convolutions*.

Eidinger, E., Enbar, R., & Hassner, T. (2013). Age and Gender Estimation of Unfiltered Faces. In *IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY*. http://www.adience.com

Fukushima, K. (1980). Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biological Cybernetics*, *36*(4), 193–202. https://doi.org/10.1007/BF00344251

GoogLeNet architecture. (n.d.). Retrieved July 24, 2022, from https://developer.ridgerun.com/wiki/images/thumb/e/eb/Googlenet.png/1800px-Googlenet.png

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep Residual Learning for Image Recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 770–778. https://doi.org/10.1109/CVPR.2016.90

Hubel, D. H., & Wiesel, T. N. (1968). Receptive fields and functional architecture of monkey striate cortex. *The Journal of Physiology*, *195*(1), 215–243. https://doi.org/10.1113/jphysiol.1968.sp008455

Huo, Z., Yang, X., Xing, C., Zhou, Y., Hou, P., Lv, J., & Geng, X. (2016). Deep Age Distribution Learning for Apparent Age Estimation. *2016 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 722–729. https://doi.org/10.1109/CVPRW.2016.95

Jordan, M. I., & Mitchell, T. M. (2015). Machine learning: Trends, perspectives, and prospects. *Science*, *349*(6245), 255–260. https://doi.org/10.1126/science.aaa8415

Keras Models. (n.d.). Retrieved August 20, 2022, from https://keras.io/api/applications/

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2017). ImageNet classification with deep convolutional neural networks. *Communications of the ACM*, *60*(6), 84–90. https://doi.org/10.1145/3065386

LeCun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W., & Jackel, L. D. (1989). Backpropagation Applied to Handwritten Zip Code Recognition. *Neural Computation*, *1*(4), 541–551. https://doi.org/10.1162/neco.1989.1.4.541

Lecun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, *86*(11), 2278–2324. https://doi.org/10.1109/5.726791

Levi, G., & Hassncer, T. (2015). Age and gender classification using convolutional neural networks. *2015 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 34–42. https://doi.org/10.1109/CVPRW.2015.7301352

Liu, X., Li, S., Kan, M., Zhang, J., Wu, S., Liu, W., Han, H., Shan, S., & Chen, X. (2015). AgeNet: Deeply Learned Regressor and Classifier for Robust Apparent Age Estimation. *2015 IEEE International Conference on Computer Vision Workshop (ICCVW)*, 258–266. https://doi.org/10.1109/ICCVW.2015.42

Moschoglou, S., Papaioannou, A., Sagonas, C., Deng, J., Kotsia, I., & Zafeiriou, S. (2017). AgeDB: The First Manually Collected, In-the-Wild Age Database. *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 1997–2005. https://doi.org/10.1109/CVPRW.2017.250

Nam, S. H., Kim, Y. H., Truong, N. Q., Choi, J., & Park, K. R. (2020). Age Estimation by Super-Resolution Reconstruction Based on Adversarial Networks. *IEEE Access*, *8*, 17103–17120. https://doi.org/10.1109/ACCESS.2020.2967800

Niu, Z., Zhou, M., Wang, L., Gao, X., & Hua, G. (n.d.). *Ordinal Regression with Multiple Output CNN for Age Estimation*.

Original ResNet-18 Architecture. (n.d.). Retrieved July 25, 2022, from https://www.researchgate.net/figure/Original-ResNet-18-Architecture_fig1_336642248

Panis, G., Lanitis, A., Tsapatsoulis, N., & Cootes, T. F. (2016a). Overview of research on facial ageing using the FG-NET ageing database. *IET Biometrics*, *5*(2), 37–46. https://doi.org/10.1049/iet-bmt.2014.0053

Panis, G., Lanitis, A., Tsapatsoulis, N., & Cootes, T. F. (2016b). Overview of research on facial ageing using the FG-NET ageing database. *IET Biometrics*, *5*(2), 37–46. https://doi.org/10.1049/iet-bmt.2014.0053

Punyani, P., Gupta, R., & Kumar, A. (2020). Neural networks for facial age estimation: a survey on recent advances. *Artificial Intelligence Review*, *53*(5), 3299–3347. https://doi.org/10.1007/s10462-019-09765-w

Qawaqneh, Z., Mallouh, A. A., & Barkana, B. D. (2017). *Deep Convolutional Neural Network for Age Estimation based on VGG-Face Model*.

Qiu Jiayan. (2016). *Convolutional Neural Network based Age Estimation from Facial Image and Depth Prediction from Single Image*. Australian National University.

Ranjan, R., Zhou, S., Chen, J. C., Kumar, A., Alavi, A., Patel, V. M., & Chellappa, R. (2015). Unconstrained Age Estimation with Deep Convolutional Neural Networks. *2015 IEEE International Conference on Computer Vision Workshop (ICCVW)*, 351–359. https://doi.org/10.1109/ICCVW.2015.54

Ricanek, K., & Tesafaye, T. (n.d.). MORPH: A Longitudinal Image Database of Normal Adult Age-Progression. *7th International Conference on Automatic Face and Gesture Recognition (FGR06)*, 341–345. https://doi.org/10.1109/FGR.2006.78

Rothe, R., Timofte, R., & van Gool, L. (2018a). Deep Expectation of Real and Apparent Age from a Single Image Without Facial Landmarks. *International Journal of Computer Vision*, *126*(2–4), 144–157. https://doi.org/10.1007/s11263-016-0940-3

Rothe, R., Timofte, R., & van Gool, L. (2018b). Deep Expectation of Real and Apparent Age from a Single Image Without Facial Landmarks. *International Journal of Computer Vision*, *126*(2–4), 144–157. https://doi.org/10.1007/s11263-016-0940-3

Saha Sumit. (2018). *A Comprehensive Guide to Convolutional Neural Networks — the ELI5 way*.

Simonyan, K., & Zisserman, A. (2014). *Very Deep Convolutional Networks for Large-Scale Image Recognition*.

Szegedy, C., Wei Liu, Yangqing Jia, Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., & Rabinovich, A. (2015). Going deeper with convolutions. *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1–9. https://doi.org/10.1109/CVPR.2015.7298594

WIKI_ART. (2019). *Facial-age dataset on Kaggle website.* https://www.kaggle.com/datasets/frabbisw/facial-age

Yun Fu, Guodong Guo, & Huang, T. S. (2010). Age Synthesis and Estimation via Faces: A Survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *32*(11), 1955–1976. https://doi.org/10.1109/TPAMI.2010.36

Zhang, K., Gao, C., Guo, L., Sun, M., Yuan, X., Han, T. X., Zhao, Z., & Li, B. (2017). Age Group and Gender Estimation in the Wild With Deep RoR Architecture. *IEEE Access*, *5*, 22492–22503. https://doi.org/10.1109/ACCESS.2017.2761849

CURRICULUM VITAE

First and Surname : Ahmad ALSALEH

Foreign Language : Arabic, English, Turkish

Education and Experience:

- 2022, Anadolu University, Institute of Graduate Programs, Department of Computer Engineering.
- 2015, Aleppo University, Faculty of Electrical and Electronic Engineering, Department of Computer Engineering.