



ISLAMIC UNIVERSITY OF TECHNOLOGY (IUT)

Age Estimation from Facial Images Using Transfer Learning and k-fold Cross-Validation

Authors

Mahruf Islam Prottoy(160041007)

S. M. Shihab Uddin(160041042)

Md. Samin Morshed(160041023)

Supervisor

A.B.M. Ashikur Rahman

Assistant Professor,

Department of Computer Science and Engineering, IUT

*A thesis submitted in partial fulfilment of the requirements
for the degree of B. Sc. Engineering in Computer Science and Engineering*

Academic Year: 2019-2020

Department of Computer Science and Engineering (CSE)

Islamic University of Technology (IUT)

A Subsidiary Organ of the Organization of Islamic Cooperation (OIC)

Dhaka, Bangladesh

October 5, 2021

Declaration of Authorship

This is to certify that the work presented in this thesis is the outcome of the analysis and experiments carried out by Mahruf Islam Prottoy, Shihab Uddin and Md. Samin Morshed under the supervision of A.B.M. Ashikur Rahman, Assistant Professor, Department of Computer Science and Engineering, Islamic University of Technology (IUT), Dhaka, Bangladesh. It is also declared that neither this thesis nor any part of it has been submitted anywhere else for any degree or diploma. Information derived from the published or unpublished work of others has been acknowledged in the text and a list of references is given.

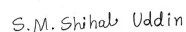
Authors:

Mahruf Islam Prottoy



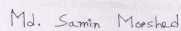
Student ID - 160041007

S. M. Shihab Uddin



Student ID - 160041042

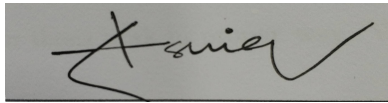
Md. Samin Morshed



Student ID - 160041023

Approved By:

Supervisor:

A handwritten signature in black ink on a grey rectangular background. The signature is cursive and appears to read 'Ashikur Rahman'.

A.B.M. Ashikur Rahman
Assistant Professor,
Department of Computer Science and Engineering (CSE)
Islamic University of Technology (IUT)

Acknowledgement

We would like to express our grateful appreciation for **Professor Md. Hasanul Kabir**, Department of Computer Science & Engineering, IUT for being our adviser and mentor. His motivation, suggestions and insights for this research have been invaluable. Without his support and proper guidance this research would never have been possible. His valuable opinion, time and input provided throughout the thesis work, from first phase of thesis topics introduction, subject selection, proposing algorithm, modification till the project implementation and finalization which helped us to do our thesis work in proper way. We are really grateful to him.

We are also very much grateful to **A.B.M. Ashikur Rahman**, Assistant Professor, Department of Computer Science & Engineering, IUT from the core of our heart for his valuable inspection and suggestions on our proposal of Age Estimation using CNN with Ensemble Method. His motivation and constant help from various sources regarding our thesis work has indeed played a big part in completing the research work we carried out.

+

Abstract

The increasing use of video-based security systems and robotics has increased research on the image analysis of human faces. Thus, face recognition, face detection, gender classification, and facial expression recognition have attracted much attention in digital image processing field. [1, 2, 3, 4, 5]. Estimating the age of a person from the analysis of his/her face image is a relatively new research topic. This thesis is focused on exploring different CNN approach on increasing accuracy in age classification. To achieve higher accuracy we used a few pre existing models and used pre trained weight which is trained for face detection, on Wild and Youtube face dataset. We also fine tuned the model and used k-fold validation. This method shows us higher accuracy. Then we compared our work with already existed papers. In other words, we tried to increase accuracy in age classification. We also compared our work with pre existed paper to show performance of our model relative to other models

Keywords: CNN, VGG16, Resnet50, Senet50, UTKFace, classification, model.

Contents

1	Introduction	5
1.1	Overview	5
1.2	Problem Statement	6
1.3	Motivation and scope of research	6
1.4	Research Challenges	7
1.5	Thesis Outline	10
2	Background study/Literature Review	11
2.1	Background study	11
2.2	Hand crafted feature extraction:	12
2.2.1	LBP (Local binary patterns [6])	12
2.2.2	BSIF (Binarized statistical image features)[7]	14
2.2.3	HOG (Histogram of oriented gradients[8])	14
2.3	Deep feature extraction	15
2.3.1	Convolutional neural network(CNN)	15
2.4	Transfer learning	20
2.4.1	Using scenario of transfer learning	21
2.4.2	Benefits of transfer learning	22
2.4.3	Approach to transfer learning	23
2.4.4	Customize a pretrained model	24
2.5	Literature review	25
3	Proposed Approach	27
3.1	Skeleton of Proposed Method	27
3.1.1	Neural Network model	27
3.1.2	Weight	30
3.2	Customizing model	30
3.2.1	Fine tuning	30
3.2.2	k-fold cross validation	31

3.2.3	Adam optimization	32
3.3	Data set	32
3.3.1	Data-set Augmentation	33
4	Model Training and Result Analysis	34
4.1	Training	34
4.1.1	Training configuration:	34
4.1.2	Test Bench	35
4.2	Result analysis	35
4.2.1	Evaluation Metrics	35
4.2.2	Result	36
4.2.3	Example Outcome	40
4.3	Weakness	41
4.4	Comparison with previous works	42
5	Conclusion	43
5.1	Summary	43
5.2	Future Work	43

List of Figures

1	Type of feature extraction method	11
2	LBP code generating[9]	13
3	Input image and HOG output.	15
4	Basic CNN structure.	16
5	Convolutional layer structure	17
6	Pooling layer structure	18
7	Flatten layer structure	18
8	Fully connected layer structure	19
9	Transfer learning process	21
10	Transfer learning benefits. [10]	22
11	k fold cross validation visualization	25
12	Vgg16 model architecture.	28
13	Resnet50 model architecture	29
14	UTKFace dataset example.[11]	33
15	Train accuracy	36
16	Validation accuracy of models in each fold	37
17	Precision of all model	38
18	f1 score of all model	39
19	Heatmap of confusion matrix: i. Resnet50, ii. VGG16, iii.Senet50.	40
20	Example output: i. True age=2, ii. True age=12, iii. True age=32.	40
21	Example output: i. True age=50, ii. True age=70.	40

List of Tables

1	Accuracy of each fold.	37
2	Precision of models in each fold.	38
3	f1 score of models in each fold.	39
4	Performance compare with other models	42

1 Introduction

1.1 Overview

Age and gender information have many important usage in various real world applications, such as social understanding, biometrics, identity verification, video surveillance, human-computer interaction, electronic customer, crowd behavior analysis, online advertisement, item recommendation, and many more. Although having a huge number of applications, being able to automatically estimate age from different grouped facial images is a very complex problem, due to the different sources of intra-class variations found on the facial images of people, which makes the use of these models in real world applications limiting.

There have been an enormous number of works carried out for age/age-group prediction or estimation in the past years. The previous works were primarily based on hand-crafted features extracted from facial images followed by a classifier. But with the massive success due to using deep learning models in numerous computer vision problems in the past few years [12, 13, 14, 15, 16] the more recent works on age and gender predictions are generally shifted towards deep neural networks based models. The features extracted from Neural Network do not need to be designed by human, which saves a lot of time and reduce complexity.

In this work, we are proposing a CNN framework to simultaneously estimate the age and gender from face images. We used three pre-built models for our work. We bring some change to the models. This thesis is focused on exploring different CNN approach on increasing accuracy in age classification. To achieve higher accuracy we used a few pre-existing models- VGG16, Resnet50, Senet50 and used weight of VGG-Face on face detection, which is pretrained on Wild and Youtube face dataset. We also fine tuned the model and used k-fold validation. This method shows us a higher accuracy. Then we compared our work with already existed papers, which works on age estimation on same UTKFace dataset. We

tried to increase accuracy in age classification. We also compared our work with pre existed paper to show performance of our model relative to other models.

1.2 Problem Statement

Face recognition is one of the most efficient methods to recognize a person by facial properties. How these features are used and how the method is optimized has been a very standard challenge for the computer vision researchers, for a long time. Facial recognition was, in past, mostly used for documents verification like assets such as registration of land, verification of passport and identification of a specific face in maximum-security facilities. There have been many works in age estimation already which use different method to estimate an age group or a distinct age. In our work we are trying to go with age estimation classification that utilizes Transfer Learning with ensemble method to increase the validation and test accuracy. Our Problem Statement can be stated as follows: Our work would be to make a simpler, efficient and accurate similar to modern state-of-the art age estimation systems using transfer learning and ensemble method which uses K-fold cross validation on top. This will greatly reduce the complications of the modern existing methods and yet will show the same amount of accuracy.

1.3 Motivation and scope of research

Age estimation can be used in plethora of applications. They consists of log in systems, HCI, face identification, data mining and other organizations. Various applications for the categories mentioned above are as follow:

Access Control: In many cases, restrictions are applied based on age to control virtual or physical access. For example we can show that many places have entry restrictions below a certain age, also the same can be said about many websites or even for purchase of products like alcoholic beverages and cigarettes by minors and who do not fulfill age requirements for those products. The systems as of now, in these cases the jurisdiction is enforced by human judgement, or by some necessary papers like licenses possessed by the person. In the contrary, however, applying

automatic facial age estimation can be useful for providing objective, accurate and congenital estimation of the age, and thus will help solving the problem of access control in either different places or websites.

Human computer Interaction : People from a multitude of age groups interact with advanced machines such as computers or other machines of sort for a variety of reasons and not all are the same. Adjusting the user interface of a platform can be done along with embedded age estimation system to measure age the system user is of, and thus calibrating the whole system for the user. Example: children love animated interface and for them that can be managed, while for older users using the machine the system can use simple, bold and bigger texts for reading convenience. For example, Information kiosks can use these systems to deliver information to different people of different age groups.

Age Progression for Criminal Identification: Using different age progressing algorithms which can detect how a person will look in a future age, can be used to identify runaway fugitives. This technique however more often than not needs a huge amount of information regarding the age of the criminal. This is where automatic age estimation systems comes. Thus it can help build a stable age progression algorithm which will identify a person regardless of which age he or she escaped from the law.

Database Management and Retrieval: We can use age estimation to fetch specific aged personnel data and and manage the database those are in, thus paving the way for a data mining system which can collect photos of various age group from internet image repositories and archives.

1.4 Research Challenges

As time passes forward, the age estimation sector has dealt with quite a few challenges, and they keep on growing as new ways of facial age estimation are invented. There are some ways to reduce their effect but it is still hard to totally remove those problems. Age estimation has shown similar challenges faced in the a multitude of

different estimating or interpreting works regarding facial image. A few of them are facial detection, recognition, gender identification, expressions etc. Attributes like deformation in facial appearance due to different expression, interpersonal variation, facial orientation, illumination disparity and different occlusions on face images have a detrimental to the performance of the age estimation.

The challenges in age detection can be categorized into following labels:

Physical factors:

- **Intrinsic factors:** These factors include genetic difference, make-up, different age growth etc. These are one's own factors. Not all of them are control-able. Genetic features may confuse a model as people of different places have different age features at different ages. Make-up can also cause problems, as this hides facial features, which is important for age detection.
- **Extrinsic factors:** These factors include Pollution, environment, UV-ray etc. These factors are not control-able. Pollution, environment affects input images, which may cause noise or low quality images. These affect the learning and detection phase of a model.

Facial expression: Facial expression is a very big factor while addressing the age estimation process. Most of the time humans express their mental state by their face. These expressions give different texture information of the same face. So, it is hard for a model to identify them as the same.

Lighting variations: Under an experimentally controlled environment where the front facing images are captured in static lighting conditions with no expressions the age can be accurately detected. But in a real world environment it is not straightforward to get the same accuracy level because of the different facial poses, camera setups and illumination conditions.

Image quality: Image quality is a big part of age identification. Bad image quality gives wrong facial information. So, it is important to manage datasets with good image quality.

Face occlusions: Face occlusions denote scarf, sunglasses, mask, facial hair etc. It hampers the performance of facial recognition greatly. But in real world scenario face occlusion is a common attribute especially in non-cooperative surroundings. For example, occlusion in the upper part of the face and thus the absence of important information costs a mere margin of the overall performance drop. However, the main factor is thought to be errors in facial feature localization which ultimately causes registration errors.[17] Getting proper datasets to train to give accurate prediction of age for occluded images is also a challenge

Some of the significant problems which are unique to age estimation only are discussed as below:

Sparse Distinction Amount between inter age groups: It is a problem mainly occurs mostly in Adult age groups in age estimation classification. Usually what happens is the estimation problem fails to classify it between adjacent groups, for example, an age group between 35-39 and age group of between 40-44 will have some wrong classifications.

Difference in aging: People age differently and so, many factors are in work in different ways for aging in a random way. [18] shows that different people in the same age group can have different types of wrinkle features, so that contributes a lot to age estimation. These challenges, as mentioned, causes age estimation a bit more challenging and thus the performance is not up to the mark. Physical factors like the color of skin, genetic makeup have a negative effect on the age estimation process. Consequently, we specific age estimation methods are needed for different group of people.

External Factors: A lot of external factors influence how a person's aging pattern. Health condition, psychology, lifestyle, wearing make-up, and usage of anti-aging cosmetics to intervene with the aging process or undergoing cosmetic surgery affects the appearance of a person causing performance issues in the age estimation process.

Availability of Data: To accurately develop an age estimation system we need appropriate datasets. First we will train the dataset and then we will test it. The dataset should contain multiple sets of images of the same age group, and a large number of sets at that. A collection of images captured in the past is required as a dataset since people can't directly control the aging process. Mainly two datasets are publicly available which are MORPH [19] and FG-NET [20] which are mainly focused to support the research of facial features. But both of the datasets don't fulfill all the requirements. The former contains a relatively less number of images of every subject and the latter emphasizes on the significance of related dissimilarity of non aging subjects.

1.5 Thesis Outline

In Chapter 1, we discuss which way we are proceeding through our study. Chapter 2 deals with the literature review and how we progressed into our cause from it. In chapter 3, we stated the skeleton of our proposed method, algorithm and necessary procedures we followed. This provides a detailed insight into our working method, the ensemble method with K-fold on top of Deep learning using pre-trained models. Chapter 4 shows the results and comparative analysis of successful implementation of our proposed method. The final segment of this study contains all the references and credits used.

2 Background study/Literature Review

In this chapter we will try to give a brief overview of the general procedure of age classification, different types of Facial feature representation approaches and different term related to age classification.

2.1 Background study

Modern technology uses various security systems of both manual and automated types. Security and surveillance systems these days must analyze human faces in image or video format. So, in modern image processing field. age estimation is very crucial where we can use the technology to evaluate the age and gender of a subject without the need to identify the subject.

Feature extraction is an important part of age estimation. There are two main type of feature extraction method:

- Hand crafted feature extraction
- Deep feature extraction

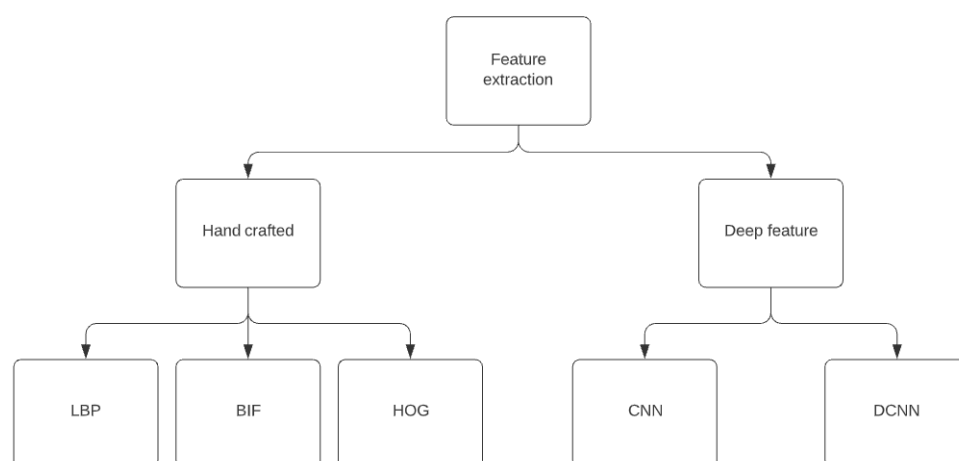


Figure 1: Type of feature extraction method

The handcrafted features were mostly used in the beginning of feature extraction. These are relatively older methods. Handcrafted features were used along

with machine learning for object identification mainly. SVM can also be used in deep learning problems here. In the contrary, modern neural networks do not use hand crafted methods for feature extraction. Rather they use convolution to extract deep features from the image.

2.2 Hand crafted feature extraction:

Hand Crafted features refer to properties which are extracted using various algorithms from the information present in the image. In this case human design what feature to find and how to find. For example, some features can be easily extracted from image on basis of edges, corners, lines, dots etc. A basic edge detector algorithm works by finding areas where the image intensity changes suddenly. By thresholding matrix value it is possible to find edge, line, dot, shade change etc. But human also need to decide how to use this feature and which one should be used. Later, NN models are being used in these filtered image to learn these features and use the models in different problem solving.

There are different hand crafted feature extraction method. Some of them are:

- LBP (Local binary patterns)
- BSIF (Binarized statistical image features)
- HOG (Histogram of oriented gradients)

Here are some small description on them:

2.2.1 LBP (Local binary patterns [6])

In the case of image texture, Local Binary pattern is one of the most basic of the properties. Local binary Pattern can be used to describe the texture of the image. By generating a code for each pixel, the Local Binary Pattern Operator makes use of the textures it found in the image. The code is generated when a single pixel is compared to its adjacent ones. After getting all the pixel values,

a histogram can be generated, which in turn can be used as the texture feature, and so the whole Local Binary Pattern code is generated. In order to generate the Local Binary Pattern code, the 8 adjacent neighbours have to be found out. Then they are compared with the central pixel. Here, if the neighbouring value is greater than the central pixel then it is assigned 1, and 0 for otherwise. After following the same procedure for every pixel, the code is generated. Figure shows how LBP code for a 3×3 area is calculated. This process is computed across the whole image.

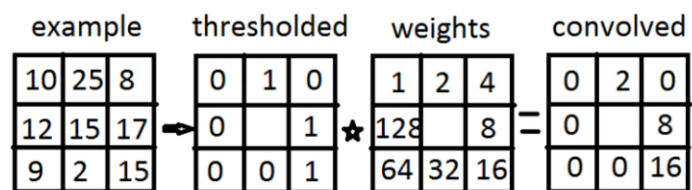


Figure 2: LBP code generating[9]

There are two types of LBP code: uniform and non-uniform. Not all code have same significance, the former pattern consisting of bitwise transitions of at most two of from zero to one or one to zero. It is accounted for almost 90% of all patterns, if (8,1) neighbourhood is used. [21]

To compute the histogram, bins are created. All uniform codes are in separate bins where whereas a single bin is created for the non uniform ones. As there are 58 uniform bins, total number of bins are 59. The histogram is finally generated when the pixel numbers are plotted in the bins.

The histogram consists of the information about the distribution of spots, edges, region variation etc. of the complete image. For the most optimized representation, the image is segmented into different smaller regions which in turn possess one one histogram of their own. These small portions can be combined to build a global descriptor in which weights can be assigned in accordance with their respective important information stored within. Like $X=[x(0), x(1), x(2)\dots, x(n-1)]$ can be assigned for each region.

For the classification, LBP technique calculates the minimum distance between the mean histogram of the target group and the acquired histogram of the tested image. Then the images can be sorted in respect with their distances. for calculating the distances, in LBP technique we use Euclidean distances, Manhattan distances or Chebyshev Distances.

2.2.2 BSIF (Binarized statistical image features)[7]

BSIF works by generating a string of binary code. This code is generated for the image being trained of tested and works as a local binary descriptor. The descriptor works with intensity variation. The descriptor can also be regarded as a texture descriptor because the generated code for each pixel brings out the characteristics of the texture properties. So, it can work the same way as LBP [21] or it can work like a magnitude for each phase [22]. For the binary code string, taking the binary value of the linear filter response, the value of a single element can be computed. Every single bit represents a distinct filter. The amount of filter refers to the length of the bit string. The collection of filters is trained by a set of raw or from the dataset image patches. Thus, the maximized independence according to the statistics of the filter responses calculate the descriptors[23].

2.2.3 HOG (Histogram of oriented gradients[8])

Histogram of Oriented Gradients is nothing more than feature descriptor. In order to retrieve image features, HOG descriptors can be used. It has gained popularity in computer vision tasks for object detection. Here, for pre process image is needed to be resized so that the image can be divided into 8*8 or 16*16 pixel square easily. In each square we will check every pixel and will find change in Cartesian axes G_x and G_y . This gradient and magnitude of every pixel of a pixel square will be calculated and used to make a histogram, which will be converted into a feature vector for every pixel square. By concatenating all feature vector we will find the HOG output.

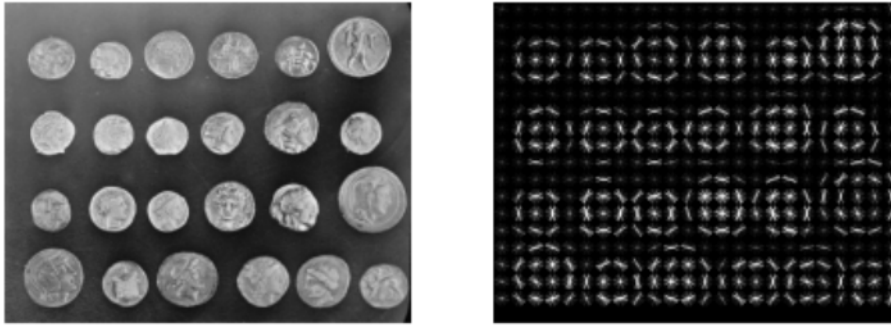


Figure 3: Input image and HOG output.

In HOG feature extraction the output shows gradient change. The output is convoluted with target object model. Then the convoluted output put into a classifier to classify the object.

2.3 Deep feature extraction

In a neural network layer, we have inputs and outputs. If an input response has some sort of consistency, which corresponds to the output, it can be regarded as a deep feature. The importance of the feature depends on the premature activation of the response. Deep feature extraction is completely done by CNN. In this method a CNN is trained with dataset related to target work. CNN model trains by extracting feature from image. Image passed through different filter and internal node of the model extract feature from them. The main difference from hand crafted features with neural networks is that neural networks can extract the important feature without designing and instructing. CNN model extract features by itself based on how it is build and how it is trained. Extracted feature in CNN model is not always easily understandable. But it is easy to design and less complicated.

2.3.1 Convolutional neural network(CNN)

Convolutional neural network has always been about containing a complex mathematical problem. Convolutional neural networks are considered as a unique kind of neural network which uses convolution instead of applying general matrix mul-

tiplication. It must have at least one of the convolutional layers among all other layers. Basic CNN structure has three parts. They are defined as input layer, output layers and the intermingled hidden layers. In any CNN, all middle layers are called hidden layers. In a CNN, convolutions are performed in the hidden layers. Typically in the hidden layer, there we have a layer that multiplies the weight and the input value, and an activation function, most of time it is Tanh or Retified Linear Unit function. There are also pooling layers which does the pooling function, fc or fully connected layers and last but not the least we have some layers called normalization layers. Different layer of CNN is described here:

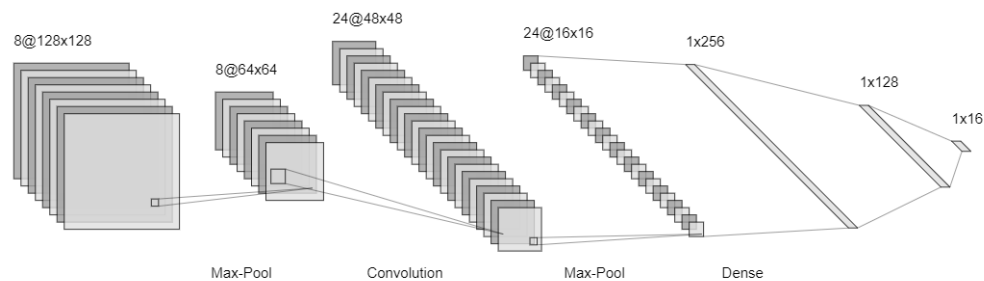


Figure 4: Basic CNN structure.

2.3.1.1 Convolutional layers

By scanning the whole image with a filter, convolutional layers generate a feature map for every single feature. So, the whole image now has a feature image after it has proceeded after the convolutional layers. The feature map is used to predict the class probabilities. A convolutional layer of a NN have some parameters. Such as: Convolutional filters/kernels, how many input and out channels it has, how deep the convolutional kernel is, hyper-parameters of the convolution operation, like padding size and stride. This layer works as a filter. To convolute image with kernel, the image is padded with '0's. The convoluted output is used as input in next layer.

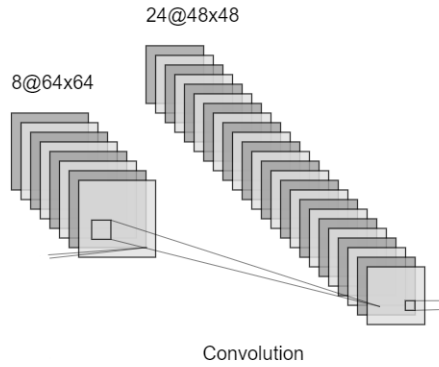


Figure 5: Convolutional layer structure

2.3.1.2 Pooling

Sometime convolutional networks have some pooling layers. They make the computation with the layers much easier. Pooling layers are applied in order to achieve dimensional reduction of the data. it achieves this when neuron cluster outputs are combined into one single neuron in a single layer. This is then used as input for the upcoming layer. We have local pooling, which is a combination of a small group of clusters, whereas the global one is combined from the whole set of neurons in a convolutional layer. There are two pooling methods which are used now-a-days. These are max-pooling method and average-pooling method. The maximum value of each cluster of a layer is taken and in the next layer that is used as input in max pooling method, and in average pooling method, it uses the average value of a cluster is used.

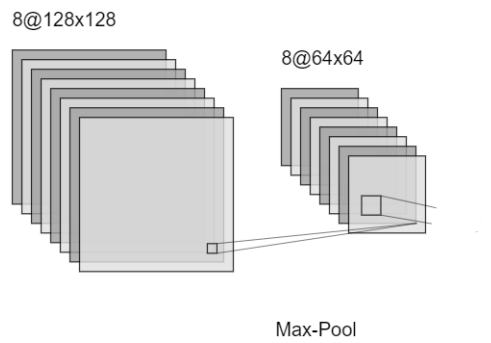


Figure 6: Pooling layer structure

2.3.1.3 Flatten layer

In flatten layer matrix is taken as input. Most of time the matrix is output of convolution or pooling layer. In flatten layer input matrix is turned into a single vector. Single vector is used as input in fully connected layer. So that it become easier to do dot(.) operation.

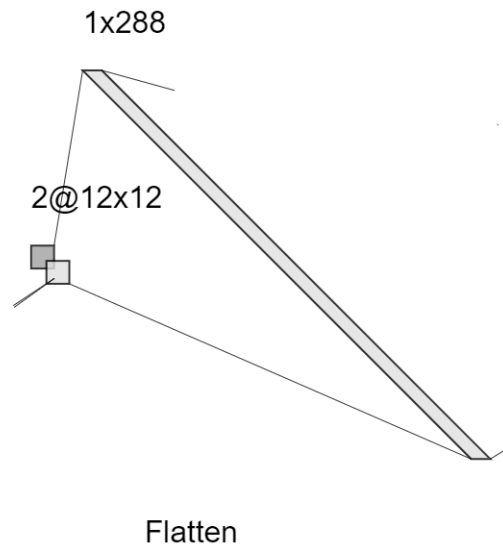


Figure 7: Flatten layer structure

2.3.1.4 Fully connected layer

In a fully connected layer, the whole set of nodes of previous layer is connected to

the full set of nodes of this layer. In every node of FC layer, input is multiplied with weight, then added with bias. The result the pass through a activation function. In back propagation stage, weight of these node changes for better result.

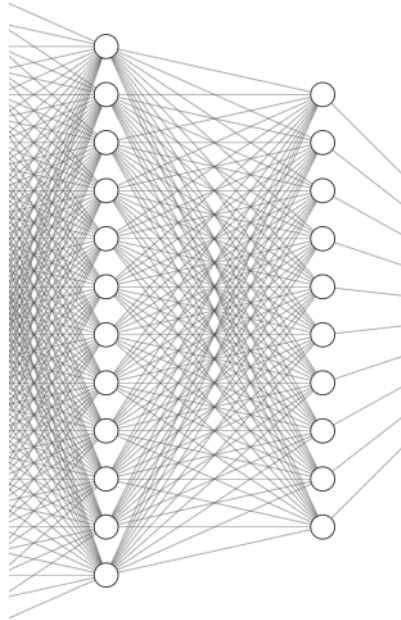


Figure 8: Fully connected layer structure

2.4 Transfer learning

One of the new approaches to machine learning is transfer learning. In this paper, a model that is developed and trained for a task is skill transferred to a second model using transfer learning.

Transfer Learning is one of the research problems in machine learning that mainly focuses on the storing of knowledge. The main idea is the knowledge or skill that is gained while solving one problem is applied to another related problem by transferring that knowledge. Transfer Learning is a process in which a model trained to perform a specific task is used in another model. For example, the knowledge gained while being trained to detect cat images can be used to detect other animal images. By using transfer learning a model can be built by saving time since it doesn't require to build a model from the scratch. Many pre-trained NN which are trained on natural images show some common traits. Most of the time they learn features similar to different filters and color blobs on the model's first layer. Such first-layer features don't appear to be specific to a particular task or dataset. These are some general features which can be used for many purposes and so they can be used for many datasets and tasks. In the first layer, these features occur in every situation regardless of the natural image dataset and cost function. The features of the first layer are called general features. For example, for successfully trained with supervised classification objective in a network of softmax output layer the output of each unit is specific to distinct class. For this output layer is called last-layer features specific

To use transfer learning, firstly we train a network on the basis of a dataset and task. Then the learned attributes are re-used or transferred to another network to be used for a task. This process works well if the features are general in both base and target tasks, instead of being specific to the base task.

In practice, very few times a CNN is trained from scratch because finding an appropriate dataset size that is sufficient to train the required model is relatively more difficult to find. So a common practice is to pre-train a CNN model on a larger dataset and use it to initialize or use as feature extractor for the required

task.

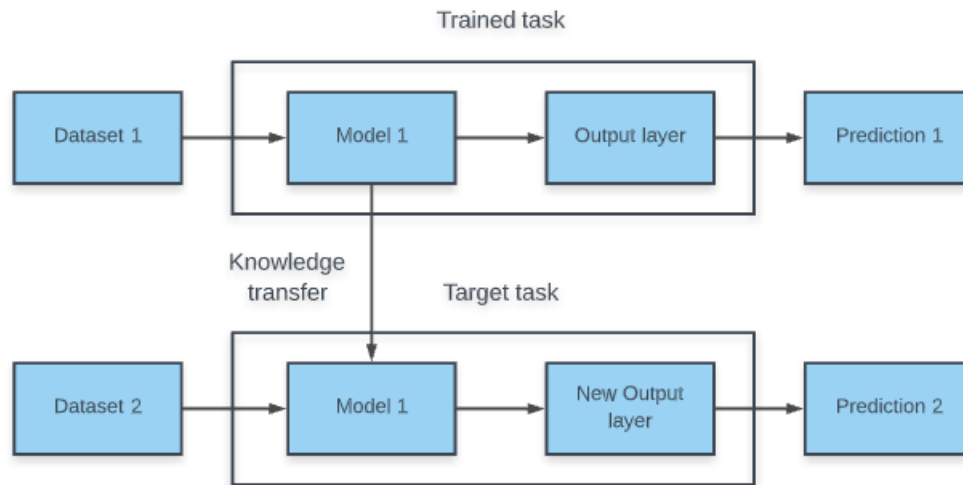


Figure 9: Transfer learning process

2.4.1 Using scenario of transfer learning

Transfer learning is a great method to transfer knowledge and train a model with a smaller dataset. It is hard to form rules that can specify the usage of TL, but some possible ways to use transfer learning on some scenarios are:

- TL can be applicable when there aren't adequate labeled datasets available in order to train a network from scratch. A model cannot train well when the dataset is small. But finding proper dataset for every type of work is hard. So transfer learning can be used in such cases.
- TL can be used if a model already exists which is trained on similar task and is trained on a large dataset. Models that are trained on larger dataset are more robust than models that are trained on smaller dataset. So usage of a pre-trained model can really increase the performance of the model
- TL can be used when target task and pre trained task have the same input. It does not always have to be trained on dataset of same topic. Pre trained model on same type of input.

2.4.2 Benefits of transfer learning

Transfer learning can be defined as an optimization problem. It gets better performance on a model and saves time in the process. It isn't assured that transfer learning will get good performance since it is dependent on the model it is transferring from being developed enough. Some of the benefits that might be gained for using transfer learning :

- Higher start: The initial gain (before fine-tuning the model) on the base model is high compared to a non-trained model. That means it requires less performance gain.
- Higher slope: The rate at which the performance or accuracy gain occurs while training the source model is more steep than others.
- Higher asymptote: The point of convergence of a model that is trained is better than the non-trained model yielding better performance. That means it can reach peak performance in less epoch.
- Less Computational Power: As it uses weights that are pre-trained and only require to train the weights of the few layers that are added at the end of the model requires a lot less computational power than CNNs.

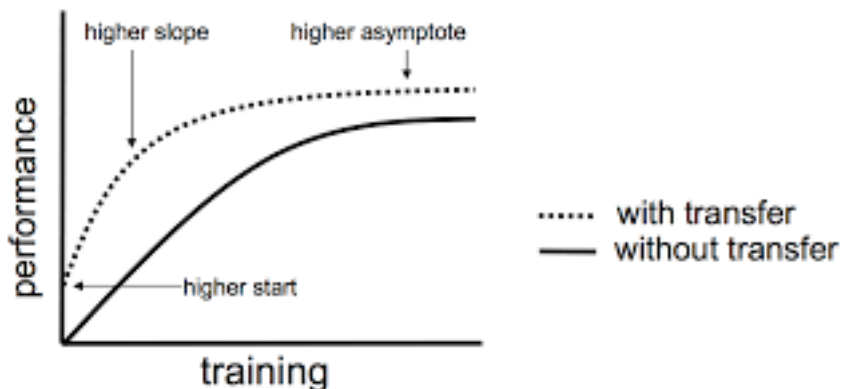


Figure 10: Transfer learning benefits. [10]

2.4.3 Approach to transfer learning

Two common approaches are as follows:

- **Training a model:** Training a model: If a task X doesn't have enough data to train a deep neural network model, an alternative is to find a related task Y with adequate data. Then the NN model is needed to train in a dataset of task Y. After that, the trained model is used to solve task X. Whether the whole model or only a few layers is needed to use depends mainly on the problem is tried to solve. If both tasks need the same type of input then reusing the model to make predictions for the new input can also be considered as an option. As an alternative, the task-specific layers and the output layer can be changed and retrained. Though it is transfer learning, but it is slow, as a new model is needed to be built and it is also not easy to build an efficient model. Also the time taken to train on a large dataset is a lot more as it is needed to be made transfer learning effective.
- **Using a pre trained model:** There are a lot of available pre-trained models. First, a pre-trained base model is to be selected from the available models. A lot of the research institutions usually release different models that contains large and challenging datasets which might include a large pool of candidate models to choose from. The pre-trained model is used as the starting point of the task which it is transferred to. Doing this can also involve changing the mode, using most or partial parts of the model depending on the technique used for modeling. Also the model may be required to adapt and refine on the input-output pair data available in order to run the task. It is easy to find a pre trained model in internet. They are made by doing trial and error and trained with large dataset and also no re training required, which save time.

2.4.4 Customize a pretrained model

Customizing model is good for transfer learning. It increase performance, learning speed faster. Here, are some customization method we used in our proposed model:

- Freezing: To do transfer learning, we use the concept of Freezing the layers. In simple terms, Freezing a layer means preventing that layer weights from being updated. In TL it not necessary to train the pre trained layers. Only training the fine tuned layers are enough. It saves time in training process. It also helps to gain higher accuracy in short time. Training full model does not give that much performance boost. So, freezing is a widely used method. In our method we froze all layer of pre trained model. We only trained newly added layers. It saves time but does not hurt performance that much.
- Fine-Tuning: Pre trained models are made for trained task, to use them for another task some change is needed to make. We change the output layer according to our requirement. We also added few layers to the model. We trained them on age dataset. Rest of the the model kept frozen. So, the rest of the model does not train. This makes the model useful for different specific task. In our case these newly trained layer and pre trained layers are used for age estimation.
- K-fold cross validation: In this procedure the train dataset is divided into several parts. These parts are called fold. In one iteration one fold is taken as test set and others as train set. In another iteration another fold is taken as test set. This process continues until all fold are used as test set. The procedure has a single parameter called k. k means the number of groups or folds, such as k=10 becoming 10-fold cross-validation. In our proposed model we used 5 fold cross validation. This means the train set is divided into 5 fold. In each iteration we use 4 fold as train set and the rest 1 as test set. In every iteration each fold is used as test set. Thus the model gets an efficient training. This helps to increase accuracy.

Estimation 1	Test	Train	Train	Train	Train
Estimation 2	Train	Test	Train	Train	Train
Estimation 3	Train	Train	Test	Train	Train
Estimation 4	Train	Train	Train	Test	Train
Estimation 5	Train	Train	Train	Train	Test

Figure 11: k fold cross validation visualization

2.5 Literature review

There are many works have done on age classification. They use different method to improve the performance of age classification.

In the [24] paper, they used MTCNN (Multi-Task Convolution Neural Network) with dynamic weight loss. Their method uses features which are gathered from a fully connected layers of a DCNN. It helps for better facial feature analysis. According to their research, different part of facial feature are gathered in different parts of a CNN. For example: lower layer gather information like edges and corners. Which helps to get the information of overall structure. It is more useful for pose estimation, object detection etc. again, higher layers gather the information about surface texture. They proposed a customized Facenet [25] for face recognition with ResNet V1 inception. First summed weight of each class of validation set is calculated. A fully connected layer is added to get better feature extraction. Then softmax layer change the value of dynamic weights to positive value. To contribute to the final loss the MTCNN assign higher weight for non-relevant task with lower loss to reduce overall loss. mini-batch Stochastic Gradient Descent is used to optimize weight loss.

Here [26], they took a different approach by integrating multiple CNN. They used evolutionary-fuzzy-integral in their process. In the process after preprocessing the images are used as train data for the CNNs. In the process they select optimal

fuzzy density values, which is based on best fitness value. To calculate this particle swarm optimization (PSO) method is used. Optimal fuzzy density values are then used to calculate fuzzy measures. After that the classifier are sorted and a set is created to calculate Sugeno and Choquet. Sugeno and Choquet are two fuzzy integral rules. The result with higher accuracy between them is chosen.

In here [27], the authors used transfer learning as base of their proposed model. They used weights of a pre trained FaceNet model. It was trained on VGGFace2 dataset. 3 separate additional layer with 128 neurons was added for 3 outputs (age, race, gender). Linear activation function was used in those additional layers. The common layers are optimized using combined loss. Output layer used softmax activation function.

In here[28], the paper mostly works on image pre processing. It proposed Deep EXpectation (DEX). By pre processing, the model detect face, align and crop the image, which is helpful for better prediction. Then VGG16 CNN model is used for age estimation. They also compared performance between using TL and not using TL.

Here, we proposed a method with commonly used technique. We used weight of VGGFace model. It was pre trained on Wild and YouTube dataset. We used the weight on three CNN models: VGG16, Resnet50 and Senet50. We fine-tuned those model by adding 5 additional layers on top. There was 1 flatten layer, 3 fully connected layers with ReLu activation function and 1 output layer with softmax function. Then we trained the model using UTKFace dataset. Then we used k-fold validation during training. We used k=5 so that train test ratio during training remains 80:20.

In the paper we tried to give a comparison with stated papers and our proposed method. All methods used UTKFace dataset for performance test and classification method for age estimation.

3 Proposed Approach

3.1 Skeleton of Proposed Method

In this section we provide the details of the proposed age prediction framework. We formulate this as a classification learning problem. In another word, a convolutional neural network to predict age.

Given the intuition that transfer learning can enable us to better predict her/his age. With k-fold validation the accuracy increase by a big margin. Through experimental study, we show that doing so improves the performance of the age prediction.

Once these models are trained, their prediction accuracy (output probabilities) are used as the final performance. We will give more details on the architecture of each of these two models in the below parts.

3.1.1 Neural Network model

Traditional Convolutional Neural Networks have two parts. Convolution and pooling layers are used for feature extraction. The latter part is a classifier, where fully connected layers are used. Recently, fully connected layers is replaced with average pooling layers, which reduces large number of parameters required as well as lowering the problem of overfitting also.

Used models:

- Vgg16: VGG 16[29] was proposed by Karen Simonyan and Andrew Zisserman of the Visual Geometry Group Lab of Oxford University in 2014. The input image to the network has a dimension of (224, 224, 3). All convolution layers have (3,3) filter. The first two layers consists of 64 channels and (3,3) padding. After that comes a max pooling layer of stride (2, 2). Then two more convolution layers which consists of 128 channels. This is followed by a max pooling layer of stride (2, 2). After that, there are three convolution

layers with 256 channels. Then, there are a pair of sets of 3 convolution layers as well as a max pooling layer, each consists of 512 channels with same (3,3) padding. The input image is then passed to a stack of two fully connected layers with 4096 channels. Then another fully connected layers with 1000 channels and softmax activation function. This layer is used as output layer.

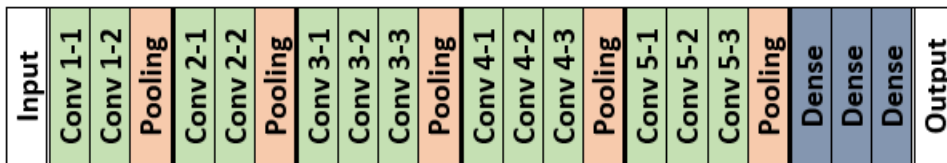


Figure 12: Vgg16 model architecture.

Resnet50[30]: Resnet50 has a input size of (224, 224, 3). Then there are 2 convolution layer, one with (7,7) filter and another with (3,3) filter. Both have 64 channels and maxpool of stride (2,2). After that there are 4 type of convolution set. All convolution set is described in the figure. Resnet consists 3 X convolution set 1, followed by 4 X convolution set 2, followed by 6 X convolution set 3, followed by 3 X convolution set 4. Each set consists 3 convolution layers. After that there is a average pooling layer and a fully connected layer with 1000 channel, which is used as output. Resnet is easy to optimise. It is deeper than VGGnet but can compute with less complexity. An ensembled version of these residual nets achieves 3.57% error on the ImageNet test set.

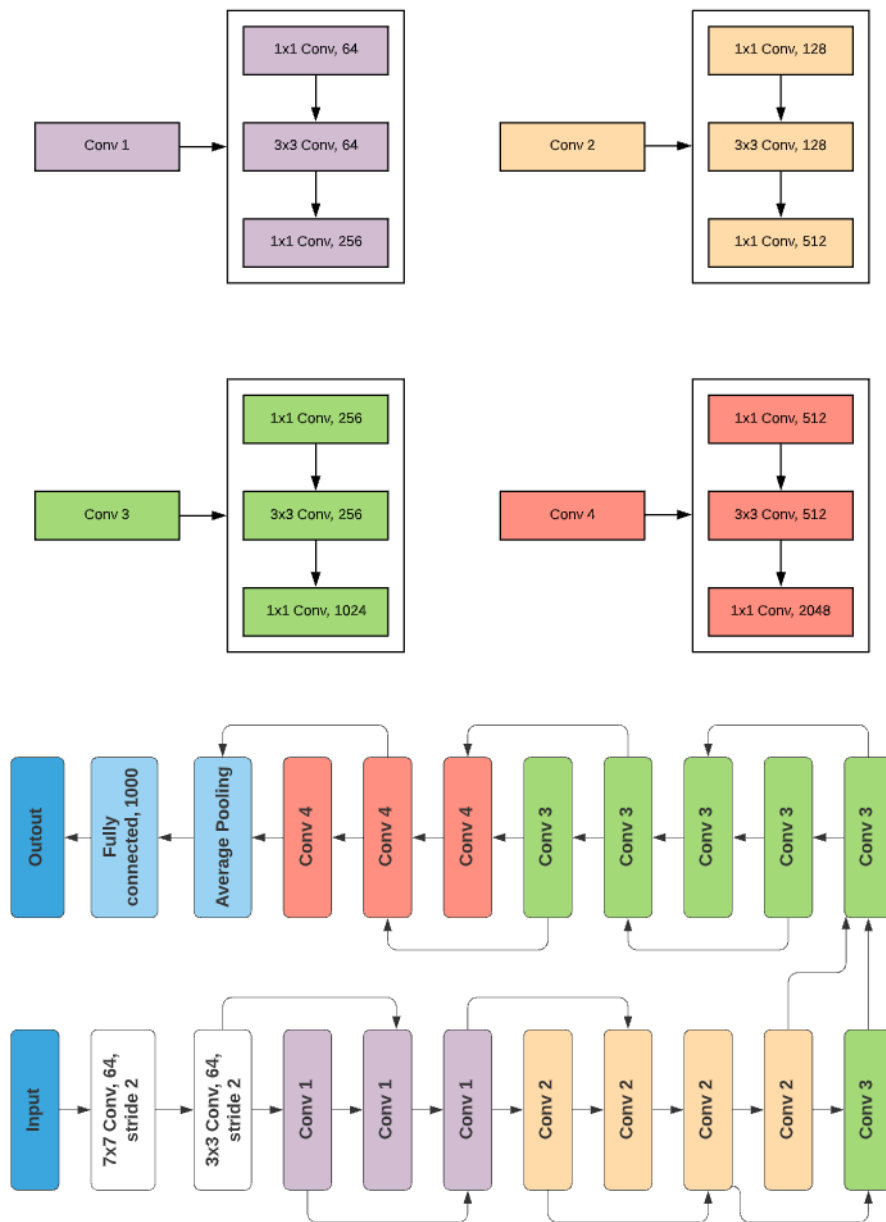


Figure 13: Resnet50 model architecture

Senet50[31]: Senet is almost as same as Resnet with some small changes. Like Resnet50, Senet50 has a input size of (224, 224, 3). Then there are 2 convolution layer, one with (7,7) filter and another with (3,3) filter. Both have 64 channels and maxpool of stride (2,2). After that there are 4 type of convolution set. All convolution set is described in the Resnet50 figure.

Resnet consists 3 X convolution set 1 with 2 fully connected layer of size (16,256), followed by 4 X convolution set 2 with 2 fully connected layer of size (32,512), followed by 6 X convolution set 3 with 2 fully connected layer of size (64,1024), followed by 3 X convolution set 4 with 2 fully connected layer of size (128,2048). Each set consists 3 convolution layers.

3.1.2 Weight

In our method we used VGGFace as our models' weight. VGG-Face has 22 layers and 37 deep units. We used pre trained weight which is trained for face detection, on Wild[32] and Youtube face[33] dataset. We select this weight because it is trained with a large number of facial image to detect face, which could be helpful to extract facial feature. These facial feature is used as crucial information for age estimation.

3.2 Customizing model

We bring some change in previously stated models. Such as- fine tuning, k-fold validation.

3.2.1 Fine tuning

Fine tune is required when a pre trained model is used for different work or we can say, during transfer learning. As the output layer was made for previously trained work. So it is needed to change according to new work. Here our classification model classify age within 8 classes.

Fine tune also consist changing, adding, removing layer of NN models. Such changes are also used to bring change in feature extracting. feature extracting is a crucial point in machine learning. It can bring change in performance of a model by a large margin. In our proposed model we added 4 more layer. First layer is a 'Flatten' layer, it takes matrix as input and turned it in an array. It concatenates the rows of matrix to create the array. It is used as input of next layer.

Next three layers of model are 'Dense' or fully connected layer. This layer mainly used in output layer. The pre built model had dense layer. But we need to add again, because those layers were trained for previously used work. To use them for different work we need to add some dense layer and train them with new dataset. These dense layer predicts the output. For first 3 dense layer 'ReLU' was used as activation function and for last dense layer 'Softmax' function was used.

For output layer 'softmax' is used as activation function. The softmax function is vastly used as activation function in output layer. Softmax function. It helps to locate the index of the largest element of an array. Here, the unit with the largest element of an array is replaced by +1 while all others are replaced by 0.

The used dense layers are:

- Dense(1024, activation='ReLU')
- Dense(1024, activation='ReLU')
- Dense(512, activation='ReLU')
- Dense(8, activation='Softmax')

3.2.2 k-fold cross validation

Cross validation is a method which is used to train a model with a smaller dataset. In this method the train set is divided into k groups or folds. Each fold contains almost same amount of images. The model is also trained for k times. In each iteration 1 fold is used as validation set and other k-1 fold is used as train set. Thus the train is continued till all folds are used as validation set for one time. During the whole process the elements of the fold does not change. This method helps to train a model thoroughly on train set and to achieve high train accuracy.

In our work we set the value of k is 5. So the train set got 80:20 split, which is a vastly used in machine learning. After every fold a performance checking is taken on validation set. This customization gives a big performance boost.

3.2.3 Adam optimization

The Adam optimization algorithm is an optimization technique for gradient descent. It is widely used for deep learning applications in computer vision related work. It can handle sparse gradients on noisy problems. Most optimizers are good for a few particular work, but Adam optimizer shows good performance boost in a wide range of work. Most deep learning libraries use the default parameters recommended by the paper. As we used Keras, the learning rate is 0.001.

3.3 Data set

UTKFace[11] is a large-scale face dataset with over 20,000 face images. It covers a long age span ranged from 0 to 116 years. We created 8 classes for age estimation: ((0 – 2), (4 – 6), (8 – 12), (15 – 20), (25 – 32), (38 – 43), (48 – 53), (60+)). The dataset additionally contains the labels of gender, as well as five races. UTKFace includes large collection of image of different facial expression, illumination, occlusion, pose and resolution. Total size of dataset is near 120 MB. Some features of the dataset:

- consists of more than 20k face images
- provides the correspondingly aligned and cropped faces
- provides 68 points landmarks
- images are labelled by age, gender, and ethnicity

Labels of the dataset image:

- 'age' is an integer from 0 to 116.
- 'gender' is either 0 (male) or 1 (female).
- 'race' is an integer from 0 to 4, denoting White, Black, Asian, Indian, and others.
- 'datetime' i.s showing the date and time an image was collected to UTKFace



Figure 14: UTKFace dataset example.[11]

3.3.1 Data-set Augmentation

The dataset initially found has uneven distribution relative to the classes of age. Which can create bias problem. The biasness can harm the learning process. To solve this issue, we binned the images in some age classes, so that every age classes have almost same number of image. Thus biasness can be avoided. We re categorise age within 8 classes((0 – 2), (4 – 6), (8 – 12), (15 – 20), (25 – 32), (38 – 43), (48 – 53), (60+)) to avoid bias.

4 Model Training and Result Analysis

4.1 Training

We used the network which was previously designed to detect face and modified the structure of it in order to do age detection. Our data set had an age range of 0 to 116. Since the network was designed for classification, we needed to define our age classes. The difference between the numbers of age classes in model also played a role in decreasing the quality of performance. We considered (0-2), (4-6), (8-12), (15-20), (25-32), (38-43), (48-53), (60+) as age classes. We kept difference between age classes to keep good visual difference in them. Having an overall number of 8 classes for age, we trained the network with 5 fold cross validation and batch size was 32. We trained all three fine tuned models(Vgg16, Resnet50, Senet50) each fold for 20 epochs, which is total 100 epochs (5 hours).

4.1.1 Training configuration:

- Models: Vgg16, Resnet50, Senet50
- Weight: VGGFace
- Dataset: UTKFace
- Optimizer: Adam
- Batch size: 32
- Learning rate: 0.001
- Number of fold: 5
- Number of epoch: 20 per fold
- Total epoch: 100

4.1.2 Test Bench

- Processor: Intel(R) Xeon(R)
- CPU clock speed: 2.3 G.Hz
- CPU Core: 2
- Ram: 12/26 GB
- Platform: Python 3 (Google colab)
- GPU: Tesla T4
- VRAM: 16 GB
- GPU Memory Clock: 1.59GHz
- Performance: 8.1 TFLOPS
- Disk space: 120 GB
- Time to train each model: 5 hours

4.2 Result analysis

This chapter presents the detailed description of our evaluation metrics we considered, as well as the results of our experiments in different models. Finally we present a comparative analysis of our proposed method with the results obtained from previous works. Our goal is to get all these results with a relative amount of more cohesive and less complex method.

4.2.1 Evaluation Metrics

We considered several evaluation metrics for our experiments. These are:

$$\text{Accuracy} = \frac{TP + TN}{TP + FP + TN + FN}$$

$$\text{Precision} = \frac{TP}{TP + FP}$$

$$F1score = 2 \times \frac{Recall \times Precision}{Recall + Precision}$$

where TP = True Positives

TN = True Negative

FN = False Negatives

and FP = False Positives

4.2.2 Result

All of our models are trained on UTKFace dataset. Here, we present accuracy, recall, precision and f1 score of our proposed models.

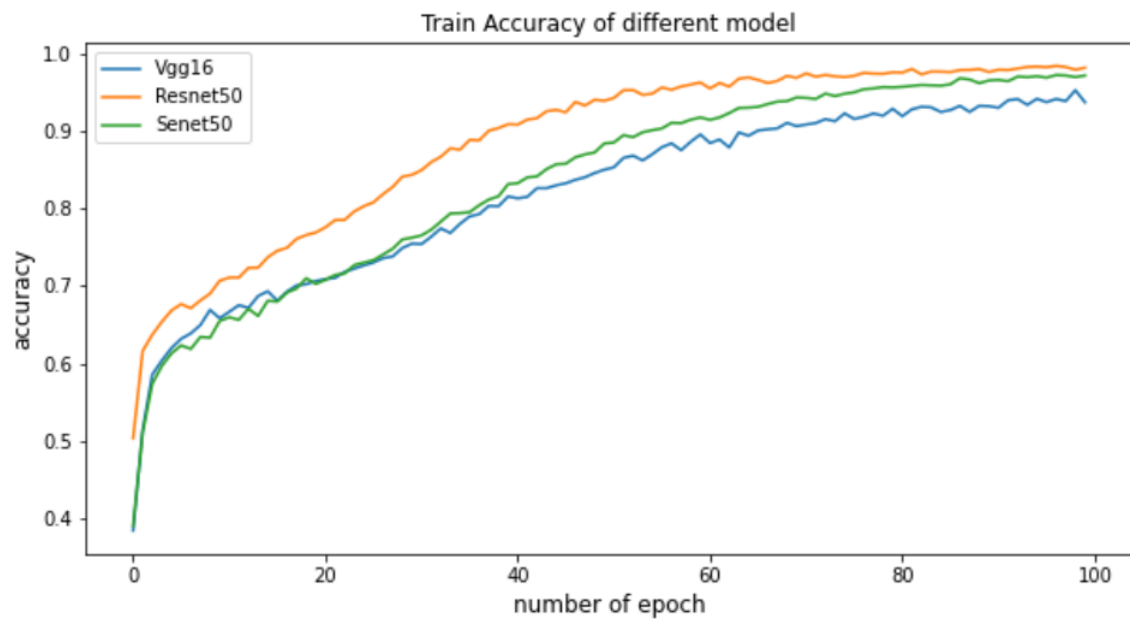


Figure 15: Train accuracy

Here, in Figure 6 we can see that all models training accuracy has higher starting point. We measured training accuracy in every epoch. As, transfer learning is used all models have some initial skill. The slope is also steeper which indicates higher rate of performance improvement. Higher asymptote means faster converging. Without transfer learning performance would increase in slower pace.

Model	fold1	fold2	fold3	fold4	fold5	Average
Vgg16	0.7337	0.8084	0.8344	0.8928	0.9185	0.8376
Resnet50	0.7500	0.8511	0.9017	0.9642	0.9345	0.8803
Senet50	0.5905	0.6696	0.7172	0.9315	0.81250	0.7443

Table 1: Accuracy of each fold.

We measured validation accuracy after every fold. k-fold validation shows some really good performance here. Figure 2 shows validation accuracy of every fold.

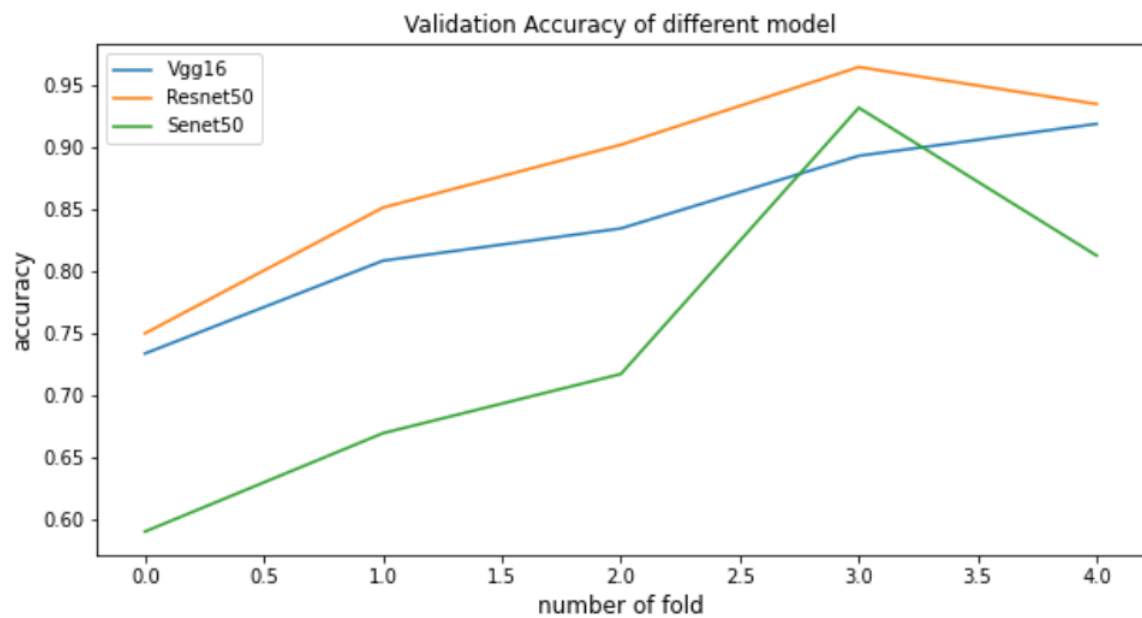


Figure 16: Validation accuracy of models in each fold

Precision means the portion of retrieved documents that are relevant to the wanted data. High precision with high accuracy is a good for accurate estimation. High accuracy with low precision can be result of biasness. In Figure we show the precision of every fold.

Model	fold1	fold2	fold3	fold4	fold5	Average
Vgg16	0.8694	0.8601	0.9069	0.9414	0.9446	0.9045
Resnet50	0.7929	0.8306	0.9167	0.9548	0.9291	0.8848
Senet50	0.7671	0.7849	0.8582	0.9506	0.8715	0.8465

Table 2: Precision of models in each fold.

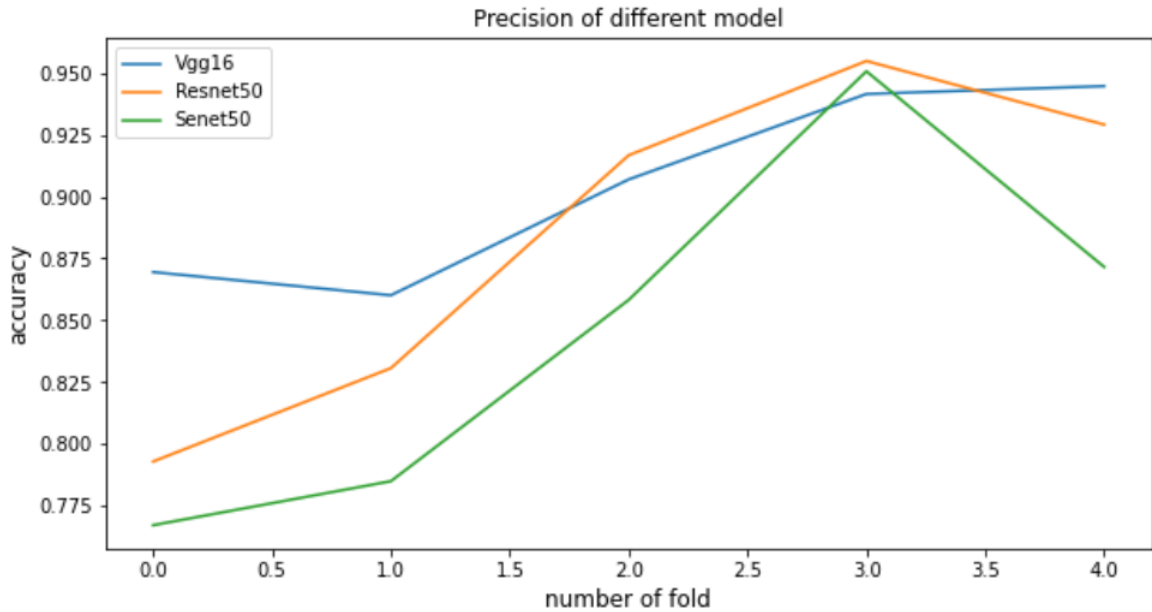


Figure 17: Precision of all model

F1-score gives a overall view of both precision and recall. It is the harmonic mean of precision and recall. It shows a combined score of those. In Figure we show the f1-score of every fold.

Model	fold1	fold2	fold3	fold4	fold5	Average
Vgg16	0.7519	0.8292	0.8533	0.9148	0.9281	0.8554
Resnet50	0.6580	0.7559	0.8866	0.9264	0.9167	0.8287
Senet50	0.6460	0.7177	0.7677	0.9400	0.8326	0.7808

Table 3: f1 score of models in each fold.

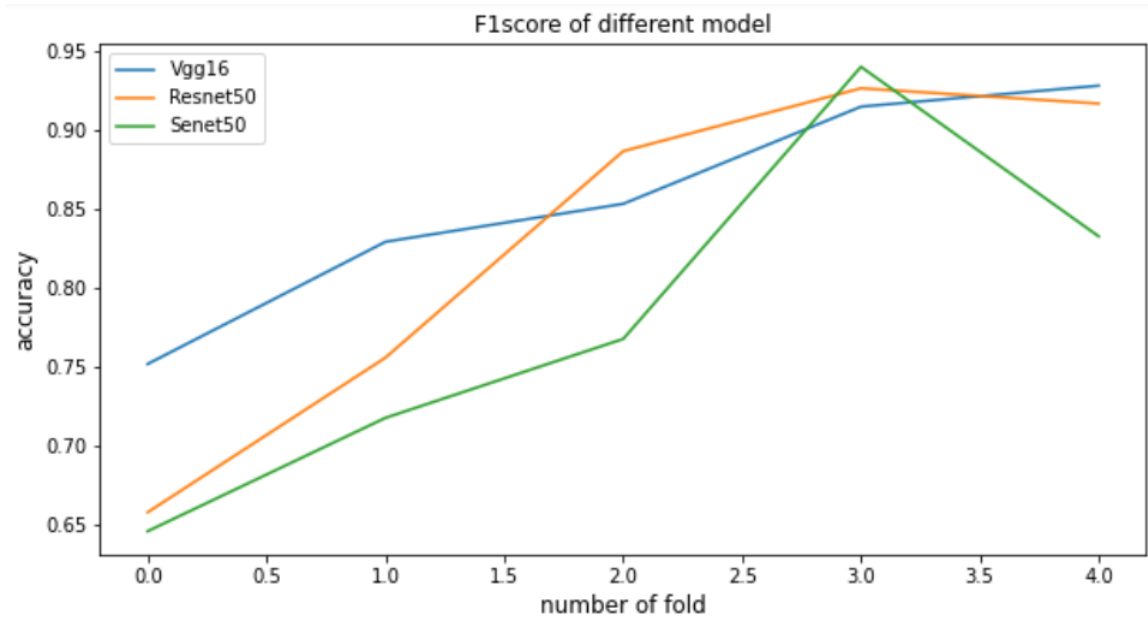


Figure 18: f1 score of all model

Heatmap is a vastly used data visualization technique. It shows magnitude of a occurrences as color in two dimensional graph. The variation in color gives some visual cues about how the occurrences are clustered or varies over parameters. Here are some heatmap of confusion matrix on test set of our proposed model:

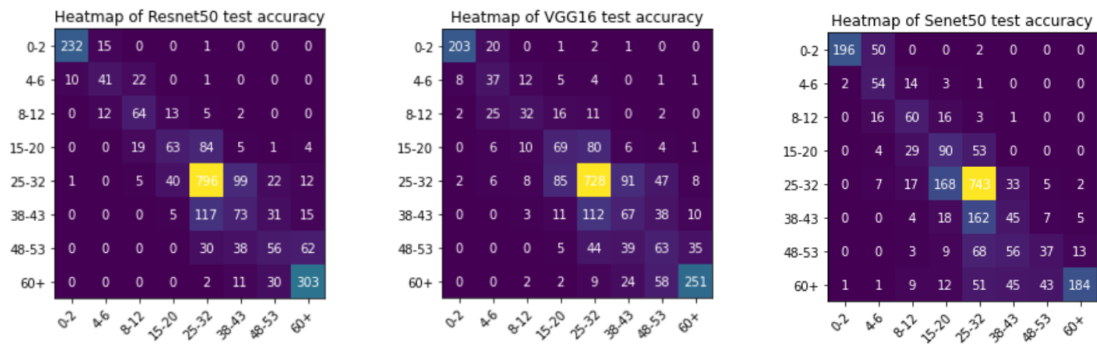


Figure 19: Heatmap of confusion matrix: i. Resnet50, ii. VGG16, iii.Senet50.

4.2.3 Example Outcome

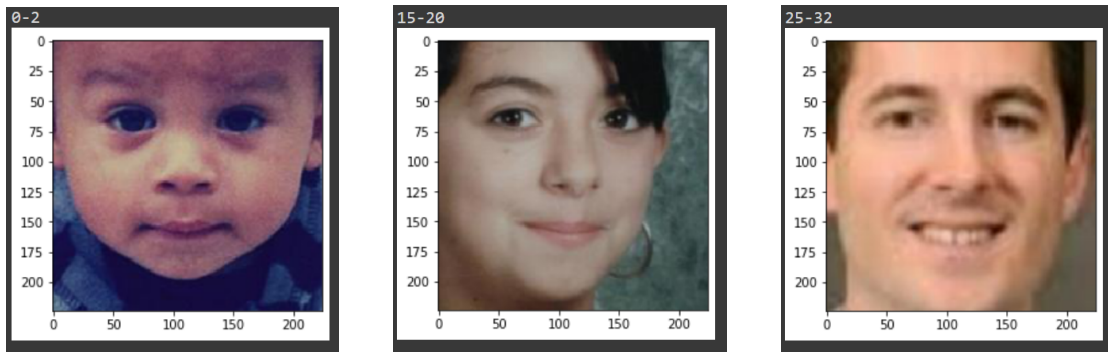


Figure 20: Example output: i. True age=2, ii. True age=12, iii. True age=32.

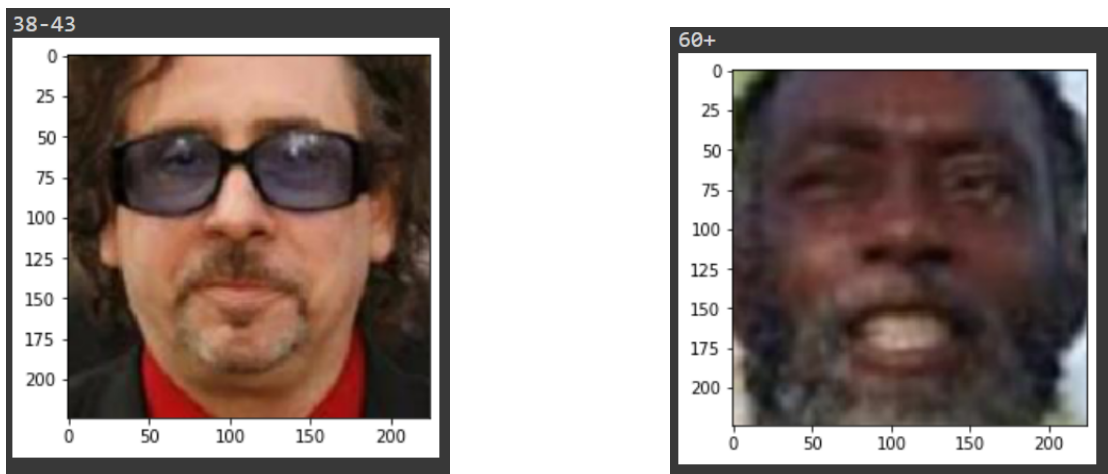


Figure 21: Example output: i. True age=50, ii. True age=70.

Here, in the example outcome section we can see that most of the out come are with in the correct age class. It performs well in spite of having image of different age, race and gender. Even if the age class is not correct, it is neighbour of original age class.

4.3 Weakness

Our new model is not 100% flawless. Here are some weakness from our point of view:

- Our train accuracy was too high, so it created over-fitting problem. Though in later iteration the models overcome the problem.
- Test accuracy is good on the dataset, but it may vary on other datasets.
- Our model is much simple then other model, so achieve same accuracy like other model we needed to train our model for more epochs, which is more time consuming.
- Facial information differs from different ethnicity. In our model there is no special measure is taken to overcome those problem.

4.4 Comparison with previous works

Our another purpose of this thesis is to compare our work with previously published paper. Main target of this work is to show that our proposed model can work as good as other renowned works. Here, we compared our test and validation weighted average accuracy with pre published model.

Model	Accuracy(%)
Facenet[24]	56.9
Finetuned Facanet (FFNet)[24]	64
MTCNN[24]	70.1
AlexNet[26]	69.59
GoogLeNet[26]	67.06
LeNet[26]	62.88
VGG16 (SGD + Adagrad + Adam) with EFI[26]	66.38
AlexNet + GoogLeNet + LeNet[26]	71.74
Facenet(ii)[27]	73
VGG16(DEX)[28]	64
Proposed Resnet50	71.84
Proposed VGG16	65.31
Proposed Senet50	61.96

Table 4: Performance compare with other models

Here, we can see that our proposed model works better then many pre published model. In [24] and [27] both paper used Facenet as one of their base model, which is mainly based on Resnet model. Mainly the use of k-fold validation shows a performance boost in this model. In But our model shows good performance compare to them. VGG16 of [26] is not as good as our best perform model, but that is better than proposed VGG16 model. A reason can be that the proposed method used 8 age classes where compared method trains on 4 age classes, which

may give them the extra accuracy. In[28], in spite of complex pre processing, our additional k-fold validation shows better performance in VGG16 model. All compared methods used classification for age estimation.

5 Conclusion

5.1 Summary

In this study we have explored different age estimation techniques and tried to obtain higher accuracy for age estimation. We used pre-trained CNN models like vgg16, Resnet50, Senet50 and used VGG-Face as weight on face detection which is pre-trained on Wild and YouTube face dataset. We fine tuned the model and used k-fold validation. Then trained our model by UTKFace dataset. It is a much simpler approach than most of the existing models. By comparing our work with the existing models found that our model performed similar if not better in some cases to a lot of the existing models.

5.2 Future Work

In the area of machine learning exploration and research is the only way to improve. There are many ways to continue and change our work in the future.

In our proposed model we used fine tuning by adding a few layers. We did not change the base model. By changing the base model new feature or different feature could be extracted, which would increase performance.

We used pre built model. Problem with these model is that they are made for general use, not for a particular work. So the model might not be optimised.

Here the model is train with one dataset. Training with more data can make our model much more robust.

We should explore different method to find a sweet point between performance and simplicity.

References

- [1] I. Huerta, C. Fernández, C. Segura, J. Hernando, and A. Prati, “A deep analysis on age estimation,” *Pattern Recognition Letters*, vol. 68, pp. 239–249, 2015.
- [2] G. Levi and T. Hassner, “Age and gender classification using convolutional neural networks,” in *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pp. 34–42, 2015.
- [3] K. Luu, K. Seshadri, M. Savvides, T. D. Bui, and C. Y. Suen, “Contourlet appearance model for facial age estimation,” in *2011 international joint conference on biometrics (IJCB)*, pp. 1–8, IEEE, 2011.
- [4] R. Ranjan, S. Zhou, J. Cheng Chen, A. Kumar, A. Alavi, V. M. Patel, and R. Chellappa, “Unconstrained age estimation with deep convolutional neural networks,” in *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pp. 109–117, 2015.
- [5] Y. Zhang, L. Liu, C. Li, *et al.*, “Quantifying facial age by posterior of age comparisons,” *arXiv preprint arXiv:1708.09687*, 2017.
- [6] A. Gunay and V. V. Nabiyev, “Automatic age classification with lbp,” in *2008 23rd International Symposium on Computer and Information Sciences*, pp. 1–4, IEEE, 2008.
- [7] J. Kannala and E. Rahtu, “Bsif: Binarized statistical image features,” in *Proceedings of the 21st international conference on pattern recognition (ICPR2012)*, pp. 1363–1366, IEEE, 2012.
- [8] N. Dalal and B. Triggs, “Histograms of oriented gradients for human detection,” in *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR’05)*, vol. 1, pp. 886–893, Ieee, 2005.
- [9] Devangini, “Local binary patterns (lbp),” Jun 2016.

- [10] J. Brownlee, “A gentle introduction to transfer learning for deep learning,” Sep 2019.
- [11] “Utkface, <https://susanqq.github.io/utkface/>.”
- [12] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” *Advances in neural information processing systems*, vol. 25, pp. 1097–1105, 2012.
- [13] S. Ren, K. He, R. Girshick, and J. Sun, “Faster r-cnn: Towards real-time object detection with region proposal networks,” *arXiv preprint arXiv:1506.01497*, 2015.
- [14] S. Minaee, A. Abdolrashidi, H. Su, M. Bennamoun, and D. Zhang, “Biometric recognition using deep learning: A survey,” *arXiv preprint arXiv:1912.00271*, 2019.
- [15] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, “Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 40, no. 4, pp. 834–848, 2017.
- [16] S. Minaee, Y. Wang, A. Aygar, S. Chung, X. Wang, Y. W. Lui, E. Fieremans, S. Flanagan, and J. Rath, “Mtbi identification from diffusion mr images using bag of adversarial visual features,” *IEEE transactions on medical imaging*, vol. 38, no. 11, pp. 2545–2555, 2019.
- [17] H. K. Ekenel and R. Stiefelhagen, “Why is facial occlusion a challenging problem?,” in *International Conference on Biometrics*, pp. 299–308, Springer, 2009.
- [18] R. Jana, D. Datta, and R. Saha, “Age estimation from face image using wrinkle features,” *Procedia Computer Science*, vol. 46, pp. 1754–1761, 2015.

- [19] K. Ricanek and T. Tesafaye, “Morph: A longitudinal image database of normal adult age-progression,” in *7th International Conference on Automatic Face and Gesture Recognition (FGR06)*, pp. 341–345, IEEE, 2006.
- [20] A. Lanitis, “Evaluating the performance of face-aging algorithms,” in *2008 8th IEEE International Conference on Automatic Face & Gesture Recognition*, pp. 1–6, IEEE, 2008.
- [21] T. Ojala, M. Pietikainen, and T. Maenpaa, “Multiresolution gray-scale and rotation invariant texture classification with local binary patterns,” *IEEE Transactions on pattern analysis and machine intelligence*, vol. 24, no. 7, pp. 971–987, 2002.
- [22] V. Ojansivu and J. Heikkilä, “Blur insensitive texture classification using local phase quantization,” in *International conference on image and signal processing*, pp. 236–243, Springer, 2008.
- [23] A. Hyvärinen, J. Hurri, and P. O. Hoyer, *Natural image statistics: A probabilistic approach to early computational vision.*, vol. 39. Springer Science & Business Media, 2009.
- [24] A. Das, A. Dantcheva, and F. Bremond, “Mitigating bias in gender, age and ethnicity classification: a multi-task convolution neural network approach,” in *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*, pp. 0–0, 2018.
- [25] F. Schroff, D. Kalenichenko, and J. Philbin, “Facenet: A unified embedding for face recognition and clustering,” *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun 2015.
- [26] C.-J. Lin, C.-H. Lin, C.-C. Sun, and S.-H. Wang, “Evolutionary-fuzzy-integral-based convolutional neural networks for facial image classification,” *Electronics*, vol. 8, no. 9, p. 997, 2019.
- [27] “Arg (age race gender) detection using transfer learning based on facenet pretrained model,” 2019.

- [28] R. Rothe, R. Timofte, and L. Van Gool, “Deep expectation of real and apparent age from a single image without facial landmarks,” *International Journal of Computer Vision*, vol. 126, no. 2, pp. 144–157, 2018.
- [29] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” *arXiv preprint arXiv:1409.1556*, 2014.
- [30] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.
- [31] J. Hu, L. Shen, and G. Sun, “Squeeze-and-excitation networks,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 7132–7141, 2018.
- [32] G. B. Huang, M. Mattar, T. Berg, and E. Learned-Miller, “Labeled faces in the wild: A database for studying face recognition in unconstrained environments,” in *Workshop on faces in ‘Real-Life’ Images: detection, alignment, and recognition*, 2008.
- [33] L. Wolf, T. Hassner, and I. Maoz, “Face recognition in unconstrained videos with matched background similarity,” in *CVPR 2011*, pp. 529–534, IEEE, 2011.