

Age Invariant Face Recognition

Prathama V¹, Thippeswamy G²

¹Student, ²Dean and Professor Department of CSE

^{1,2}BMSIT&M, Bengaluru, Karnataka, India

How to cite this paper: Prathama V | Thippeswamy G "Age Invariant Face Recognition" Published in International Journal of Trend in Scientific Research and Development (ijtsrd), ISSN: 2456-6470, Volume-3 | Issue-4, June 2019, pp.971-976, URL: <https://www.ijtsrd.com/papers/ijtsrd23572.pdf>



IJTSRD23572

ABSTRACT

Face recognition across age progression is remains one of the area's most challenging tasks now a days, as the aging process affects both the shape and texture of a face. One possible solution is to apply a probabilistic model to represent a face simultaneously with its identity variable, which is stable through time, and its aging variable, which changes with time.

This paper proposes a deep learning and set-based approach to the face recognition subject to aging. The images for each subject taken at various times are treated as a single set, which is then compared to the sets of images belonging to other subjects. Facial features are extracted using a convolutional neural network characteristic of the deep learning. Our experimental results show that set-based recognition performs better than the singleton-based approach for the both face identification and face verification. We also find that by using set-based recognition, it is easier to recognize older subjects from younger ones rather than younger subjects from older ones.

Copyright © 2019 by author(s) and International Journal of Trend in Scientific Research and Development Journal. This is an Open Access article distributed under the terms of the Creative Commons Attribution License (CC BY 4.0) (<http://creativecommons.org/licenses/by/4.0>)



Keywords: CNN- Convolution Neural network, AIFR-Age invariant face recognition, Polling, Flattening, Keras

INTRODUCTION

Facial aging is a complex process that affects both the shape and texture (e.g., skin tone or wrinkles) of a face. This aging process also appears in different manifestations in different age groups. While facial aging is mostly represented by facial growth in younger age groups (e.g., 18 years old), it is mostly represented by relatively large texture changes and minor shape changes (e.g., due to change of weight or stiffness of skin) in older age groups (e.g., >18). Therefore, an age correction scheme needs to be able to compensate for both types of aging processes

Age invariant face recognition refers to an automatic face recognition technique that is able to achieve temporal invariance over a wide range of the subject's age. In real time face recognition applications (e.g. passport renewal, law enforcement and biometric authentication), the AIFR system should be able to recognize faces despite of the appearance variations due to aging. Analyzing effects of aging on the performance of AIFR systems is one of the important areas in facial aging.

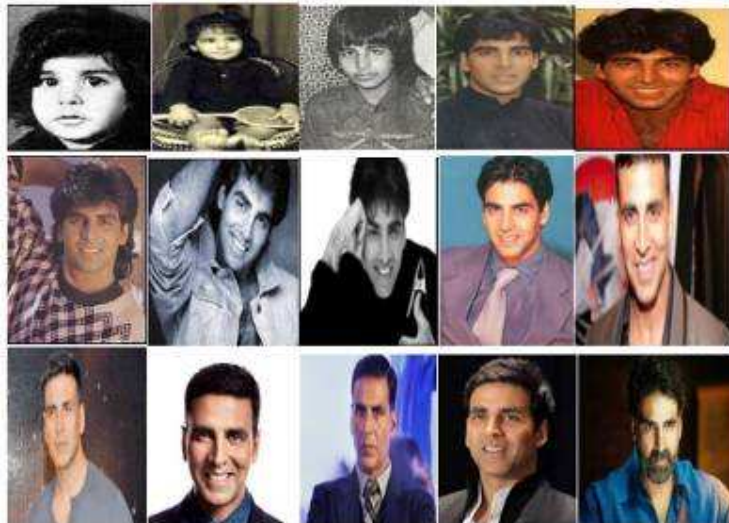


Fig:1-Dataset 1 of different age



Fig:2-Dataset 2 of different age

CNN image classifications takes an input image, process it and classify it under certain categories (Eg., Dog, Cat, Tiger, Lion). Computers sees an input image as array of pixels and it depends on the image resolution. Based on the image resolution, it will see $h \times w \times d$ (h = Height, w = Width, d = Dimension). Technically, deep learning CNN models to train and test, each input image will pass it through a series of convolution layers with filters (Kernalns), Pooling, fully connected layers (FC) and apply Softmax function to classify an object with probabilistic values between 0 and 1.

Literature Survey

N Ramanathan and R Chellappa [7] presented a Bayesian age difference classifier that identifies the age Separation between a pair of face images of an individual. While the method presented in this paper is suitable to handle age progression in adult face images, since it does not account for shape variations in faces, it is not effective for handling age progression in face images of children.

H Ling and S Soatto et al [6] proposed a robust face descriptor, the gradient orientation pyramid, for face verification tasks across ages. Compared to previously used descriptors such as image intensity, the new descriptor is more robust and performs well on face images with large age differences. In addition, the pyramid technique enables the descriptor to capture hierarchical facial information.

N R Syambas and U H Purwanto [1] focused on development of image pre- processing factors like contrast, Brightness, Sharpness in the Recognition System for improved recognition accuracy.

G Mahalingam and C Kambhamettu [8] presented a graph based image representation and an aging model constructed using GMM for each individual to model their age variations mainly in shape and texture. A two stage approach for recognition has been proposed in which a simple deterministic algorithm that exploits the topology of the graphs is proposed for efficient graph matching between the probe image and the gallery image.

J S Nayak and Indiramma M [4] proposed a novel self-PCA based approach in order to consider distinctiveness of the effects of aging of a person for age invariant face recognition. The region around the eyes is used as the input feature instead of the entire face as it is more stable part of the face.

J S Nayak and Nagarathna N et al [5] proposed this self-PCA based face recognition method to consider the aging effects by constructing the subspace at the individual level.

Z Li and U Park et al [3] proposed a discriminative model to address face matching in the presence of age variation. The scale invariant feature transform (SIFT) and multi-scale local binary patterns (MLBP) serve as the local descriptors. Since both SIFT-based local features and MLBP-based local features span a high- dimensional feature space, to avoid the over fitting problem, we develop an algorithm, called multi-feature discriminant analysis (MFDA) to process these two local feature spaces in a unified framework.

Design

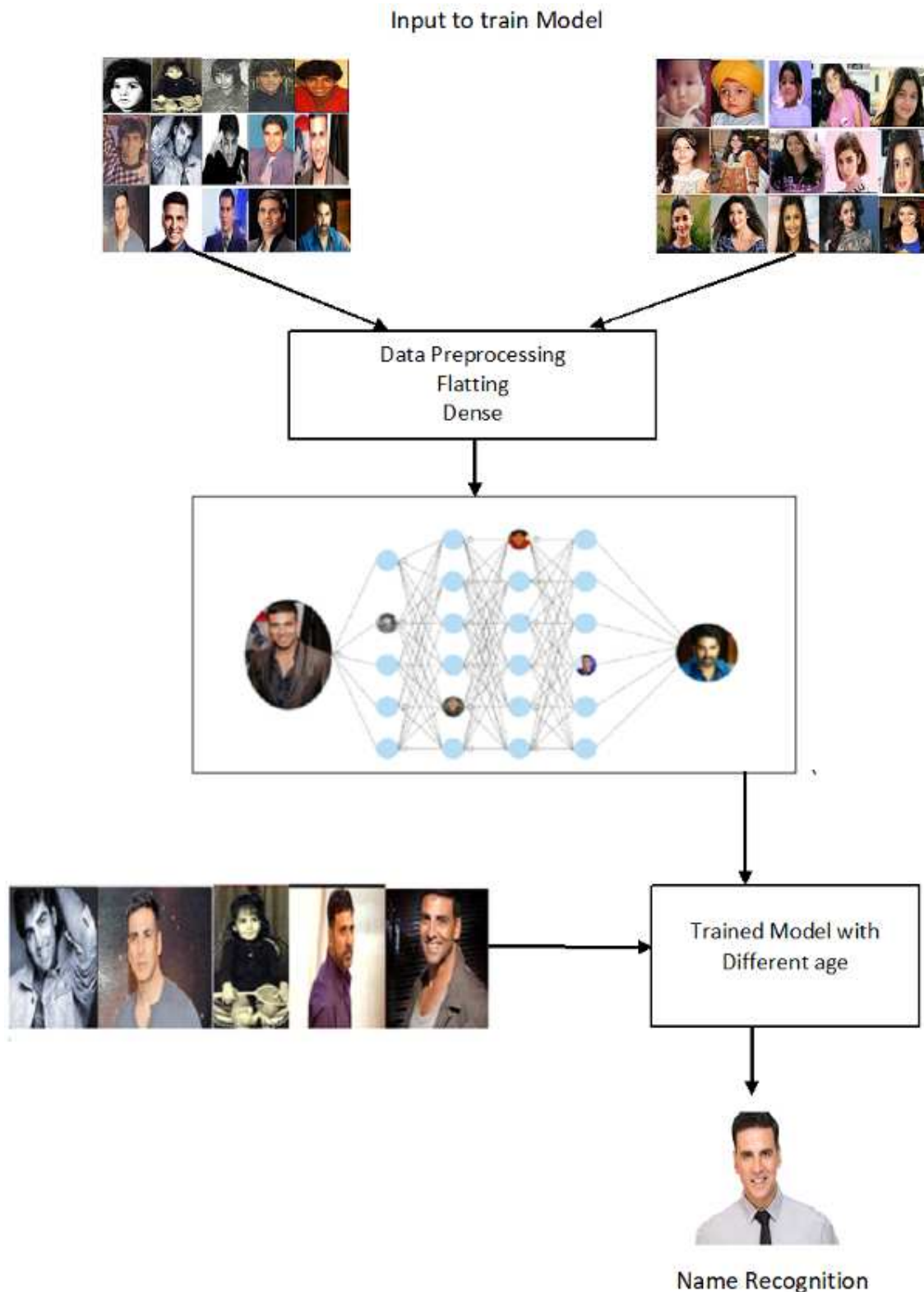
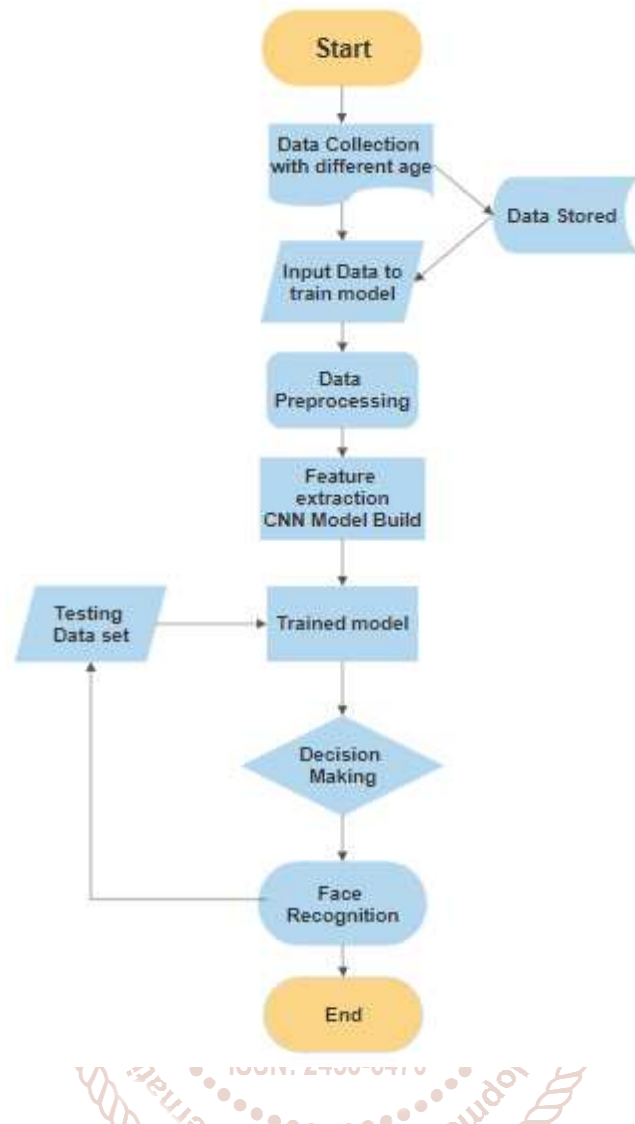


Fig: 3 Designs

Tools and Technologies

1. Anaconda — Anaconda is a free and open source distribution of the Python and R programming languages for data science and machine learning related applications, that aims to simplify package management and deployment. You can download it from the link below according to your system
2. Spyder — Spyder is an open source cross-platform IDE for scientific programming in the Python language. It comes installed with anaconda. If not, install it using anaconda navigator.
3. Tensorflow — TensorFlow is an open-source software library for dataflow programming across a range of tasks.
4. Keras — Keras is an open source neural network library written in Python. Activate Tensorflow env and install keras using 'pip install keras'.
5. CNN — Convolution Neural network, a class of deep, feed-forward artificial neural networks.

Flow Chart**Implimentation**

- Collecting the Dataset
- Importing Libraries and Splitting the Dataset
- Building the CNN
- Full Connection
- Data Augmentation
- Training our Network
- Testing

1. Collection of data

Primary Data Collection Methods: The primary data are the first-hand data, collected by the researcher for the first time and is original in nature. The researcher collects the fresh data when the research problem is unique, and no related research work is done by any other person. The results of the research are more accurate when the data is collected directly by the researcher but however it is costly and time-consuming.

Secondary Data Collection Methods: When the data is collected by someone else for his research work and has already passed through the statistical analysis is called the secondary data. Thus, the secondary data is the second-hand data which is readily available from the other sources. One of the advantages of using the secondary data is that it is less expensive and at the same time easily available, but however the authenticity of the findings can be questioned.

2. Importing Libraries and Splitting the Dataset

Install following packages and import them.

```

import tensorflow
from keras.models import Sequential
from keras.layers import Convolution2D
from keras.layers import MaxPooling2D
from keras.layers import Flatten
from keras.layers import Dense
  
```

Splitting Data into testing and training

The **training dataset** (also called training set, learning set, or AI training data) is the initial dataset used to train an algorithm to understand how to apply technologies such as neural networks, to learn and produce complex results. It includes both input data and the corresponding expected output. The purpose of the training dataset is to provide your algorithm with “ground truth” data.

The **test dataset**, however, is used to assess how well your algorithm was trained with the training dataset. You can't simply reuse the training dataset in the testing stage because the algorithm will already “know” the expected output, which defeats the purpose of testing the algorithm.

3. Building the CNN

Convolutional Neural Network is a class of deep neural network that is used for Computer Vision or analyzing visual imagery. This is most important step for our network. It consists of three parts -

1. Convolution
2. Polling
3. Flattening

The primary purpose of Convolution is to extract features from the input image. Convolution preserves the spatial relationship between pixels by learning image features using small squares of input data.

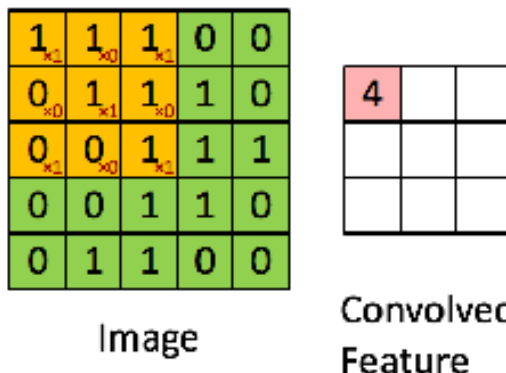
Since every image can be considered as a matrix of pixel values. Consider a 5 x 5 image whose pixel values are only 0 and 1 (note that for a grayscale image, pixel values range from 0 to 255, the green matrix below is a special case where pixel values are only 0 and 1):

| | | | | |
|---|---|---|---|---|
| 1 | 1 | 1 | 0 | 0 |
| 0 | 1 | 1 | 1 | 0 |
| 0 | 0 | 1 | 1 | 1 |
| 0 | 0 | 1 | 1 | 0 |
| 0 | 1 | 1 | 0 | 0 |

Also, consider another 3 x 3 matrix as shown below:

| | | |
|---|---|---|
| 1 | 0 | 1 |
| 0 | 1 | 0 |
| 1 | 0 | 1 |

Then, the Convolution of the 5 x 5 image and the 3 x 3 matrix can be computed as shown in the animation in fig below:



The obtained matrix is also known as the feature map. An additional operation called ReLU is used after every Convolution operation. The next step is of pooling. Pooling (also called subsampling or downsampling) reduces the dimensionality of each feature map but retains the most important information. In case of Max Pooling, we define a spatial neighborhood (for example, a 2×2 window) and take the largest element from the rectified feature map within that window. Instead of taking the largest element we could also take the average (Average Pooling) or sum of all elements in that window. In practice, Max Pooling has been shown to work better. After pooling comes flattening. Here the matrix is converted into a linear array so that to input it into the nodes of our neural network.

4. Full Connection

Full connection is connecting our convolutional network to a neural network and then compiling our network. Here we have made 2 layer neural network with a sigmoid function as an activation function for the last layer as we need to find the probability of the object being different age person.

5. Data Augmentation

While training your data, you need a lot of data to train upon. Suppose we have a limited number of images for our network. What to do now?? You don't need to hunt for novel new images that can be added to your dataset. Why? Because, neural networks aren't smart to begin with. For instance, a poorly trained neural network would think that these three tennis balls shown below, are distinct, unique images. So, to get more data, we just need to make minor alterations to our existing dataset. Minor changes such as flips or translations or rotations. Our neural network would think these are distinct images anyway.

Data augmentation is a way we can reduce overfitting on models, where we increase the amount of training data using information only in our training data. The field of data augmentation is not new, and in fact, various data augmentation techniques have been applied to specific problems.

6. Training our Network

So, we completed all the steps of construction and its time to train our model. If you are training with a good video card with enough RAM (like an Nvidia GeForce GTX 980 Ti or better), this will be done in less than an hour. If you are training with a normal cpu, it might take a lot longer. With increasing number of epochs, the accuracy will increase.

7. Testing

Now lets test model by giving the following test case

- Giving untrained image and testing the accuracy.
- Giving trained image and testing .
- Giving different image not related to trained and testing.
- Giving multiple images in a single picture and testing.

The test_image holds the image that needs to be tested on the CNN. Once we have the test image, we will prepare the image to be sent into the model by converting its resolution to 64x64 as the model only expects that resolution. Then we are using predict() method on our classifier object to get the prediction. As the prediction will be in a binary form, we will be receiving either a 1 or 0, which will represent a person with his different age.

Conclusion

AIFR(Age Invariant Face recognition) has several helpful and sensible applications. for example, it may be applied to find missing kids when years or checking whether or not the identical person has been issued multiple government documents in several ages. However, it still remains a difficult drawback in plenty of applications of face recognition systems. the foremost challenge of AIFR is generally attributed to the nice changes in face look caused by aging method over time. We have a tendency to propose a sturdy age-invariant deep face recognition framework. To the simplest of our data, it's the primary work to indicate the effectiveness of deep CNNs in AIFR and attain the simplest results up to now.