

Expression Invariant Face Recognition Using Convolutional Neural Networks

Minu Correya¹, Dr. Thippeswamy G²

M. Tech Student, Department of CSE, BMS IT&M, Bangalore, India¹

Dean, BMS IT&M, Bangalore, India²

Abstract: For the past few decades, one of the most extensively studied topics under computer vision is face recognition. This technology has many applications in real life. However, the performance of a face recognition system suffers with variation in lighting, pose and facial expression. Out of these major concerns faced by the recognition system this paper mainly focuses on the impact of expression variation in recognition systems. Facial expression changes have a negative effect on the accuracy of recognition of a face recognition system. Hence this paper proposes an expression invariant face recognition technique using convolutional neural network.

Keywords: Face Recognition, Expression Invariant, Convolutional Neural Networks, Deep Learning

I. INTRODUCTION

The research on face recognition was started in the late 1970s. Since 1990, this area of research has become more active in the field of computer vision and pattern recognition [1]. Three concerns in the field of face recognition are: illumination variation, the variation in pose and more specifically the impact of change in expression which is the main focus of this paper. Facial expression adversely effects the performance of a face recognition system. To overcome this challenge of expression variation, expression invariant technology is introduced. Expression-invariant face recognition deals with expression which changes human appearance, which eliminate a broad range of inconsistency by considering dynamics of expression in recognition. There are various methods that have been implemented for expression invariant face recognition. However, Deep Learning (DL) is the one that is currently mostly used, and providing the best results, especially the Convolutional Neural Networks (CNN). A Convolutional Neural Network (CNN) is a type of artificial neural network that uses a machine learning unit algorithm called perceptron, to analyse data. CNNs can be applied to image processing, natural language processing and other kinds of cognitive tasks [2]. Like artificial neural networks, a convolutional neural network has an input layer, various hidden layers and an output layer as shown in Fig 1. These layers use a mathematical model which passes on the results to successive layers.

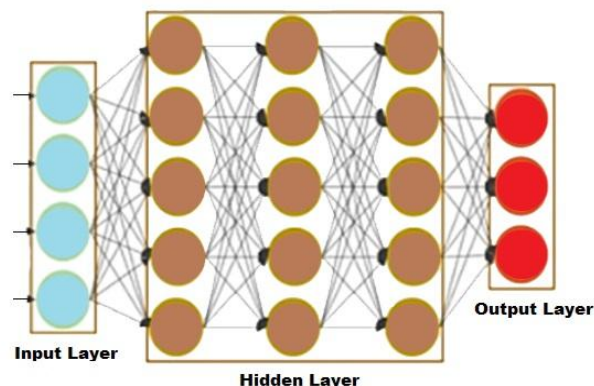


Fig 1. The Layers of Convolutional Neural Networks

II. CONVOLUTIONAL NEURAL NETWORK

A convolutional neural network just like neural networks consists of neurons that have learnable weights and biases. However, in convolutional neural networks the neurons in one layer connect to only a small part of the neurons in the following layer and not to all of them. The main aim of the convolutional neural network is to minimize the size of the image into a structure which is easier to process, without losing features which are important in order to get an accurate prediction.



The network has an input layer, an output layer and hidden layers which contains greater than one convolution layer. The input layer accepts the pixels of the image as input and stores it in the form of an array. The feature extraction is carried out by the hidden layers which is done by performing certain calculations and manipulations. And finally, the output layer or the fully connected layer is responsible for identifying the object in the image [3].

The convolutional neural network hidden layers are mainly composed of convolutional layers, nonlinearity layers (activation function) and pooling layers.

A. Convolution Layer:

In the convolution layer a convolutional filter or kernel is applied on the input data to obtain a feature map. The convolution operation is performed by moving the kernel over the input. Each time performing a matrix multiplication the sum of which is stored in the feature map. This area where the operation takes place is known as the receptive field. The convolution layer consists of a set of filters which performs multiple convolutions on an input resulting in a distinct feature map. These feature maps are then piled up to form the final result of the convolution layer. Fig 2. Shows the convolution operation where the filter on the top left corner (pink shade) is applied over the image to obtain a feature map shown on the right where the result of the operation is 4.

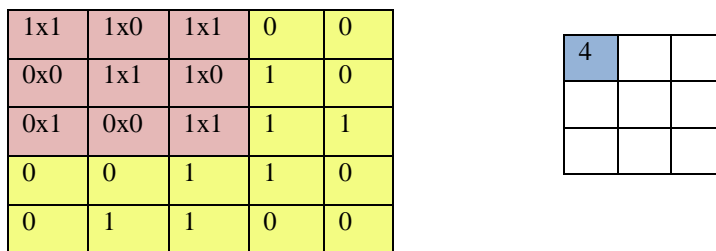


Fig 2. Convolution operation

B. Nonlinearity Layer:

The activation functions main purpose is to introduce nonlinearity in the network since the output of the convolution layer is linear. The generated output is adjusted or saturated using nonlinearity [4]. The activation function performs mathematical operations on single numbers. The commonly used activation functions are shown below –

- Sigmoid function – The range of its output value is in between 0 and 1, while it takes a real-valued function as input. Mostly used in neural networks but does not work well with convolutional neural networks.

$$f(x) = \frac{1}{1 + e^{-x}}$$

- Hyperbolic tangent function – The range of its output value is in between -1 and 1. The tanh is preferred when compared to sigmoid since the outputs are zero-centred.

$$\tanh(x) = 2f(2x) - 1$$

- ReLU activation function – The rectified linear unit is currently the most widely used activation function. It is a linear function and has its threshold at zero. When compared to the sigmoid and tanh function the ReLU learns much quicker. It always returns a positive value otherwise it returns zero.

$$\text{rect}(x) = \max(x, 0)$$

C. Pooling Layer:

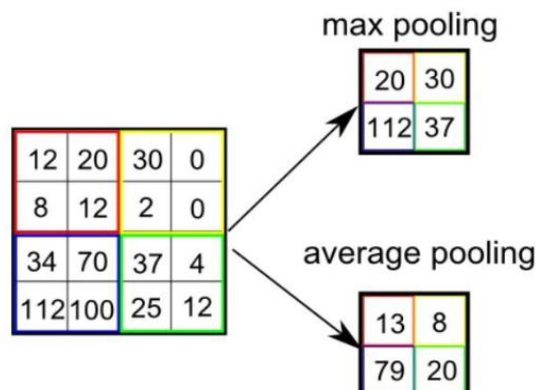


Fig.3. Types of pooling



The pooling layer reduces the dimensionality of the feature maps hence reducing the number of parameters and also the computation time. It decreases the training time and tackles the problem of overfitting. It also helps to effectively train the model by extracting the dominant features. There are two types of pooling namely the max pooling and the average pooling. In average pooling the result is the average of the values in the pooling window whereas in max pooling the result is the maximum value of all the values in the pooling window. Max pooling is commonly used as it performs better when compared to average pooling. Noise is reduced considerably using max pooling. Fig 3 below shows the types of pooling techniques.

III. METHODOLOGY AND IMPLEMENTATION

This section describes the proposed methodology and implementation for an expression invariant face recognition using Convolutional Neural Networks. Since the convolutional neural networks has performed well in the field of face recognition it is applied here to obtain a face recognition system which is invariant to facial expression changes. This is implemented with the help of Python and TensorFlow which are one of the many frameworks available for deep learning [5]. Initially in order to train the model, it is necessary to collect the accurate database of images. Here a database created, consists of various folders, where each folder holds a set of images of a single person. Fig 4 shows an example of a possible database of a single person with varying expressions. Then once the database is collected it is divided into two, the first set is that of training database images and the other set is that of testing database images. There are many readily available sets of databases from various sites on the net.



Fig 4. An example of the possible dataset

After the database is collected and all the required libraries are imported. The CNN is built with the help of a library called Keras which is a neural network library that runs on top of TensorFlow. Keras consists of a number of implementations of the commonly used building blocks of neural networks. Using Keras the different layers of CNN are created. The type of activation function used is the ReLU function and the sigmoid function is used for classification. Once the model is created it is trained using backpropagation with gradient descent. The model is trained for the specified number of epochs. The accuracy increases with the increase in epochs. The trained model can be saved for later use and there is no need to train the model again for recognition classification. Finally, the model can be tested with the test database available and the results can be verified.

IV. CONCLUSION

This paper proposes an approach to expression invariant face recognition system using convolution neural networks. The results obtained successfully recognises individuals even with varying facial expressions. The accuracy obtained is higher than traditional methods that have been used. Hence, convolutional neural networks can be used to successfully implement expression invariant face recognition.

ACKNOWLEDGEMENT

I would like to express my sincere gratitude for the support and guidance rendered by all the faculty members of Computer Science and Engineering Department, BMS Institute of Technology and Management, Bangalore.

REFERENCES

- [1]. Marryam Murtaza, Muhammad Sharif, Mudassar Raza, and Jamal Hussain Shah, "Analysis of Face Recognition under Varying Facial Expression: A Survey", The International Arab Journal of Information Technology, Vol. 10, No. 4, July 2013.
- [2]. https://en.wikipedia.org/wiki/Convolutional_neural_network
- [3]. Saad Albawi, Tareq Abed Mohammed and Saad Al-zawi, "Understanding of a Convolutional Neural Network", IEEE, 2017.
- [4]. Neena Aloysius and Geetha M, "A Review on Deep Convolutional Neural Networks", International Conference on Communication and Signal Processing, April 6-8, 2017, India.
- [5]. Liping Yuan, Zhiyi Qu, Yufeng Zhao, Hongshuai Zhang, Qing Nian, "A Convolutional Neural Network based on TensorFlow for Face Recognition", Lanzhou, China, IEEE, 2017.