International Journal of Engineering & Technology, 7 (4.41) (2018) 79-82

International Journal of Engineering & Technology

Website: www.sciencepubco.com/index.php/IJET

Research paper

Handwriting Analysis Using Convolutional Neural Networks

Tushar Sadana[1*], Monika Jain[2], Rahul Saxena[3], Aashis Kumar[4], Vidhyanshu Jain[5] , Saurabh Gupta[6]

[1,2,3,4,5,6] *Manipal University Jaipur, Rajasthan, INDIA*
**Corresponding author E-mail: tusharsadana@gmail.com*

Abstract

Convolution is the technique to blend or overlap two or more functions. This technique when provided to artificial neural networks, works together to learn the features of different categories of objects and detects them based on its features instead of the shape and edges. This helps to detect the objects even in unusual positions. Since, features of an object remains constant, CNN provides high efficiency significantly better than traditional cascade methods. CNN networks follow convolution, max pooling, flattening. These process combines preprocess the image for training and then the image is transferred to artificial neural networks.

Keywords: *Convolution; Detection; Flattening; Identification*

1. Introduction

How fascinating it is to see when babies learn to speak, learn to walk, learn to identify their parents and other things. In this advancing world of technology, can our machines also perform with the same intelligence? The answer is yes. Convolutional Neural Networks are designed in such a way, where the machines acquire the ability to identify different features of the objects and classify them in different categories. These

advances to identify the minute features of the face and detect different faces and identify different personalities. How CNN works and How the features are extracted plus the classification would be answered in this article.
A set or piece of information which is relevant to solve any computational task, is called a feature. In image detection or processing features is commonly referred to as a continuous or repeating pattern which can be observed in a single classified category.

Features are the information extracted from the image of the repeating pattern in the
form of numerical value that are difficult to understand and corelate by humans. The images are interpreted as a matrix (2D grayscale and 3D RGB). This matrix contains
a value between 0 to 1, portraying the grayscale or the colour value of that pixel. These matrices can be as large as 4000 x 4000 and as small as 28 x 28. Features can be extracted by first detecting them and then changing it to a 1D array from a 2D or 3D array[4].

Features are used for detection and identification. Detection is to predict the existence of an object while identification is to identify the object.

Originally, Image processing was not that intelligent and cascade methods where used to identify the objects in an image. The algorithm was based on the detection of borders and then detecting the shapes and further comparing with the objects like the shape [7]. This was done by applying different operations like threshold and edge detection to find the border and edges and detect the shapes. The limitation was if the position of the object changes or the photograph might be tilted the shape comparison algorithm do not identifies the object properly.

In CNN, the basic operation is to detect the features in the image and then identification takes place. It is better because image may be tilted or the object may change its place, but the feature tends to remain same, even in the upside-down photograph..

2. Convolution

Convolution is a mathematical operation on two functions to generate the third function. It is an integral function which expresses the amount of overlap of one function over the other by generating the third one, simply by blending one function to other [1].

Convolution of two function f and g is given by

$$[f*g](t) = \int f(\tau)g(t-\tau)\,d\tau.$$

Where [f*g] (t) denotes the convolution of function f and g.

Convolution is more often taken from negative infinity to positive infinity.

In words, convolution is an operation defined as the product of two functions, out of which one is a function of τ and one a function of $(t - \tau)$.

For Example:

If we take f(t) = sin(t) and g(t) = cos(t)

The convolution of f and g is
$[f * g](t) = \int_0^t \sin(t-\tau)\cos\tau \, d\tau$

$[f * g](t) = \int_0^t \cos\tau(\sin t \cos\tau - \sin\tau \cos t)d\tau$

$[f * g](t) = \int_0^t \sin t \cos 2\tau \, d\tau - \int_0^t \cos t \sin\tau \cos\tau \, d\tau$

Since, t is a constant
$[f * g](t) = \sin t \int_0^t \cos 2\tau \, d\tau - \cos t \int_0^t \sin\tau \cos\tau \, d\tau$

Finally resulting to:
$[f * g](t) = 1/2 \, t \sin t$

τ is the variable which varies from 0 to t. τ is used to identify the different areas of the functions. t- τ is used to discover the values from the other side while overlapping or blending both the functions [2].

3. Neural Network

Neural networks are the replication of the biological neurons present in our brains in computer machinery which helps us to compute even the smallest to the largest of the computation required.
Everything what humans do, or feel is the result of this basic structure of the human brain called the neurons. Human brain consists of 100 billion neurons, just like a single ant could never build an anthill a single neuron can't think or feel or remember. These neurons are of no use if used individually, but their performance significantly increases when used in numbers. The neuron's power is a result of its connections to other neurons. Each neuron is connected to as many as a thousand of its neighbours. These trillions of connections provide the plane filled upon which the complex activity of the brain takes place.
In the same manner, different nodes are created which are the input nodes forming the input layer of the network. Further this layer is connected to different layers between the output layer. The layers in between are generally called hidden layers. Each node in each layer contains a number which varies from 0 to 1. This value is called the activation. This portrays the probability that the network predicts. In grayscale images, this 0 to 1 can show the grayscale value of a pixel on the image grid. Then comes the output layer, where the final answer is predicted in the form of activation value. The connection between two nodes is denoted by weights. It is the value that refers to the strength or the amplitude of the connection between the nodes. The higher the weight the more effective node is formed.

3.1. Training the Neural Networks

Neural Networks are trained on a dataset provided where the training values are predicted and then compared with the original by applying the cost function. There after the feedback is given back to the network which adjusts the weights and the cost function accordingly. This is called the Reinforcement Learning.
Output value or the result $\hat{y}$ equals to

$$\hat{y} = \varphi\left(\sum_{i=1}^{m} W_i X_i\right)$$

Cost Function is basically a measure how good is a neural network did with respect to its training set and the expected output. It is a single value, not a vector. The most basic example of cost function is

$C = \frac{1}{2}(\hat{y} - y)^2$

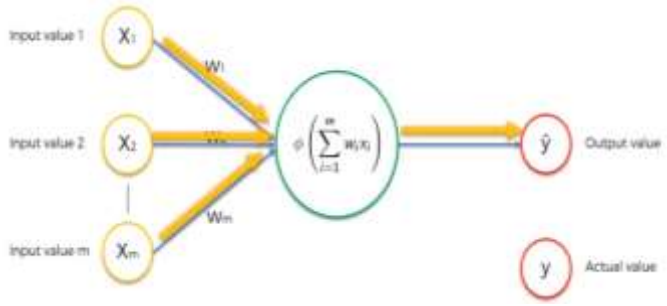

Fig. 1: Showing the Structure of a neural network. Yellow is input layer. Green is the Hidden Layers. Red is the Output Layer.

The weights are adjusted in such a way that the value of the cost function is possibly the lowest.

3.2. Methodology

As we know, an image is matrix (2D or 3D) containing the saturation value of each pixel. Grayscale images are 2D while RGB are 3D.

Convolution is applied on the first step. The feature we want to detect is taken in a different matrix called feature detector. Since, the feature is to be present in image, the feature detector is much smaller in dimension than the image. Then the feature detector is passed all over the image and a feature map is detected. A feature map is matrix which contains the value of similarity of the feature in the image. The feature map contains the value of intensified pixel matched with the feature in the part of that image. This is step is repeated for different features all over the image and thousands of feature maps are created [1].

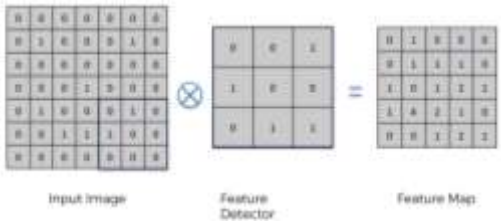

Fig. 2: Input image is a simple image of a smiley, where the intensified portion is portrayed by 1 and other are 0. Feature detector is the left part of the smile. Feature Map is created by comparing the feature detector over the image and storing the number of intensified pixel matched.

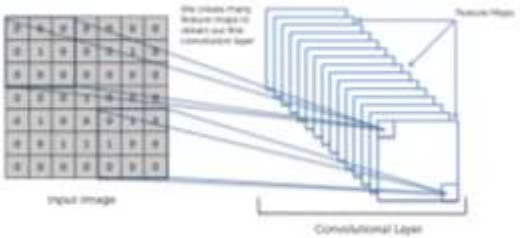

Fig. 3: Formation of all the Convolution layers.

3.3. Max Pooling

The purpose of pooling is to gain spatial invariance by reducing the resolution of the feature maps. Pooling is applied to reduce the size of the feature maps and store just the relevant and most useful value of the feature and discarding other data/values. These stored values are those values which has the highest probability of being present in the same object. A photograph of an object may differ in angles, position of the objects, lighting etc. but these features tends to remain constant, which provides the base for identification [3].
Spatial invariance is achieved by applying the max pooling. A size smaller than the feature detector is taken and is again passed over

the image and, in this case, we store the maximum value present in the matrix, and create a pooled feature map. Due to this, we lose all the irrelevant data and we stick to the most important and constant features of the image.

Fig. 4: Max pooling is applied.

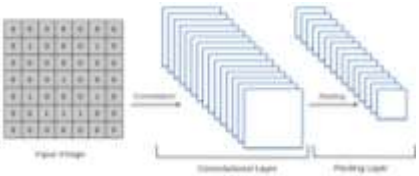

Fig. 5: All the layers formed till now.

3.4. Flattening and Full Connection

Flattening is a simple step to convert all the pooled feature maps into a single dimensional array which behaves as the input layer of the artificial neural network.

Fig. 6: Pooled Feature Map converting into flat layer

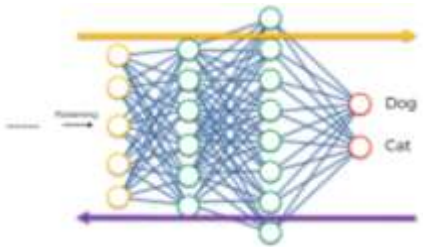

Fig. 7: The flattened layer is passed into an ANN.

The input is given to the neural network and it passes through different hidden layers. The output is decided based on the last hidden layer. Certain nodes in the last hidden layer may correspond to a dog and others may correspond to a cat. During the Reinforced learning, the output layers keeps in mind, which node supports which output. Thus, the values in these nodes decide the output. If the dog favourable nodes have higher value than the cat favouring nodes, the output is shown as dot.
The value of each node varies between 0 to 1, where each node may define any feature and the value shows the probability of the existence of that feature in the image[4]. Refer to already defined symbols, equations, theorems by using the cross reference number (Example: As pointed in (1) the…).

4. Analyzing and Recognizing the Symbols

4.1. The structure of the model

A model was created using Keras Library in Python 3.6.* where Convolutional Neural Networks were used. There were total of five layers in the model. The first input layer, then the three hidden layers and then the last output layer. Different Activation Functions were tried and optimized, but the "relu" and the softmax functions in combination worked the best. These functions work best into classification of different things.
After that max pooling and flattening was also applied, explained above. Also, a dropout rate of 20% was used, i.e. 20% of the neurons tends to remain inactive while learning. This avoids the overfitting of the training data.
The images were of the size of 28x28, which were then later passed through feature mapping of 3x3 and pooling of 2x2 and then converted to a flattened array of (23*23) i.e. 529, which is the number of inputs in the first layer.

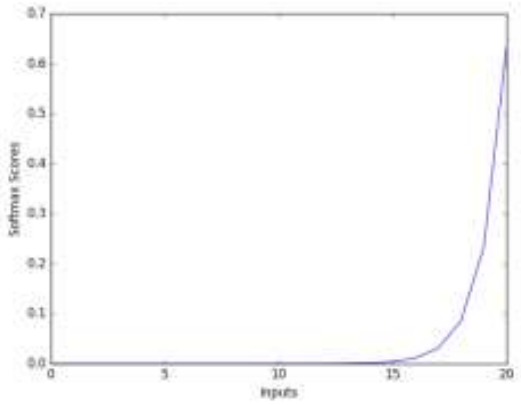

Fig. 8: Graph of Softmax Function

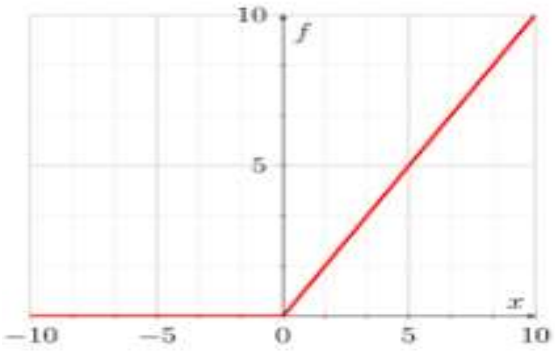

Fig. 9: Graph of Relu Function

4.2. The structure of the dataset

A Different datasets were combined to train the model. MNIST dataset [8] was used with the other random datasets from internet. They were kept in different folders labeling their identities. All the images were made of equal size i.e. of 28 x 28 by a python script (not explained in this paper).
The data was then fitted into the model and about 56 symbols were identified in the
labels with 28756 photographs in total by the model.

5. Mathematical Approach

5.1. Convolution

The initial size of the images was 28 x 28. The size of the kernel was 3x3. From the
formula:

$$O = \left| \frac{I + p - k}{s} \right| + 1$$

Where O is the output size, l is the input size, p is the padding and k is the convolved feature size. By applying the same formula, the Output size after the convolution came out to be 26 x 26, where k was 3, l was 28, p was 0 and s was 1.

This layer was applied twice, thus the final output size after 2-Convolution layer was
24x24.

5.1. Max Pooling

The current size of the image is 24 x 24 which is called the feature map. The size of the pooling map is 2x2. Therefore, according to the formula:

$$O=\left|\frac{l-k}{s}\right|+1$$

Where O is the output size, l is the input size and k is the kernel size. By applying the same formula, the Output size after the convolution came out to be 23 x 23, where k was 2, l was 24, and stride was 1.

In this, the maximum value in the kernel was taken and saved in pooling map. Thereafter the 2D Matrix of 23 x 23 was converted to a single array of values, which was further input into ANN.

6. Results

The model created could give about 98.75% of accuracy to detect a symbol correctly, provided the images where tilted, handwritten or ill shaped at times.
Shape of the images was reduced to 28 x 28. Feature Detector was 3 x3 while pooling was done on 2 x 2. Convolution and pooling was performed four times on the images, while four hidden layers were added in the artificial neural network, full connection layer. Training was performed on k – cross fold method with a batch size of 32.
Later on single images were also given, out of the test and train dataset to the model and the following results were obtained.

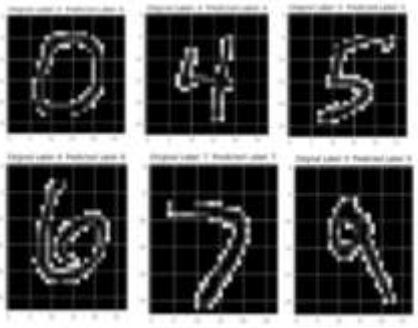

Fig. 10: Individual prediction of symbols

7. Conclusion

CNN has proved to be efficient and accurate. CNN could recognize different symbols in different shape, way of writing, angle and positioning. with an accuracy of 98.75%. A training dataset of 28756 different symbols with labels were provided to the model out of which 20% were kept as the testing data.

Further, more and more symbols can be added to the training dataset which will allow the machine to detect even more symbols in a photograph and become more accurate. Since, these computations were done on a low-grade machine, images were resized to 28x28 and a feature map of 3x3 was taken. If high end computation power machine is provided over the internet, then larger images can be taken for training and thus we can obtain even better results.

References

[1] Jianxin Wu, 2017, Introduction to Convolutional Neural Networks
[2] C.-C. Jay Kuo, 2016, Understanding Convolutional Neural Networks with A Mathematical Model
[3] Dominik Scherer et al., 2010, Evaluation of Pooling Operations in Convolu-tional Architectures for Object Recognition
[4] www.superdatascience.com/deep-learning
[5] Monika Jain et al./ Elixir Digital Processing 88 (2015) 36377-36380 “Development of image processing software for online measurements at streak camera system in Indus-1 synchrotron radiation source”.
[6] Upadhyay, J., Garg, Akash Deep, Ojha, Avanish, Tyagi, Y., Sharma, M.L., Puntambekar, T.A., Navathe, C.P., Vora, H.S., & Jain, Monika (2015). Measurement of longitudinal electron beam parameters using indigenously developed streak camera system at Indus-1 synchrotron radiation source. India: Bhabha Atomic Research Centre.
[7] Monika Jain, Rahul Saxena “Parallelization of Video Summarization over Multi-Core Processors” February 2018, International Journal of Pure and Applied Mathematics 118(9)
[8] http://yann.lecun.com/exdb/mnist/