Signature Recognition using Convolutional Neural Networks

Jash Patel(B18CSE040)
Heman Vaghasiya(B18EE059)
Preet Shah(B18EE036)
Kushal Shah(B18EE047)
Pratik Goyal(B18BB024)

Abstract

Recognition is regarded as a fundamental attribute of human beings. Signatures are used in everyday life to sign millions of important documents and sanction thousands of monetary transfers. The main aim of our project is to identify counterfeit signatures using an existing dataset of signatures. Initially, the signature is pre-processed and it is then fed to the convolutional neural networks. Using the below-shown feature extraction steps and some other over normal processing provides us with a highly efficient layout to be fed. After training the neural network with some specimens, we can proceed with signature recognition.

1. Introduction

This is a new field of research interest in domains of image processing, deep learning, and neural networks. Signatures are a special type of handwriting featuring intrapersonal variation and interpersonal differences. Thus signatures are considered as images and not a collection of alphabets. Each signature can be categorized differently on the basis of unique features it possesses. The main objective is to bifurcate authentic and inauthentic signatures to ensure transparency.

2. CNN based Signature Recognition

2.1 Preprocessing

Preprocessing is an important aspect of image recognition. Image pre-processing techniques have an important role in increasing the accuracy and efficiency of the post-processing application.

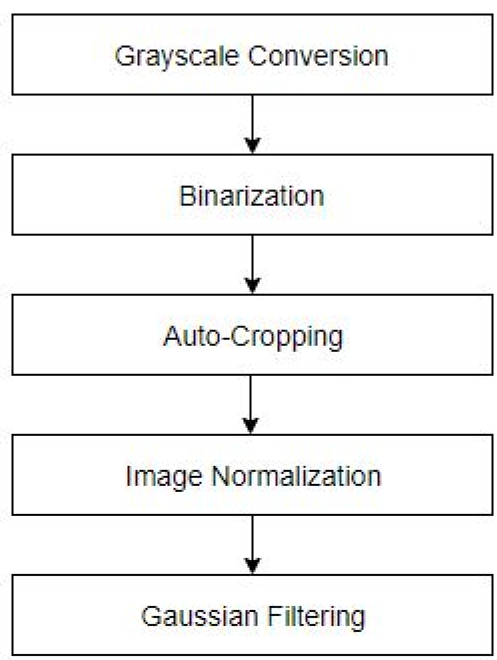


Fig 2.1 Signature pre-processing steps

It is very useful to enhance low-quality images to produce better outputs. Many pre-processing techniques have been devised so far however an important issue is to maintain a balance between editing image (eg.noise removal) and data loss as rapid pre-processing may result in loss of useful information which may affect the later stages of the system.

2.1.1 Grayscale conversion

The first step in the given approach is to convert the signature to grayscale form, i.e. shades of gray on a scale of 0 to 255. Working in this form is more useful because it converts the 3-dimensional RGB image to a 2-dimensional grayscale image.

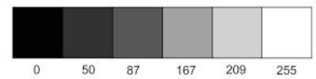


Fig 2.2 Grayscale strip

2.1.2 Binarization

The next step is to convert the grayscale image to binary form, i.e. black(0) and white(1) pixels. Binarization works by finding a threshold value in the histogram- a value that effectively divides the histogram into two parts, where each part represents one of the two objects(the object and the background) It further reduces the 2-dimensional gray-scale image to 2 bits representation of the image.

Table 2.1 Radon transform processing time comparison

Type	Coloured(3D)	Grayscale(2D)	Binary
Time	6.2340	2.3440	2.0780

2.1.3 Auto-Cropping

Using the auto-cropping approach, the region of interest(ROI) is determined in this stage. This leads to a smooth and precise segmentation of the signature. This will lead to the reduction of the area of processing. Normally two types of cropping techniques are used: a) Manual cropping b) Auto-cropping. Manual cropping is achieved using the Matlab function(imcrop) which may cause incorrect cropping rectangle. It is also a time-consuming method. Instead, using a self-made, appropriate and effective auto-cropping algorithm, precise ROI can be determined without any chances of error.

2.1.4 Image Normalization

Changing the range of pixel intensity values is called normalization. It is used to bring the image or other types of a signal into a range that is more familiar. It transforms an N-Dimensional Grayscale image with intensity values in the range (min1, max1), into a new image with intensity values in the range (min2, max2).

2.1.5 Gaussian filtering

To blur the image and to reduce the noise a Gaussian filter is used. It uses a Gaussian function to blur the image. Basically, the result is a convolution of the image and a gaussian function. The formula of a Gaussian function (Fig 2.3) where x is the distance from the origin

$$G(x,y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}$$

Fig 2.3 Gaussian Function

in the horizontal axis, y is the distance from the origin in the vertical axis, and σ is the standard deviation of the Gaussian distribution.

2.2 Feature Extraction

Features can be divided into two types namely global and local features, where global features describe the image as a whole whereas on the contrary local features are confined to a limited portion of the signature like a limited grid. Some examples of the global features include the dimensions of the image like width and length of the given image whereas local features include the specifications for a small grid. Here is a list of features that have been used in the system:

2.2.1 Width to height ratio

2.2.2 High-pressure region percentage

2.2.3 Projection mean

2.2.4 Horizontal projection's standard deviation

2.2.5 Vertical projection's standard deviation

2.2.6 Centre of gravity

2.2.7 No. of Edges and Crosses

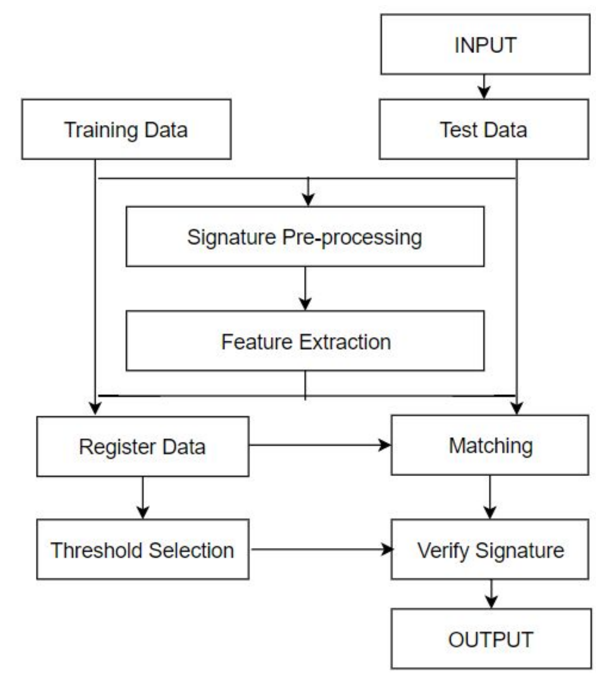
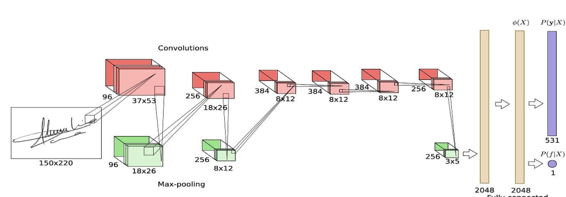


Fig 2.4 Flow Algorithm

2.3 Convolutional Neural Networks(CNNs)

CNNs are a set of algorithms, modelled after the human brain. They take an input image, processes it and classify it under certain categories i.e. fake and genuine signature.



[Fig 2.5 Convolutional Neural Network]

2.3.1 Types of layers:

- **Input layer:** It is used for holding the raw input of an image
- **Convolutional layer:** It computes the dot product between all filters and image patch and gives out the output volume.

- **Activation function layer:** It applies the activation function to the output of the convolutional layer. Eg: Sigmoid function is used as an Activation function layer.
- **Pool layer:** It reduces the size of the volume to make the computation faster.
- **Fully-Connected layer:** This layer takes input from the previous layer and computes the class scores and outputs t5the 1-D array of size equal to the number of classes.

3. Results and Observations

Our first task was to input a raw image, convert it into a grayscale image, which is then converted into a binarized image and cropped by a self-made algorithm. Below is the real-time output from the system which uses a signature from the given database:

Fig 2.5 Original RGB image

Fig 2.6 Grayscale Image

Fig 2.7 Binarized Image

Fig 2.8 Cropped and Filtered Image

Then we are to train a CNN to recognize whether individual signatures are forged or genuine, having seen examples of both forged and genuine versions of that same person's signature during training. Below are the forged signatures of the given original images:

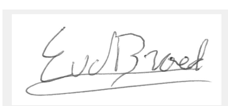


Fig 2.9 Original RGB image

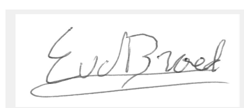


Fig 2.10 Grayscale Image

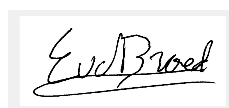


Fig 2.11 Binarized Image

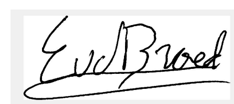


Fig 2.12 Cropped and Filtered Image

Table 3.1 Feature Comparison

Approach	Real Signature 1	Real Signature 2	Forged Signature
Width to Height Ratio	3.4481	3.7059	2.4843
High-Pressure Region per cent	0.1857	0.2085	0.1139
Projection Mean	0.7943	0.7915	0.8861
Horizontal Projection's Deviation	0.0966	0.1516	0.0689
Vertical Projection's Deviation	0.0943	0.1436	0.0590
Centre of Gravity	(0.5061, 0.5063)	(0.5073 , 0.5077)	(0.4994 , 0.4985)
Area Percent of Edges and Crosses	0.0182	0.0275	0.0188

4. Conclusions and Limitations

We experimented with various variations of signatures. The system showed that by using Convolutional Neural Networks one can verify signatures by training them with forged and genuine signatures of the same people whose signatures are observed at the time of the test.

There are several constraints in the phase of acquiring data. The length of the signature proved to be a constraint. An increase in the length of the signature will lead to an increase in behavioural data. This will make it difficult for the system to identify unique and regular data points. We were constrained by the fact that it is difficult to find good publically available signature datasets. We look forward to the following flow or direction for future work. Our offline signature recognition system, since its applicability and ease of use in real-time, is more in comparison to other online signature recognition systems. E.g. On cheques, offline signature verification will be more functional and practical than the online verification system. Moreover, online signature verification methods require some special hardware to capture the dynamic features in real-time, which the offline verification methods do not need to.

References

[1] Bassam Al-Mahadeen and Mokhled S. Altarawneh, "Signature Region Of Interest using Auto cropping, "Umm Al-Qura University, Jordan, April 2010. [Online]. Available: https://www.researchgate.net/publication/43225363

[2] Edureka, Bengaluru, Karnataka, India. Convolutional Neural Network (CNN) | Convolutional Neural Networks With TensorFlow. (Sep. 25, 2017). Accessed: Sep.20,2019. [Online Video]. Available: https://www.youtube.com/watch?v=umGJ30-15_A

[3] Vinayak Ashok Bharadi and Hemant B. Khekre," Off-line Signature Recognition Systems," Finolex Institute of Management and Technology, Narsee Monjee Institute of Management Studies, India, February 2010. [Online]. Available: https://www.researchgate.net/publication/43807907

[4] Gabe Alvarez, Blue Sheffer, and Morgan Bryant," Offline Signature Verification with Convolutional Neural Networks," Stanford University, California, USA, 2015. [Online]. Available: http://cs231n.stanford.edu/reports/2016/pdfs/276_Report.pdf

[5] The Mathworks, Inc. "Functions" Mathworks.com. https://in.mathworks.com/help/referencelist.html?type=function. (Accessed Sep. 20,2019).

[6] Divyansh Rai, "Handwritten Signatures, Genuine and forged signature examples," India, Nov. 2019. [Online]. Available: https://www.kaggle.com/divyanshrai/handwritten-signatures