A PROJECT REPORT

ON

"Analysis of Signature Verification System"

BACHELOR OF ENGINEERING

Under

DIBRUGARH UNIVERSITY

Submitted by:

PANKAJ SAHA	**(CS-26/16)**
HIMJYOTI DAS	**(CS-48/16)**
AKHIRUL ALOM MONDAL	**(CS-33/16)**
RAJAN SAHU	**(CS-37/16)**

Under the guidance of

Mr. Himanish Shekhar Das

Assistant Professor (CSE Dept.)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

JORHAT ENGINEERING COLLEGE

JORHAT-785007, ASSAM

Session 2018-2019

ACKNOWLEDGEMENT

Our heartiest gratitude, appreciation and honour is credited to Mr. Himanish Shekhar Das, Assistant Professor, Department of Computer Science and Engineering, Jorhat Engineering College, Jorhat for his benevolent guidance, apt suggestions, unstinted help, constant check-up and evaluation of the project which have helped us in completing this dissertation report, without his guiding lamp the project could not have come into reality.

We remain in extreme debt to Dr. Rupam Baruah , Head of the Department, Department of Computer Science and Engineering, Jorhat Engineering College, for providing us the wonderful practical approach of learning and co-operative achievement. We remain grateful for providing us with funds, facilities, advice for the completion of the project.

Department of Computer Science and Engineering

JORHAT ENGINEERING COLLEGE

JORHAT-785007

CERTIFICATE

This is to certify that the project report entitled "**Analysis of Signature Verification System"** is hereby accorded our approval as a study carried out and presented for acceptance in partial fulfilment for the Mini Project in the 6th Semester and does not necessary endorse or accept every statement made opinion expressed or conclusions drawn as recorded in the report. It only signifies the acceptance of the project report for a purpose for which it is submitted.

Dr. Rupam Baruah

Head of the Department

Computer Science and Engineering Department

Jorhat Engineering College, Jorhat

CONTENTS:

Introduction

Signature verification has been a challenging and interesting problem for biometric researchers from many years in the recent past. Signatures can be broadly divided into two categories
1) Offline Signatures and

2) Online Signatures.

Offline signatures are those that take place on paper and analysis of them can be carried out only when information of a signature on paper is somehow extracted through image processing and stored digitally. This information can be termed as features and are used primarily to distinguish between two signatures. In a way, it would be safe to say that the features provide a unique identity to a signature.

Online Signatures on the other hand are signatures that take place on an electronic device that are capable of recording the movement of signature at a fixed interval of time, digitally. Therefore, x-y coordinates of the user's signatures along with attributes like pressure, time and pen up/down are also acquired. It is because of these vast range of dynamic features that online signature verification system usually achieves a better accuracy than an offline system.

This paper performs online signature on touch sensitive devices. They represented online signature as a discriminative feature vector derived from histograms of attributes which is computed in linear time. A resulting template signature is generated for every user which is used to validate a user as genuine or forgery. The dataset used is MCYT-100 and SUSIG data sets. The features used are coordinates of x, y, pressure and timestamps. The steps involved in their methodology are pre-processing, histogram feature extraction, template generator, matcher and verification. Validation of their method is done using acceptance false rate and rejection false rate.[1]
This paper implements field-programmable gate arrays of an embedded system for online signature verification. Steps involved are Signature pre-processing, Sigma-Lognormal Parameter Extraction, Signature Reconstruction, Generation of duplicate signatures and Verification of signatures. Their methodology consists of three stages. In pre-processing stage, removal of noise and normalizing x and y coordinates of signature. Later dynamic time warping algorithm is used to align this processed signature with its template previously stored in a database. Next features are extracted and Gaussian mixture model is applied, it gives degree of similarity of signatures. The algorithm was tested using a public database of 100 users, obtaining high recognition rates for both genuine and forgery signatures which gave good results. [2]
Usually online signature verification requires 5 -10 genuine signatures of a person. But this implemented online signature using single signature of user. Their methodology is generating duplicate signatures from one single signature of user. Duplicates are generated using sigma lognormal decomposition. Two methods are presented to create human-like duplicated signatures: the first varies the strokes lognormal parameters whereas the second modifies their virtual target points. Their results proved that their accuracy is matching with models that their 5 signatures for verification.[3]
In this paper a feature extraction approach based on Legendre series representation of the time functions associated with the signatures is proposed. A consistency factor is proposed to

quantify the discriminative power of different combinations of the time functions. Initial step is feature extraction, in this step normalization, extended functions like x-velocity, y-velocity, log curvature radius, consistency computation are calculated. They have used classifiers like support vector machine and random forests. They have explored different combinations of feature vectors. Dataset used in this paper is SigComp2011, it consists of Chinese and western dataset. [4]

This paper implemented on offline signature verification by using image processing, geometric feature extraction, neural network technique, resilient back propagation and radical basic function. Steps involved in their methodology are training stage and testing stage. In training stage there are four major steps those are retrieval of a signature image from a database, image pre-processing feature extraction and neural network training. In the next stage testing is done to verify the model designed. The accuracy of their system is 86.25. Their database consists of 2106 signatures containing 936 genuine and 1170 forgeries. [5]

Applications

Signature verification has been and is used in many applications ranging from governmental use to commercial level to forensic applications. Some of them are as follows:-

1) **Security for Commercial Transactions*:*** Nowadays signature verification used for commercial use. It can be used for authentication on ATMs, for package delivery companies. The internationally recognized courier service UPS has been using signature verification for many years now for personnel identification.
2) **Secure computer system authentication:** Logging on to PCs can be done with a combination of signature verification system and fingerprint identification system to achieve a higher level security in a sensitive area. We can also use a combination of password and signature verification system. This would allow the users to have a higher level of security and confidentiality for their clients and protection of their work.
3) **Cheque Authentication:** Signatures have been using for decades for cheque authentication in banking environment. But even experts on forgeries can make mistakes while identifying a signature. In general, Off-line signature verification can be used for cheque authentication in commercial environment.
4) **Forensic Applications:** Signature verification techniques have been used for cheque fraud and forensic applications.

Advantages

In the point of view of adaption in the market place, signature verification presents three likely advantages over other biometrics techniques.

- First nowadays it is a socially accepted verification method already in use in banks and credit card transaction.
- Second, it is useful for most of the new generation of portable computers and personal digital assistants (PDAs) use handwriting as the main input channel.
- Third, a signature may be changed by the user. Similarly to a password while it is not possible to change finger prints iris or retina patterns.

Therefore, automatic signature verification has the unique possibility of becoming the method of choice for identification in many types of electronic transactions, not only electronics but also for other industries.

Motivation

Biometric recognition and verification systems have existed for over a century and are extremely crucial to every industry to keep their information and data secure.

There are many biometric verification systems like finger print scanning, face recognition, iris scanning, and voice recognition etc.

But cost of deploying for some systems is extremely high and some of the systems are not portable.

In-order to find a middle ground between efficiency, effectiveness in mass quantity and portability, we take a deeper look into signature verification system which can be deployed to produce fruitful results.

Problem Statement :

The problem is to verify whether given signature is genuine or forgery of a particular user using feature extraction, matching and machine learning algorithms**.**

Literature Survey

Sl.No	Title/Broad Area and year of publication	Details about Frameworks/Algorithms used	Details about Tools, Datasets	Summary of the research outcome
1.	Online Signature Verification -2014	An online signature is represented with a discriminative feature vector derived from attributes of several histograms that can be computed in linear time. The resulting signature template is compact and is verified with other signature for genuine or forgery.	The algorithm was first tested on the well-known MCYT-100 and SUSIG data sets	The results demonstrate the problem of within-user variation of signatures across multiple sessions and the effectiveness of cross session training strategies to alleviate these problems
2.	Embedded System for Biometric Online Signature Verification -2014	1. Field-programmable gate arrays (FPU) 2. Vector floating-point unit (VFPU) 3. DTW 4. Gaussian mixture model (GMM)	MCYT-100 data set of signatures	Acceptance rate And rejection rate are calculated.
3.	Dynamic Signature Verification System Based on One Real Signature-2016	1. Dynamic time wrapping. 2. Hidden Markov model	MCYT-100 data set of signatures	Experimental results suggest that our system, with a single reference signature, is capable of achieving a similar performance to standard verifiers trained with up to five signature specimens
4.	Dynamic Signature Verification System Based on One Lengedre polynomials Based feature extraction for online signature verification .Consistency analysis of feature combinations-2014	1. Legendre polynomials 2. Support Vector Machines 3. Random Forests.	SigComp2011 dataset, it consists of Chinese and western dataset.	Results show that there is a good correlation between the consistency factor and the verification errors, suggesting that consistency values could be used to select the optimal feature combination.
5.	Offline Handwritten Signature Verification using Neural Network-2015	1.Neural network technique 2.Resilient back-propagation 3.Radical basic function	Using a database of 2106 signatures containing 936 genuine and 1170 forgeries	The accuracy of their system is 86.25. Their neural network says whether a signature is forged or genuine.

System Requirement Specifications

Hardware Interface:

- Operation System: Windows 10
- Hard Disk: 500 GB and up
- RAM: 1 GB and up
- Processor: Intel core i3 and up

Software Interface:

- Python (Pycharm IDLE)

Dependencies:

Image Processing -

- OpenCV
- Python Image Libarary(PIL)
- Scikit-Image

Machine Learning-

- Scikit-Learn
- Tensorflow
- Keras

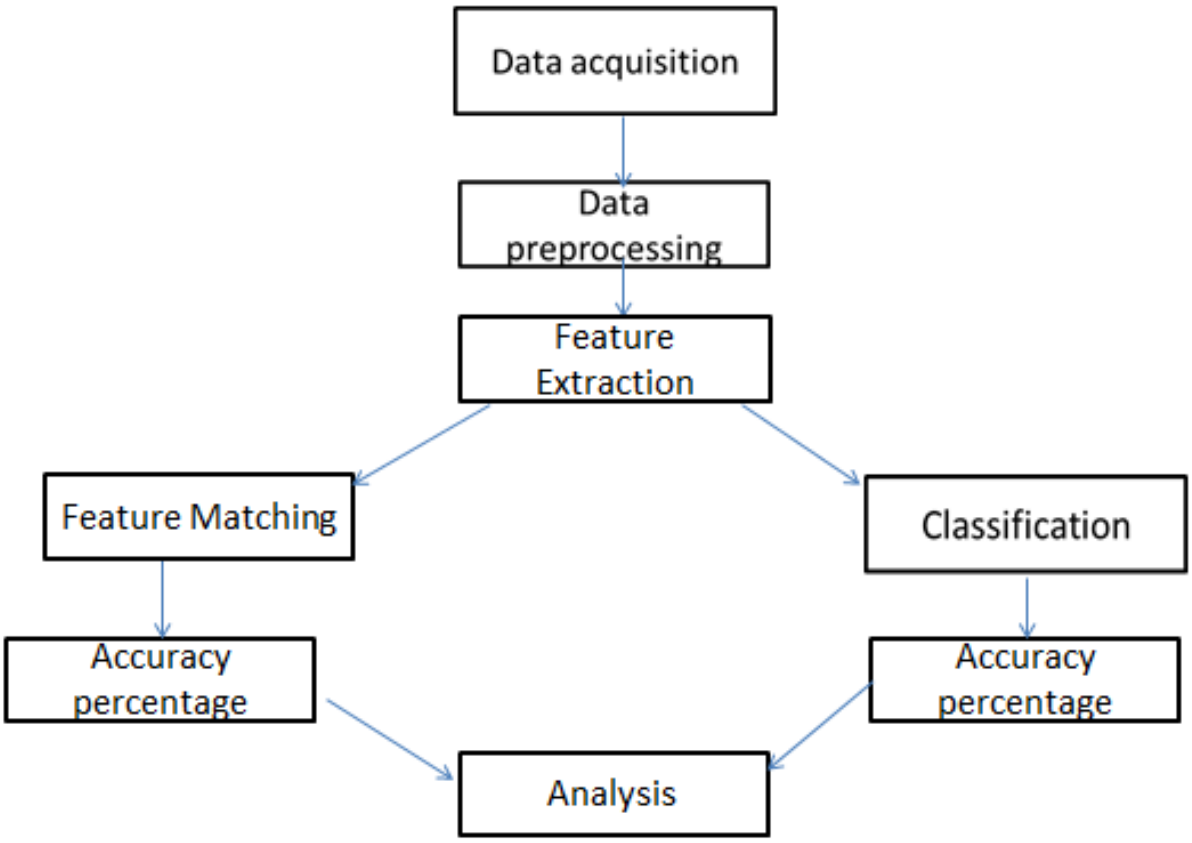
Mathematical Computations-

- SciPy
- NumPy

Dataset:

The dataset has been taken from Kaggle to conduct research on signature verification. Contains genuine and forged signatures of 30 people. Each person has 5 genuine signatures which they made themselves and 5 forged signatures someone else made. The naming of images is explained here.NFI-00602023 is an image signature of person number 023 and done by person 006. This is a forged signature.NFI-02103021 is an image of signature of person number 021 done by person 021. This is a genuine signature.

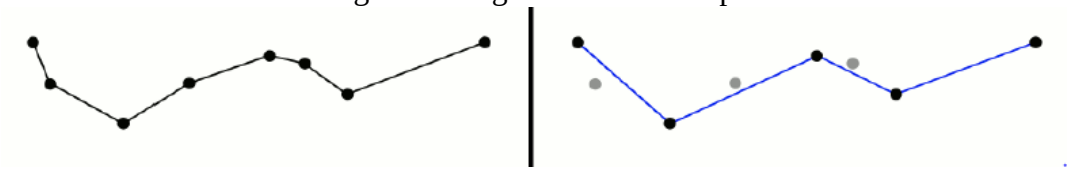
Proposed Approach : System Flow Diagram:



Pre-processing

The following steps are performed in data pre-processing :-

1) Filtering
   Filtering is done to remove noise from signature
   Ramer-Douglas-Peucker algorithm is used for filtering. After applying number of coordinates decreased in signature. Signature looks crisp and noise is removed.



2) Conversion from RGB to GrayScale
   Calculation becomes easier as we have perform the calculations in one channel rather than three.

3) Size Normalization
   Size normalization is done because no individual can do signatures of same size every time he does signature. Hence Size normalization is necessary. Every point is normalized between 0 and 1

Feature Extraction

In simple words, features are nothing but the unique properties that defines an image.

An image consists of pixels. Considering each pixel can have an 8bit value, even a 640x480 image will have 640x480x8 bits of information.. Too much for a computer to make head or tail out of it directly. So in feature extraction we figure out what parts of an image are distinctive , like lines, corners, special patches that can uniquely describe the image.

We look for the regions in images which have maximum variation when moved (by a small amount) in all regions around it. So finding these image features is called Feature Description.

Once we have the features and its description, you can find same features in all images and align them, stitch them for feature matching in the next step.

We have used three algorithms for feature extraction purpose:

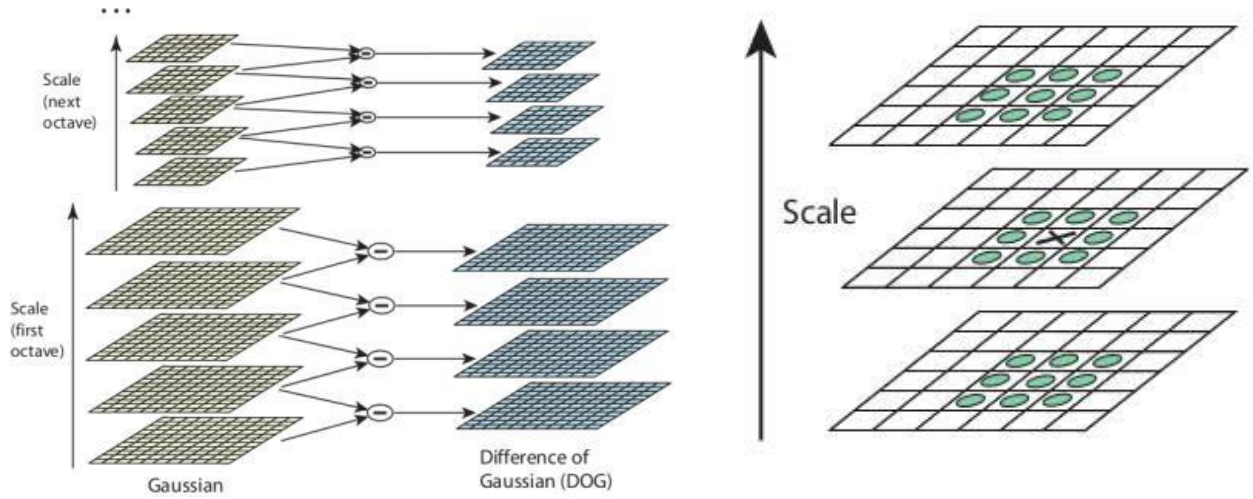
1) Scale Invariant Feature Transform (SIFT)
2) Speeded Up Robust Features Detection (SURF)
3) KAZE

Scale Invariant Feature Transform (SIFT)

- SCALE SPACE EXTREMA DETECTION: It is based on DoG, hence we further down-sample the image.

DOG- Difference of Gaussian, The DoG function D(x, y,σ) can be computed without convolution by subtracting adjacent scale levels of a Gaussian pyramid separated by a factor k.

- We need to select the center pixel and see whether it is greater than every other pixel around it .

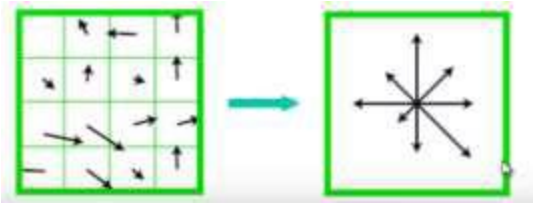


- KEYPOINT LOCALIZATION : There are some changes in intensity values upon changing the brightness and other things and hence to prevent changes in the keypoints some measures have to be taken. So we have intensity threshold determined empirically and similarly we have contrast threshold determined.

We also have hessian matrix for computing the curvature.

ORIENTATION ASSIGNMENT: We have to have , the keypoints invariant to rotation.

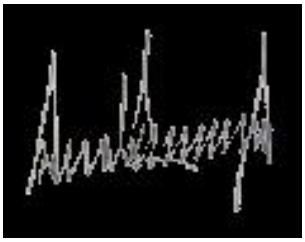
- A set of orientation histograms is created where each histogram contains samples from a 4×4 sub-region of the original neighbourhood region and having eight orientations bins in each.



- Hence we have 36 bins of size 10 each (36 X 10 = 360 degrees).
- The peak is the direction of the key point.

· SIFT DESCRIPTOR: The descriptor is then formed from a vector containing the values of all the orientation histograms entries.
Since there are 4 × 4 histograms each with 8 bins, the feature vector has 4 × 4 × 8 = 128 elements for each key point

- Total Feature key points detected = 78



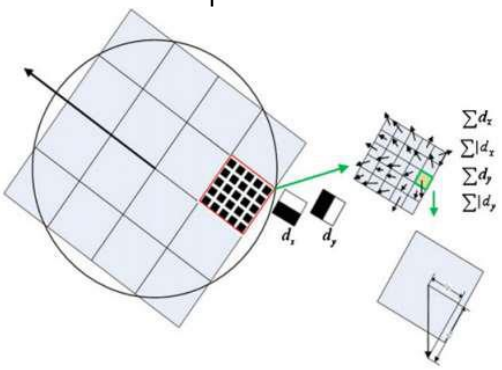
Speeded Up Robust Features Detection (SURF)

It uses 2D Box Filter like Hessian Operator unlike SIFT its basic idea is to approximate the second order Gaussian derivatives in an efficient way with the help of integral images using a set of box filters.

$$\det(H_{approx}) = D_{xx}D_{yy} - (wD_{xy})^2$$

- w is a relative weight for the filter response and it is used to balance the expression for the Hessian's determinant.
- The approximated determinant of the Hessian represents the blob response in the image.
- The SURF descriptor starts by constructing a square region centered around the detected interest point, and oriented along its main orientation

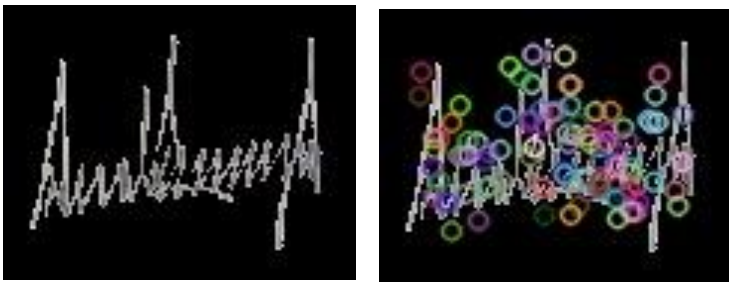
The interest region is further divided into smaller 4 × 4 sub-regions and for each sub region the Harr wavelet responses in the vertical and horizontal directions.



$$v = \left(\sum d_x, \sum |d_x|, \sum d_y, \sum |d_y|\right)$$

- The wavelet responses dx and dy are summed up for each sub-region and entered in a feature vector v.
- Computing this for all the 4 × 4 sub-regions, resulting a feature descriptor of length 4 × 4 × 4 = 64 dimensions.

- Total Feature key points detected = 95



KAZE

KAZE Features is a novel 2D feature detection and description method that operates completely in a nonlinear scale space. Previous methods such as SIFT or SURF find features in the Gaussian scale space (particular instance of linear diffusion). However, Gaussian blurring does not respect the natural boundaries of objects and smoothes in the same degree details and noise when evolving the original image through the scale space.

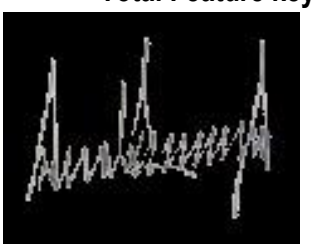
By means of nonlinear diffusion we can detect and describe features in nonlinear scale spaces keeping important image details and removing noise as long as we evolve the image in the scale space. We use variable conductance diffusion which is one of the simplest cases of nonlinear diffusion. The nonlinear scale space is build efficiently by means of Additive Operator Splitting (AOS) schemes, which are stable for any step size and are parallelizable.

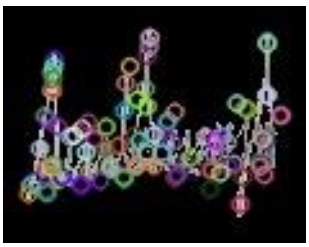
It exploit non-linear scale space through non-linear diffusion filtering . This makes blurring in images locally adaptive to feature-points, thus reducing noise and simultaneously retaining the boundaries of regions in subject images. KAZE detector is based on scale normalized determinant of Hessian Matrix which is computed at multiple scale levels. The maxima of detector response are picked up as feature-points using a moving window. Feature description introduces the property of rotation invariance by finding dominant orientation in a circular neighborhood around each detected feature. KAZE features are invariant to rotation, scale, limited affine and have more distinctiveness at varying scales with the cost of moderate increase in computational time.

$$\frac{\partial L}{\partial t} = div(c(x, y, t).\nabla L)$$

Equation shows the standard nonlinear diffusion formula. Where "c" is conductivity function, "div" is divergence, "∇" is gradient operator and "L" is image luminance

- **Total Feature key points detected = 96**





Feature Matching

Once the features and their descriptors have been extracted from two or more images, the next step is to establish some preliminary feature matches between these images.
For feature matching, we have used FLANN based Matcher

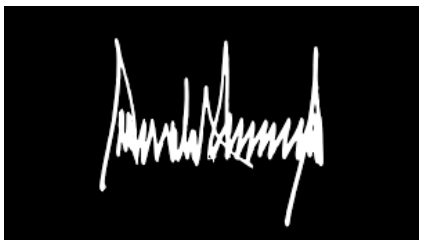
- FLANN stands for Fast Library for Approximate Nearest Neighbours. It contains a collection of algorithms optimized for fast nearest neighbour search in large datasets and for high dimensional features. It works more faster than BFMatcher for large datasets as well as match more number of feature key points.

To suppress matching candidates for which the correspondence may be regarded as ambiguous, the ratio between the distances to the nearest and the next nearest image descriptor is required to be less than some threshold.

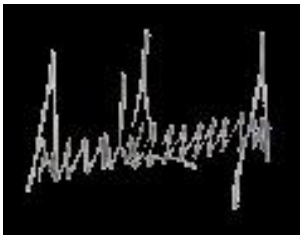
In our case we have used the threshold value of 0.85

Results obtained after feature matching :

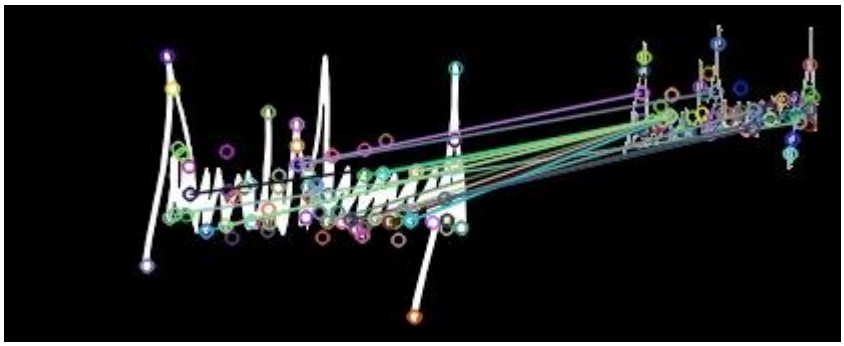
Original signature



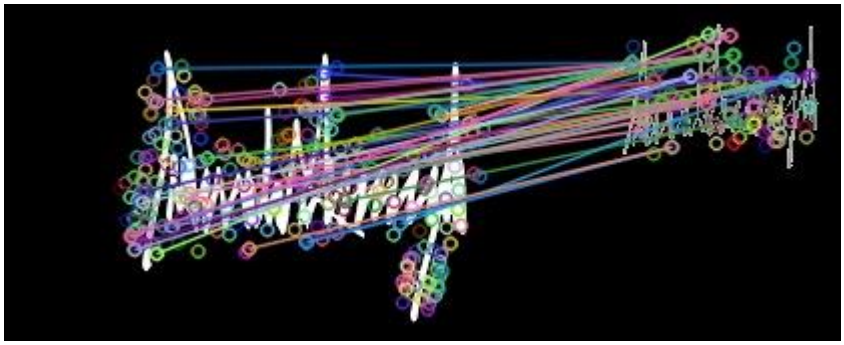
Forged signature



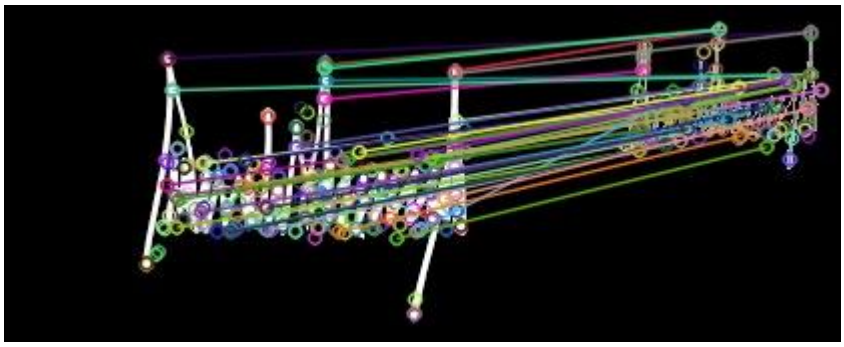
SIFT feature matching
Similarity accuracy=25.64%



SURF feature matching
Similarity accuracy=42.10%



KAZE feature matching
Similarity accuracy=40.62%

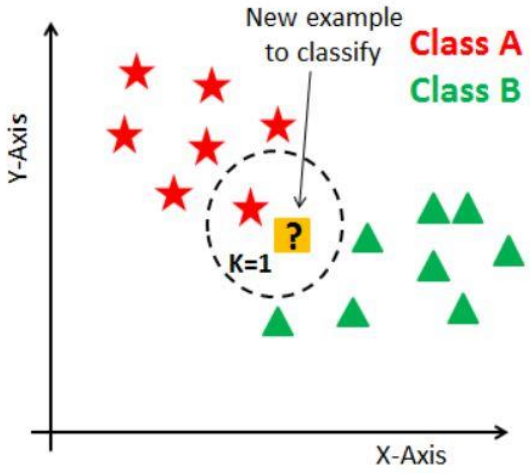


Classification

After forming feature vector final step in the verification process is classification which can be done by many machine learning algorithms like KNN, Neural Networks etc. After experimenting with the classifiers we found that the best model for our Signature Verification Process.

K-Nearest Neighbour(KNN)

- Global features from the signatures are extracted using random transform. For each registered user in the system database a number of reference signatures are enrolled and aligned for statistics information extraction about his signature. Extreme points warping algorithm is used to align two signatures.
- During the k-nearest neighbor classifier training, a number of genuine and forged signatures are chosen. A test signature's verification is established by first aligning it with each reference signature for the claimed user. The signature is then classified as genuine or forgery, according to the alignment scores which are normalized by reference statistics, using standard pattern classification techniques.
- In KNN, K is the number of nearest neighbors. The number of neighbors is the core deciding factor. K is generally an odd number if the number of classes is 2. When K=1, then the algorithm is known as the nearest neighbor algorithm.
- Suppose P1 is the point, for which label needs to predict. First, you find the one closest point to P1 and then the label of the nearest point assigned to P1.



- Suppose P1 is the point, for which label needs to predict. First, you find the k closest point to P1 and then classify points by majority vote of its k neighbors. Each object votes for their class and the class with the most votes is taken as the prediction. For finding closest similar points, you find the distance between points using distance measures such as Euclidean distance, Hamming distance, Manhattan distance and Minkowski distance.

- KNN has the following basic steps:

  1. Calculate the Euclidean distance

  2. Find closest neighbours

  3. Vote for labels

Convolutional Neural Network (CNN)

The next classification algorithm used is CNN image classification which takes an input image, process it and classify it under the categories genuine or forged . 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 x w x d(h = Height, w = Width, d = Dimension). Eg., An image of 6 x 6 x 3 array of matrix of RGB (3 refers to RGB values) and an image of 4 x 4 x 1 array of matrix of grayscale image.

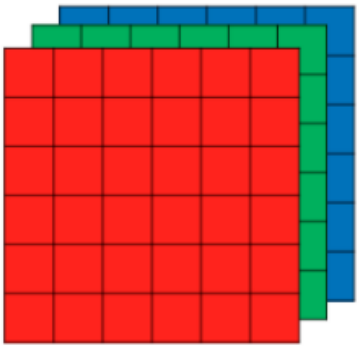


Figure 1 : Array of RGB Matrix

Technically, deep learning CNN models to train and test, each input image will pass it through a series of convolution layers with filters (Kernals), Pooling, fully connected layers (FC) and apply Softmax function to classify an object with probabilistic values between 0 and 1. The below figure is a complete flow of CNN to process an input image and classifies the objects based on values.

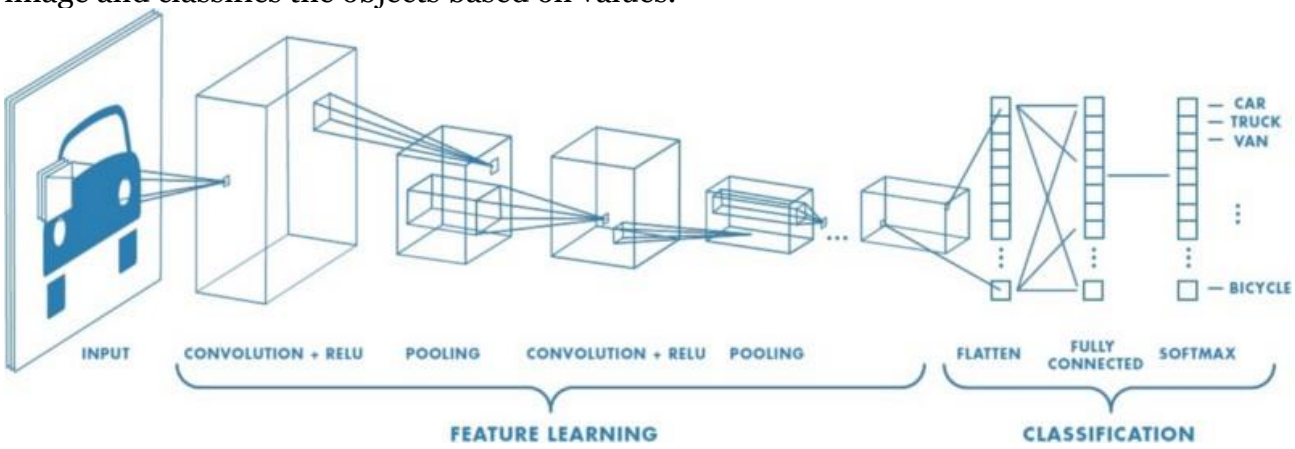


Figure 2 : Neural network with many convolutional layers

Convolution Layer

Convolution is the first layer to extract features from an input image. Convolution preserves the relationship between pixels by learning image features using small squares of input data. It is a mathematical operation that takes two inputs such as image matrix and a filter or kernal

- An image matrix (volume) of dimension **(h x w x d)**
- A filter **(f_h x f_w x d)**
- Outputs a volume dimension **(h - f_h + 1) x (w - f_w + 1) x 1**

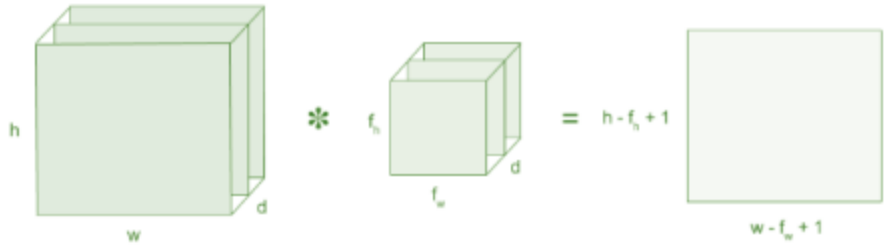


Figure 3: Image matrix multiplies kernel or filter matrix

Consider a 5 x 5 whose image pixel values are 0, 1 and filter matrix 3 x 3 as shown in below

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

5 x 5 – Image Matrix

*

1	0	1
0	1	0
1	0	1

3 x 3 – Filter Matrix

Figure 4: Image matrix multiplies kernel or filter matrix

Then the convolution of 5 x 5 image matrix multiplies with 3 x 3 filter matrix which is called **Feature Map.**

Strides

Stride is the number of pixels shifts over the input matrix. When the stride is 1 then we move the filters to 1 pixel at a time. When the stride is 2 then we move the filters to 2 pixels at a time and so on. The below figure shows convolution would work with a stride of 2.

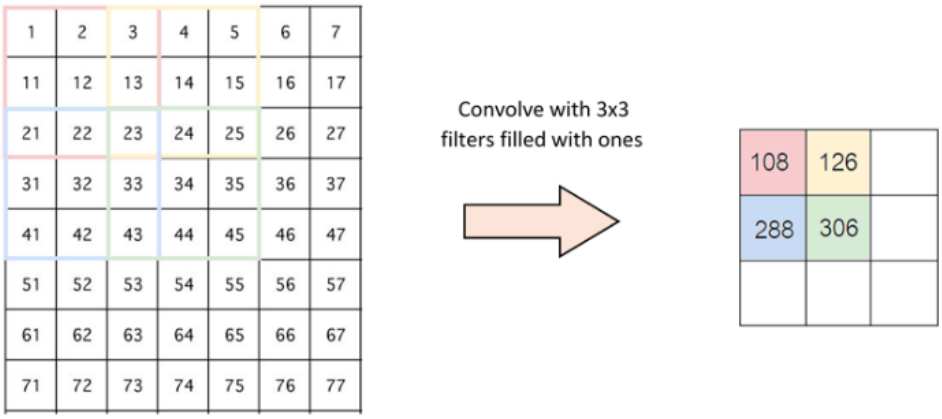


Figure 6 : Stride of 2 pixels

Padding

Sometimes filter does not fit perfectly fit the input image. We have two options:

- Pad the picture with zeros (zero-padding) so that it fits
- Drop the part of the image where the filter did not fit. This is called valid padding which keeps only valid part of the image.

Non Linearity (ReLU)

ReLU stands for Rectified Linear Unit for a non-linear operation. The output is $f(x) = max(0,x)$.

Why ReLU is important : ReLU's purpose is to introduce non-linearity in our ConvNet. Since, the real world data would want our ConvNet to learn would be non-negative linear values.

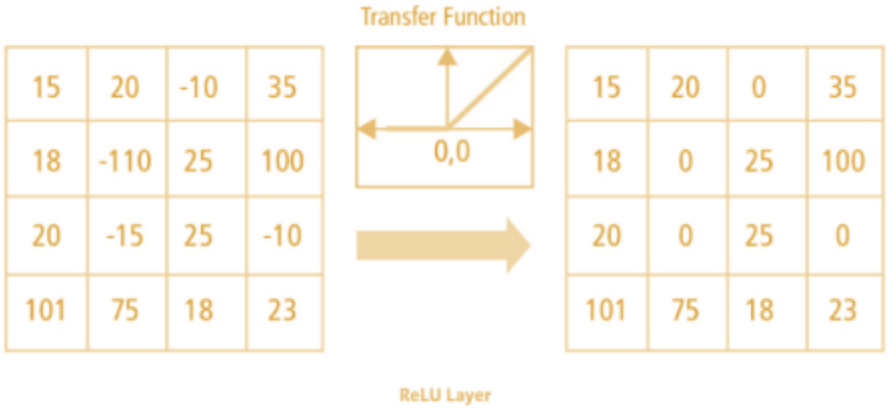


Figure 7 : ReLU operation

There are other non linear functions such as tanh or sigmoid can also be used instead of ReLU. Most of the data scientists uses ReLU since performance wise ReLU is better than other two.

Pooling Layer

Pooling layers section would reduce the number of parameters when the images are too large. Spatial pooling also called subsampling or downsampling which reduces the dimensionality of each map but retains the important information. Spatial pooling can be of different types:

- Max Pooling
- Average Pooling
- Sum Pooling

Max pooling take the largest element from the rectified feature map. Taking the largest element could also take the average pooling. Sum of all elements in the feature map call as sum pooling.

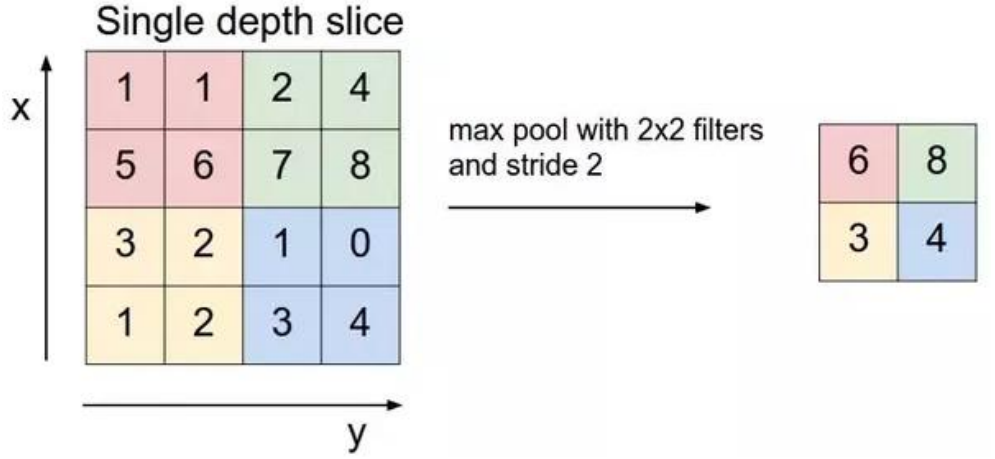


Figure 8 : Max Pooling

Fully Connected Layer

The layer we call as FC layer, we flattened our matrix into vector and feed it into a fully connected layer like neural network.

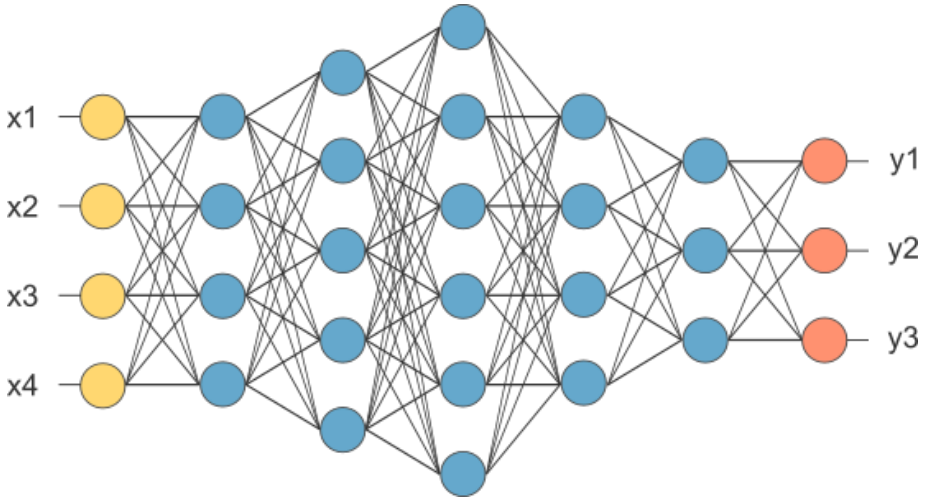


Figure 9 : After pooling layer, flattened as FC layer

In the above diagram, feature map matrix will be converted as vector (x1, x2, x3, ...). With the fully connected layers, we combined these features together to create a model. Finally, we have an activation function such as softmax or sigmoid to classify the outputs as a genuine signature or a forged signature.

For our problem, we have used the the following configuration:-

- Adam (Adaptive Moment Estimation) as the optimizer
- Optimizer parameter is to choose the stochastic gradient descent algorithm.
- Binary cross entropy as the loss parameter to choose the loss function.
- Sigmoid activation function for the final layer.
- Shear range = 0.2 , Zoom range =0.2
- Target size = (64,64)
- Batch size = 32
- Class mode = binary
- For training set , Epochs = 5, Number of steps per epoch = 100
- For test set, Validation steps = 50

Results

SIFT:

Featues obtained in genuine set = 111

Featues obtained in forged set = 78

Features Matching = 20

Accuracy obtained = 25.64 %

SURF:

Featues obtained in genuine set = 264

Featues obtained in forged set = 95

Features Matching = 40

Accuracy obtained = 42.10 %

KAZE:

Featues obtained in genuine set = 221

Featues obtained in forged set = 96

Features Matching = 39

Accuracy obtained = 40.62 %

KNN

Image size = (400,200)

k = 5

Bins = (200 , 100 ,50)

Accuracy in 200 = 69.07 %

Accuracy in 100 = 77.31 %

Accuracy in 50 = 87.63 %

The input image is classified into the right class.

Therefore the final accuracy = 87.63 %

CNN

Accuracy obtained after epoch number 1 = 57.95 %

Accuracy obtained after epoch number 2 = 78.45 %

Accuracy obtained after epoch number 3 = 92.69 %

Accuracy obtained after epoch number 4 = 95.63 %

Accuracy obtained after epoch number 5 = 97.10 %

Number of steps performed per epoch = 100

The input image is classified into the right class

Therefore the final accuracy = 97.10 %

Final Results Obtained :

Techniques(algorithms) Used	Accuracy Percentage
SIFT	25.64%
SURF	42.10%
KAZE	40.62%
KNN	87.63%
CNN	97.10%

Conclusion

We experimented with several variations on signature verification tasks. We showed that convolutional neural networks do an excellent job of verifying signatures when allowed access during training to examples of genuine and forged signatures of the same people whose signatures are seen at test time. We then conducted an experiment where we tested our network on the signatures of new people whose signatures had not been seen at all during training, resulting in performance little better than a naive baseline due to the inherent difficulty of this task. Finally, we proposed a novel architecture for the comparison of signatures which has promise for future work in signature verification, specifically in situations where a possibly-forged signature can be compared to known genuine signatures of a specific signer. We propose two directions for future work. First, access to more resources would allow us to achieve better performance on our main task. Specifically, being able to train on a larger dataset with more signature examples per person could achieve higher accuracies, as well as training a larger network for more epochs, which we were not able to do due to time and computational resource constraints. We were also constrained here by the fact that our dataset was relatively small, only on the order of thousands of examples, and that it is difficult to find good publically available signature datasets.

We are acquiring good accuracies comparable to other algorithms in this field.

It is experimentally verified that CNN model with approx 97.10 % accuracy has the best result compared with the traditional feature extraction algorithms like SIFT, SURF, Kaze as well the KNN algorithm which showed an accuracy of approx 87.63 %.

Moreover CNN is also has more memory efficiency as it reduce the size of the training data and just retain the information which are neccesary and do not add extra information.

References

1) Napa Sae-Bae and Nasir Memon, “Online Signature Verification on Mobile Devices”, IEEE transactions on information forensics and security, Vol. 9, No. 6, June 2014.

2) Mariano López-García, Rafael Ramos-Lara, Oscar Miguel-Hurtado, and Enrique Cantó-Navarro, “Embedded System for Biometric Online Signature Verification”, IEEE Transactions on industrial informatics, Vol. 10, NO. 1, February 2014.

3) Moises Diaz, Andreas Fischer, Miguel A. Ferrer, Réjean Plamondon, “Dynamic Signature Verification System Based on One Real Signature”, IEEE Transactions on Cybernetics 2016.

4) Marianela Parodi, Juan C.Gómez, “Legendre polynomials based feature extraction for online signature verification. Consistency analysis of feature combinations”, Elsevier Volume 47, Issue 1, January 2014, Pages 128-140.

5) Pallavi V. Hatkar, Prof.B.T.Salokhe, Ashish A.Malgave, “Offline Handwritten Signature Verification using Neural Network”, International Journal of Innovations in Engineering Research and Technology [IJIERT], Vol 2, Issue 1, Jan 2015..

6) https://docs.opencv.org/3.0-beta/doc/py_tutorials/py_tutorials.html

7) https://medium.com/@RaghavPrabhu/understanding-of-convolutional-neural-network-cnn-deep-learning-99760835f148

8) https://medium.com/@YearsOfNoLight/intro-to-image-classification-with-knn-987bc112f0c2