** PES University, Bangalore**

(Established under Karnataka Act No. 16 of 2013)

**MAY 2020: IN SEMESTER ASSESSMENT (ISA) B.TECH. IV SEMESTER**

**UE18MA251- LINEAR ALGEBRA**

MINI PROJECT REPORT

ON

**HANDWRITTEN NAMES RECOGNITION USING PRINCIPAL COMPONENT ANALYSIS**

*Submitted by*

1. Shreya sri PES1201800132

2. Mahah Sadique PES1201801529

Branch & Section : Computer Science Engineering, Section J

PROJECT EVALUATION

( For Official Use Only )

|  |  |  |  |
| --- | --- | --- | --- |
| Sl.No. | Parameter | Max Marks | Marks Awarded |
| 1 | Background & Framing of the problem | 4 |  |
| 2 | Approach and Solution | 4 |  |
| 3 | References | 4 |  |
| 4 | Clarity of the concepts & Creativity | 4 |  |
| 5 | Choice of examples and understanding of the topic | 4 |  |
| 6 | Presentation of the work | 5 |  |
|  | Total | 25 |  |

Name of the Course Instructor :

Signature of the Course Instructor :

***Abstract* –**

**INTRODUCTION**

Handwriting can be defined as a person’s writing using a pen, pencil or any other writing instrument. Handwriting is like fingerprint. It is unique for each person. Even identical twins have different styles of writing. It is very unlikely for two individuals to have the same handwriting styles. The circumstances in which a person grew up, their environment, the first language they learnt and several other factors all contribute to the overall uniqueness of a person’s handwriting. Handwriting Recognition is now a highly exciting field of research. Recognition of the handwritten document is not an easy task since each person has his/her own style of writing. This plays a very important role when it comes to understanding the prescription written by the doctors, the amount written on a cheque etc. Hence the understanding of the handwritten document is very important. The goal of this project is to solve the task of name transcription from handwriting images implementing a Neural Network approach on a database with a large number of images of handwritten names using Principal Component Analysis for feature extraction. It can be performed in either offline or online. The approach taken here is offline character recognition. Offline handwritten character recognition focuses on documents that have been written on papers. The data is presented to the system as an image, requiring a segmentation of the writing from the image background before recognition can be done.

**LITERATURE SURVEY**

Researchers all over the world have achieved successful results in handwriting recognition.

There are fundamentally three domains : 1)Numeral recognition 2) Character recognition 3)cursive recognition.

For this report we have focused on character recognition .

Some of the recent achievements in character recognition have been seen in the

Works of : Srihari[4] , with a recognition rate of about 89-93% , Liou and Yang

[5] with a recognition rate of 88-95 percent and finally Shustorovich[6] with

recognition rate of about 89.40-96.44% . Another notable work was Development of

English Handwritten Recognition Using Deep Neural Network by Teddy Surya and

Ahmad Fakhrur uses a Deep Neural Network model having two Encoding layer and

one SoftMax layer on the EMNIST dataset. Their accuracy using DNN was way

better than just using ANN but for simplicity purposes we have primarily focused on ANN .

Sakshi Mehta [7], in her work presents a theoretical and practical basis for pre-

processing on handwritten text for character recognition using feed forward neural

networks . Approach was made to improve accuracy of recognition of handwritten

characters . Rokus Arnold [8] presents the implementation of character recognition

using neural networks with the help of matlabs tool . In this paper they tried to

recognize the printed and handwritten characaters by projecting them on different

sized grids . The results showed that the accuracy of the character recognition

depends upon the resolution of character projection . Regardless of the orientation ,size

and the place of characters the network still had a 60% precision . Shyla afrogee

[9] describes an artificial neural network approach for the recognition of English

characters using feed forward neural network . Ankit Sharma [10] gives offline

strategies to recognize the characters . Image preprocessing is used along with the

binarization, thresholding and segmentation method . Back propagation is used to

classify the characters in order to recognize them. The method used here gives

85% accuracy for recognition.

**METHEDOLOGY**

The dataset used includes over 125,000 images of handwritten names along with human contributors’ transcription of these written names. Most names on the dataset are in French, which means that they can include accent marks.

The database has the following characteristics:

* 1 unique identifier per entry on the database.
* 1 URL to the corresponding image per entry.
* 1 transcription of the name per entry.
* 1 label indicating if it is a first name or a last name per entry

Since we have the transcription of the names this is a supervised learning problem.

We organized the implementation of the project according to the following tasks:

1. Pre-processing of the dataset
2. Pre-processing of the images
3. Feature extraction
4. Classifiers
5. Inference
6. Validation

**Pre-processing of the dataset**

The original dataset has many format problems, so the first step was to solve this. We have two types of images, some in which the name is written next to the NOM or PRENOM word and others which the name is written under those words.

**Examples of the two different types of images in the dataset**

We solved this problem by modifying the label next to the image that indicates whether the name is the first name or last name. If the name is located on the right the labels have the values “first” or “last” and if the name is located below the labels have the values “first\_b” or “last\_b”.

**Pre-processing of the images**

We used the pandas library to import the dataset from the CSV file. The dataset does not contain the images itself so we need to download them.

Apart from the names that we want to transcribe, the images contain a lot of noise that we want to erase: for example the code of the image, the word NOM: or PRENOM: or some split characters that are part of other names. NOM or PRENOM words appear in all the images and because they are written in a machine and then printed they have always the same exact form, so we extract the last two characters of these words as a reference from one image of the database. We used a template matching algorithm from the OpenCV library (chapter 7: Histograms and Matching) using this reference image to locate the NOM or PRENOM word. After this is done, we use the information about the location of the word to crop the image and extract the name. With the use of this procedure we can also crop some noise like parts of other names written above or below. Because the images are taken with the name centred, it would never touch the corners so to remove the noise we find any figure that is touching the lower border of the image and delete it.

After this what we have is an image that contains only the name that we want to recognize. The next step of the procedure is the character extraction. This is done with a clustering algorithm. First we binarize the image, then using clustering we calculate the different connected components of the binarized image. When this is done we calculate the coordinates of every component to extract the characters, but the problem is that not every character must be a unique component, for example the point of an ”i”, of one of the lines of an capital ”E”, so for calculating this coordinates we take in account that if a component is above other component both of them belong to the same character. The characters then are extracted and rescaled to a 28x28 image, this are the images that we will use to train our Neural Networks.

The data is split into two different sets: the training set and the test set for validation. We use the same training set to learn all the classifiers, and the same test set for evaluating their accuracy.

**Feature extraction**

We have implemented Principal Component Analysis as the method for feature extraction to obtain the appropriate results.

**Principal Component Analysis**

PCA is a linear transformation algorithm that seeks to project the original features of our data onto a smaller set of features while still retraining most of the information. This algorithm tries to find the most appropriate direction/angles that maximize the variance in the new subspace. The number of features used in our project was 100.



**PCA of the first two features**

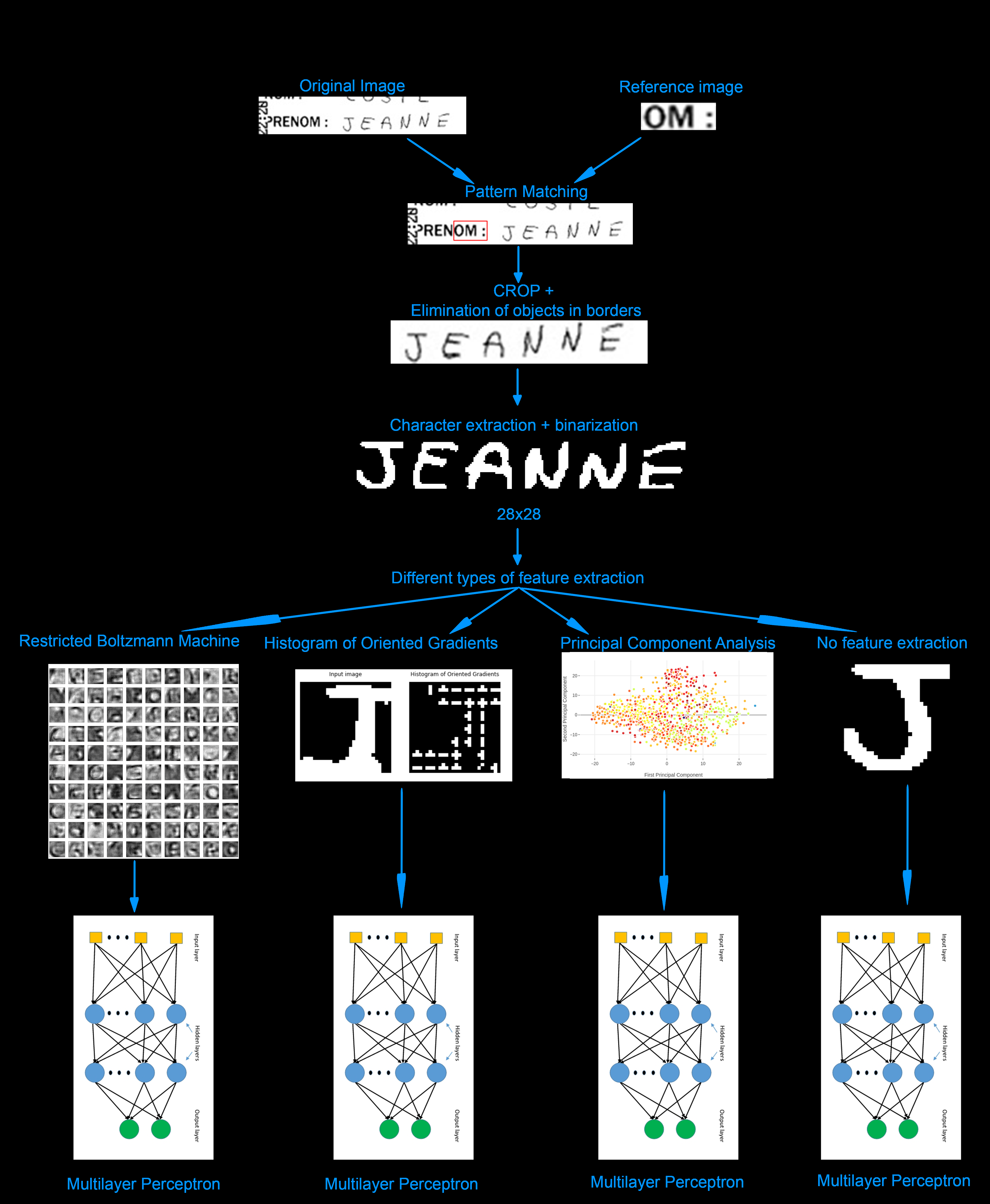
**No feature extraction**

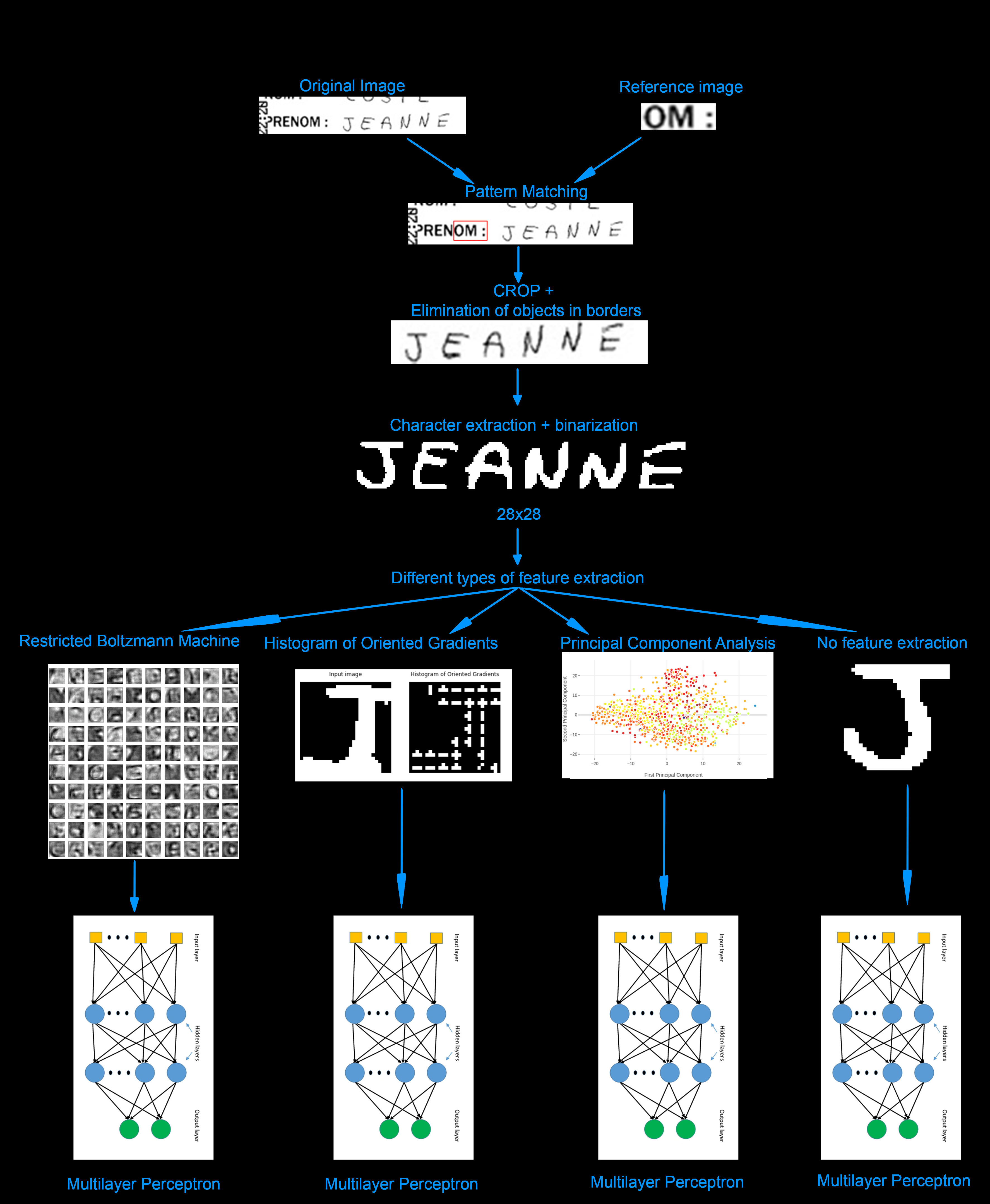
For the purpose of evaluating the previous method we also trained the neural networks without any type of feature extraction; that is using the binarized image of the characters.



**A Binarized character without feature extraction**

**The process followed to train the network:**





**Process diagram**

**Classifiers**

1. MLP classifier with PCA (Principal Component Analysis) features:

PCA:

* n\_components = 100

MLP:

* layers = (300, 400, 150)
* activation function = ’relu’ (Rectifier function)
* max iterations = 5000
* tol = 0.0001

1. MLP classifier with character images only:

* layers = (300, 400, 150)
* activation function = ’relu’ (Rectifier function)
* max iterations = 5000
* tol = 0.0001

The learning algorithm for the MLP is the default in Scikit-Learn: “Adam Optimizer”

**Inference**

Once the Neural Networks are trained the process to predict new names is very similar to the process to train it. The new image must follow the same process that the images used for training. That means, we need to extract each individual character on the image. Then, apply the feature extraction used. The multilayer perceptron then predicts a letter for each individual character.

The final step is to put the values predicted by the Neural Network together to form the name.

**Validation**

To validate our results we compute the Scikit classifier metrics and our own full name prediction metrics in the test data. Another possibility was to compute the cross-validation in the complete dataset but we used the split between train and test because it was simpler. We divided the dataset in 80% train 20% test batches.

We computed the classification report of Scikit metrics (which gives the precision, recall and f1- score for the classifier) for individual character prediction. We also tested full name recognition and we output the full correct name ratio and the correlation ratio (how similar all predicted names are to the original label) for the classifier.

**Implementation**

All the project steps were implemented in Python. We used pandas and urllib for reading the

Original database and getting the images from the servers, scikit-image, OpenCV and scipy for pre-processing the dataset, matplotlib and plotly for plotting different data and scikit-learn for the classification tasks.

**Results**

We ran our testing for the first 30000 names in the dataset because testing with more data was not possible with our machines (we don’t have enough RAM), the included prebuilt dataset and classifiers from our repository were from this testing.

The precision produced by MLP with PCA features and MLP classifiers in individual character recognition were 0:92 and 0:93 respectively.

The full name correct ratio MLP with PCA features and MLP classifiers were: 0:688 and 0:709 respectively.

The full name correlation ratios (similarity of name produced to original name) produced by the MLP with PCA features and MLP classifiers were 0:929 and 0:934 respectively.

**Conclusion**

In our project we applied a multilayer perceptron combined with Principal Component Analysis for feature extraction to the ”Transcriptions of names from handwriting” dataset. We have computed the accuracy of this Neural Networks and we observed that surprisingly the simplest implementation, the MLP without feature extraction produces the highest accuracy.

For future upgrades in this implementation we plan to work on the names where the character extraction gives us inconsistent results. Even so the program stores a list of all these names to make it possible to work with them in the future. Due to computational limitations we were not able to train with the full database, training with more names could help to improve the results. Also fine tuning of the parameters of the neural networks could improve the results but computational power is again a limitation.

**References**

[1] CrowdFlower. Data for everyone library: Transcriptions of names from handwriting [https://www.crowdflower.com/data-for-everyone/]

[2] OpenCV. Opencv documentation: Template matching [https://docs.opencv.org/2.

4/doc/tutorials/imgproc/histograms/template\_matching/template\_matching.html].

[3] Pierre Baldi and Kurt Hornik. Neural networks and principal component analysis: Learning from examples without local minima. Neural networks, 2(1):53–58, 1989.

[4] S. N. Srihari, 1993, Recognition of Handwritten and Machineprinted Text for Postal Address Interpretation, Pattern Recognition

[5] C-Y. Liou and H-C. Yang, 1996, Handprinted Character Recognition based on Spatial Topology Distance Measurement, IEEE Transactions on Pattern Analysis and Machine Intelligence, 18, 941-944.

[6] . Shustorovich, 1994, A Subspace Projection Approach to Feature Extraction: The Two-Dimensional Gabor Transform for Character Recognition, Neural Networks, 7, 1293-1301

[7] Sakshi Mehta, Khushbu. Image pre processing on Character recognition using Neural network.(IJCA 2013).

[8]Rokus Arnold, Poth Miklos.Character Recognition Using Neural Networks.IEEE,2010.

[9] Shyla Afrogee. OCR using Back propagation neural network.(ICECTE 2016).

[10].Ankit Sharma. Character recognition using neural network(IJETT 2013)