CONTENTS

Project Documentation

AN AUTOMATED OPTICAL CHARACTER RECOGNITION OF HANDWRITTEN ARABIC NUMERALS/DIGITS USING DECISION TREES & RANDOM FORESTS
PROJECT NUMBER 18

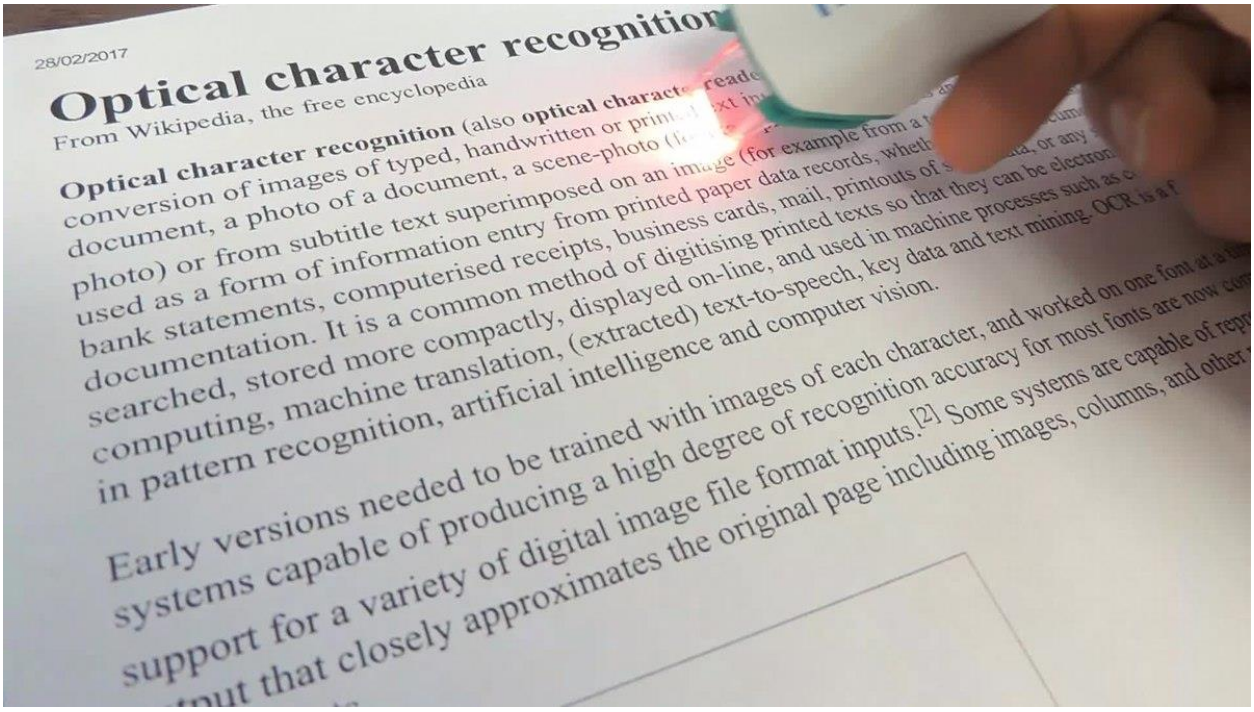
يونس يحيي احمد خليل	201901002	الفرقة الثالثة	IS
معتز رأفت ابراهيم ياسين	201900841	الفرقة الثالث	IS
محمد خالد مرسي جمعه	201900664	الفرقة الثالث	IS
هاجر محمد احمد ابراهيم	201900920	الفرقة الثالث	IS
نوران حاتم احمد حسين	201900909	الفرقة الثالث	IS
هايدي محمد احمد حسن	201900926	الفرقة الثالث	IS

PROJECT DOCUMENTATION

1/PROJECT IDEA IN DETAILS

A: OVERVIEW

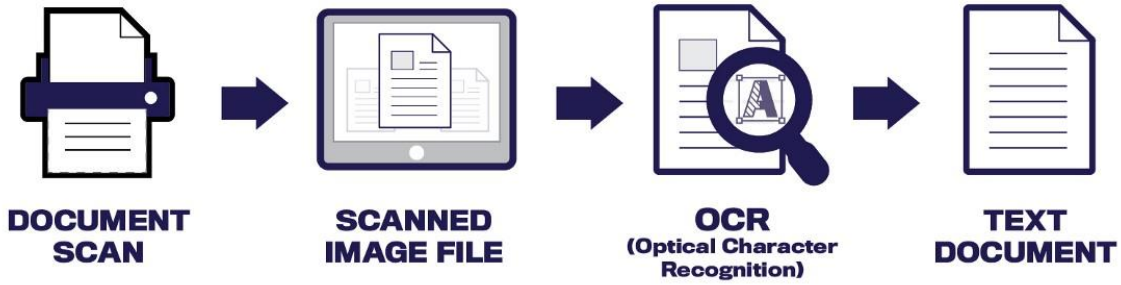
Optical character recognition or optical character reader (OCR) is the electronic or mechanical conversion of images of typed, handwritten or printed text of arabic language only into machine-encoded text, whether from a scanned document, a photo of a document, a scene-photo (for example the text on signs and billboards in a landscape photo) or from subtitle text superimposed on an image. it simplified a lot of our work in our life and make it easy in our studying life and make life better and we will discuss the main algorithm used in coding this project (decision tree and random forest) and some used datasets.



B: HOW IT WORKS

as a user view: we get a set of data image of a document or a photo of document or handwritten on a paper for different character numerals or digits and we will code our system to scan and identify every character and scan it into electronic or mechanical machine.

ex: a handwritten paper will be moved under the scanner then it will be copied into computer through an algorithm our ai system will be able to detect the character numerals or digits in paper.



2/MAIN FUNCTIONALITIES

Reading Dataset

This part of code takes dataset that contains 4 files 2 for training (pixels-labels) and 2 for testing (pixels-labels)

```
x=pd.read_csv("D:\FCAI\Year 3\Semster 1\AI\GP\csvTrainImages 60k x 784.csv")
y=pd.read_csv("D:\FCAI\Year 3\Semster 1\AI\GP\csvTrainLabel 60k x 1.csv")
testData=pd.read_csv("D:\FCAI\Year 3\Semster 1\AI\GP\csvTestImages 10k x 784.csv")
testLabel=pd.read_csv("D:\FCAI\Year 3\Semster 1\AI\GP\csvTestLabel 10k x 1.csv")
```

Building Decision tree

This part of code builds the decision tree

```
clf=DecisionTreeClassifier()
clf.fit(trainData,trainLabel)
```

Predicting & Getting Accuracy

This part of code predicts the given data set and gets us the accuracy of using the decision tree algorithm on it

```
p=clf.predict(testData)
print("Accuarcy =",metrics.accuracy_score(testLabel,p)*100,'%')
```

Output:

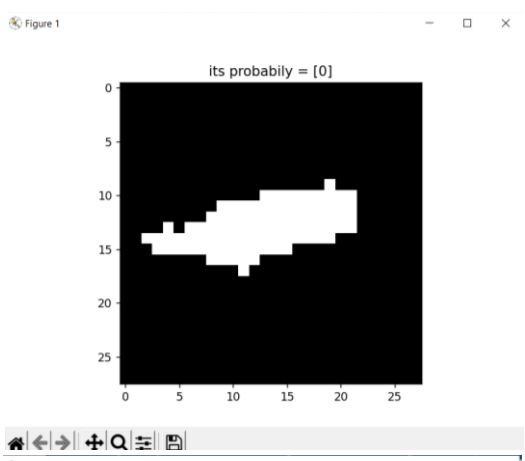
```
Accuarcy = 90.64906490649065 %
```

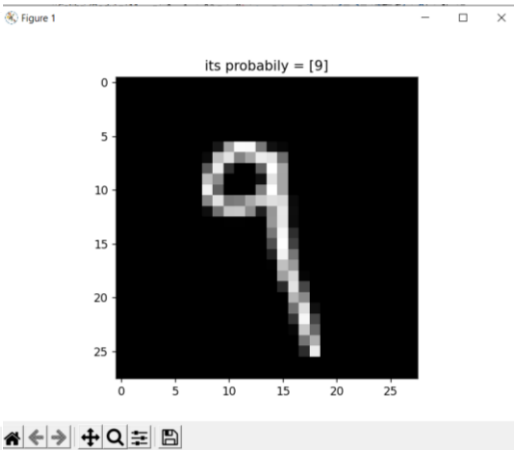
Predicting & Displaying

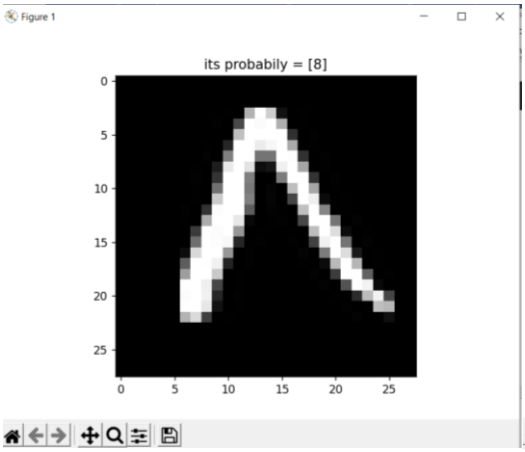
This part of code predicts 5 numbers (number of predictions is as we want) and display them as pictures with the prediction

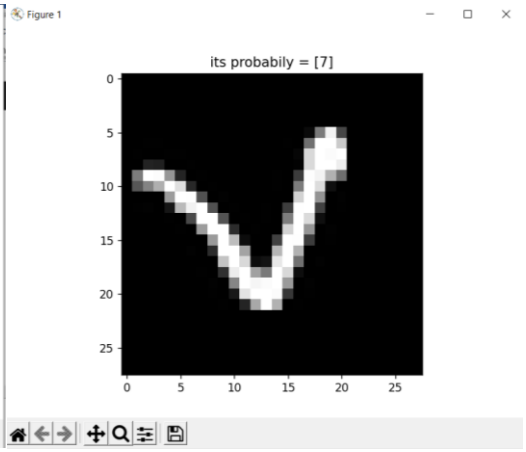
```
for x in range(5,10):
    y_predicted = clf.predict((testData.iloc[x].values).reshape(1, -1))
    label = y_predicted
    pixel = testData.iloc[x]
    pixel = np.array(pixel, dtype='uint8')
    pixel = pixel.reshape((28, 28))
    pt.title('its probabily = {label}'.format(label=label))
    pixel=np.transpose(pixel)
    pt.imshow(pixel, cmap='gray')
    pt.show()
```

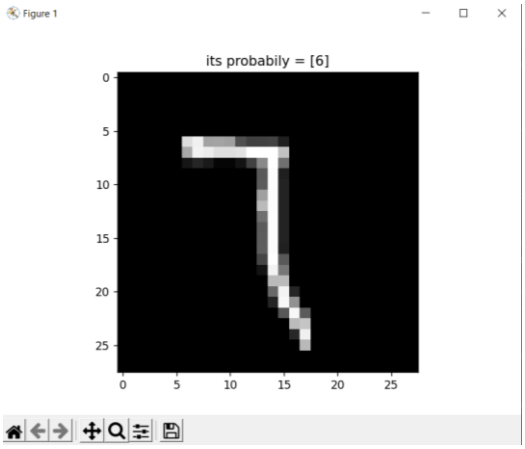
Output:











Train& Test& Split

This function is used to get the total accuracy of the dataset

```
x_train,x_test,y_train,y_test=train_test_split(trainData,trainLabel)
```

Building Random Forrest and getting accuracy

```
model =RandomForestClassifier()
model.fit(x_train,y_train)
print("Accuarcy = ",model.score(x_test,y_test)*100,'%')
```

Output:

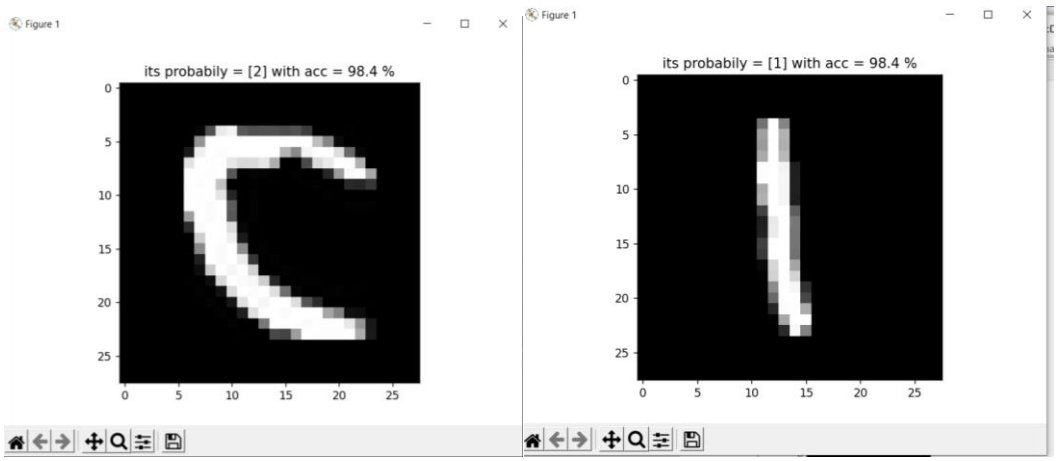
```
Accuarcy =  98.11999999999999 %
```

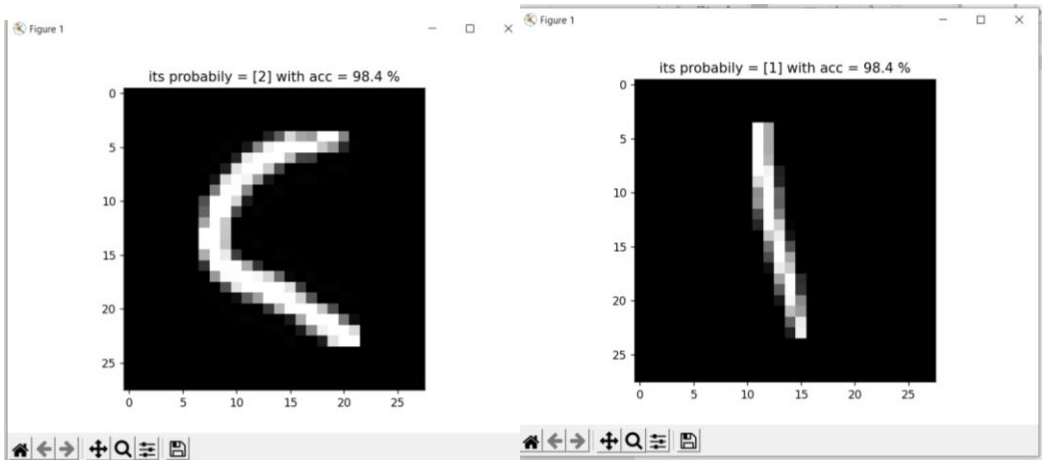
Predicting & Displaying

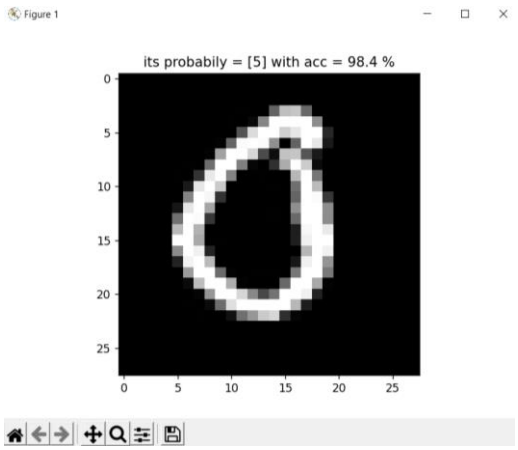
This part of code predicts 5 numbers (number of predictions is as we want) and display them as pictures with the prediction and accuracy

```
for x in range(0,5):
    y_predicted = model.predict((x_test.iloc[x].values).reshape(1, -1))
    label = y_predicted
    pixel = x_test.iloc[x]
    y_pred = model.predict(x_test)
    acc = accuracy_score(y_pred, y_test) * 100
    pixel = np.array(pixel, dtype='uint8')
    pixel = pixel.reshape((28, 28))
    pt.title('its probabily = {label} with acc = {acc} %'.format(label=label, acc=acc))
    pixel = np.transpose(pixel)
    pt.imshow(pixel, cmap='gray')
    pt.show()
```

Output:







Taking input as handwritten

At this part of code we take input from user as handwritten with paint program and he puts the picture in handwritten folder
things that we need disscus it with the user:that the file he puts the pictures in must be told to the programmer and naming of the pictures must be numbers

```
#take input from user
path="D:/FCAI/Year 3/Semster 1/AI/GP/HandWrittenTest/"
for x in range (1,10):
    img=cv.imread(path+"{x}.png".format(x=x))[:,:,0]
    img=np.invert(np.array([img]))
    img = np.transpose(img)
    label=model.predict(img.reshape(1, -1))
    img = np.transpose(img)
    pt.title('its probabily = {label} '.format(label=label))
    pt.imshow(img[0],cmap='gray')
    pt.show()
```

Output:9 given pictures by the user with their predictions

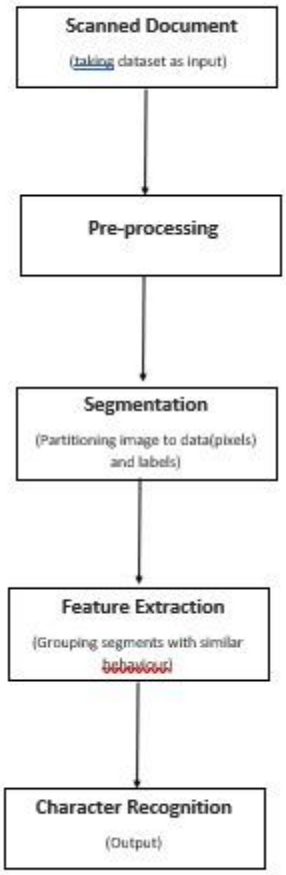
from the users' perspective

the user sees our system can only do 2 things
if he has his own pictures of handwritten digits he can put it in HandWrittenTest file and naming pictures as numbers from 0 to number of pictures
the system can predict these numbers

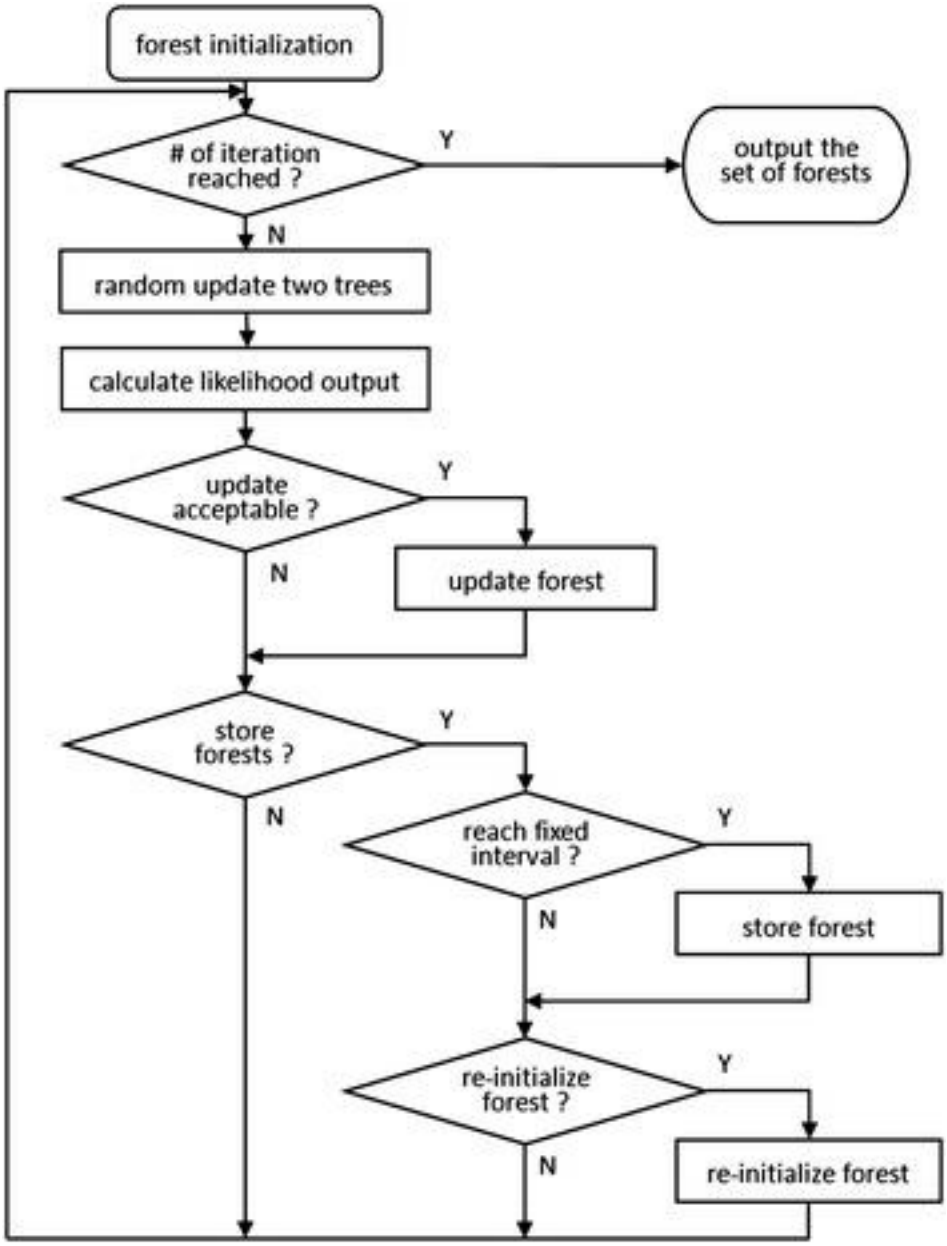
or

he can give the system test samples and the system will predict whatever he wants from these samples

Block Diagram

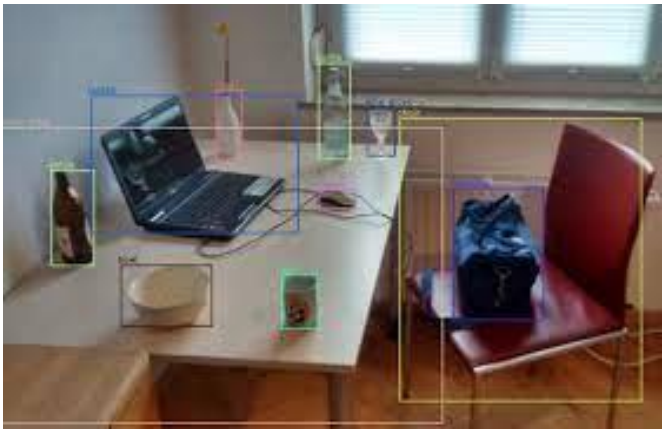


FLOWCHART FOR RANDOM FOREST



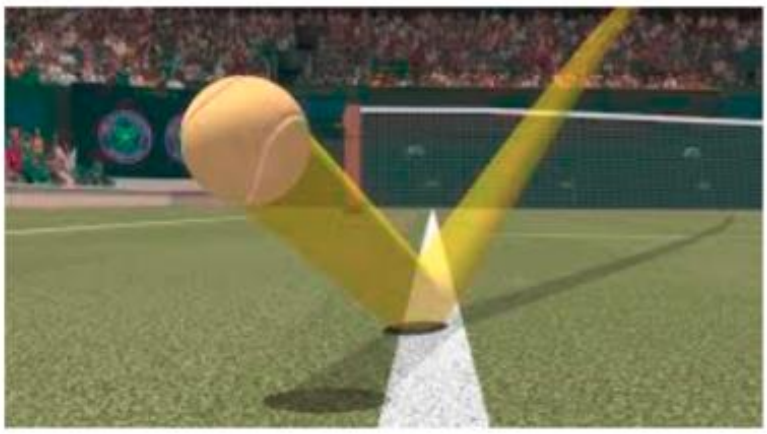
3/SIMILAR APPLICATION IN MARKET

1: DETECTION AND FACE RECOGNITION



2: OBJECT DETECTION

3: MEDICAL IMAGING



4: BALL TRACKING

IN SPORTS

5: AUTONOMOUS DRIVING

4/AN INITIAL LITERATURE REVIEW OF ACADEMIC PUBLICATIONS RELEVANT TO THE IDEA

PAPER ONE

CHARACTER DETECTION USING RANDOM FOREST ALGORITHM

Content

Abstract

Classification step is one of the most important tasks in any recognition system. This step depends greatly on the quality and efficiency of the extracted features, which in turn determines the efficient and appropriate classifier for each system. This study is an investigation of using both K- Nearest Neighbor (KNN) and Random Forest Tree (RFT) classifiers with previously tested statistical features. These features are independent of the fonts and size of the characters. First, a binarization procedure has been performed on the input characters images, and then the main features have been extracted. The features used in this paper are statistical features calculated on the shapes of characters. A comparison between KNN and RFT classifiers has been evaluated. RFT found to be better than KNN by more than 11 % recognition rate. The effect of different parameters of these classifiers has also been tested, as well as the effect of noisy characters.

Introduction

Recognition is an area that covers various fields such as, face recognition, finger print recognition, image recognition, character recognition, numerals recognition, etc. Handwritten character Recognition System is an intelligent system able to classify handwritten character as human see. Character classification is an important part in many computer vision and image problems like possessing Optical character recognition, license Plate recognition, etc. The classification of handwritten character is a more difficult task due to the different handwriting styles of the writers

Arabic Digit	English Digit	Image	Inverted Image
١	1		
٢	2		
٣	3		
٤	4		
٥	5		
٦	6		
٧	7		
٨	8		
٩	9		
٠	0		

Methodology

- Image Binarization

Binarization plays a key role in processing degraded images. In general, binarization is either global or local. In a global approach, threshold selection leads to a single threshold value for the entire image often based on an estimation of the background level from the intensity histogram of the image. Unlike global approaches, local binarization use different values for each pixel according to the local area information

- Feature Extraction

One of the basic steps of pattern recognition is features selection. Features should distinguish between classes, be invariant to input variability, and also be limited in number to

compute discriminate functions efficiently and to limit the amount of needed training data . Statistical features look for a typical spatial distribution of the pixel values that define each character. There are 14 statistical features extracted from each character, four of them are for the whole image as listed below:

1. Height / Width.

2. Number of black pixels / number of white pixels.

3. Number of horizontal transitions.

4. Number of vertical transitions.

The horizontal and vertical transitions are a technique used to detect the curvature of each character and found to be effective for this purpose . The procedure runs a horizontal scanning through the character box and finds the number of times that the pixel value changes state from 0 to 1 or from 1 to 0 as shown in figure 2. The total number of times that the pixel status changes, is its horizontal transition value. Similar process is used to find the vertical transition value .

The other 10 features are extracted after dividing the image of the character into four regions to get the following ratios as shown in figure 3:

1. Black Pixels in Region 1/ White Pixels in Region 1

2. Black Pixels in Region 2/ White Pixels in Region 2

3. Black Pixels in Region 3/ White Pixels in Region 3

4. Black Pixels in Region 4/ White Pixels in Region 4

5. Black Pixels in Region 1/ Black Pixels in Region 2

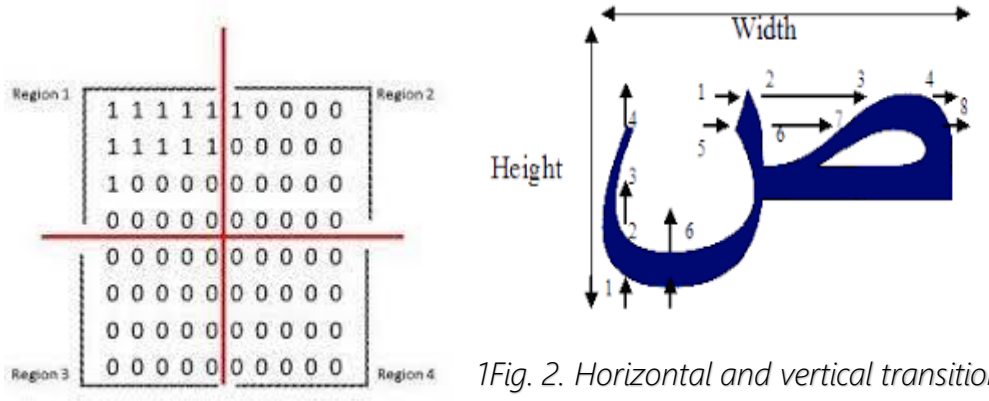
6. Black Pixels in Region 3/ Black Pixels in Region 4

7. Black Pixels in Region 1/ Black Pixels in Region 3

8. Black Pixels in Region 2/ Black Pixels in Region 4

9. Black Pixels in Region 1/ Black Pixels in Region 4

10. Black Pixels in Region 2/ Black Pixels in Region 3



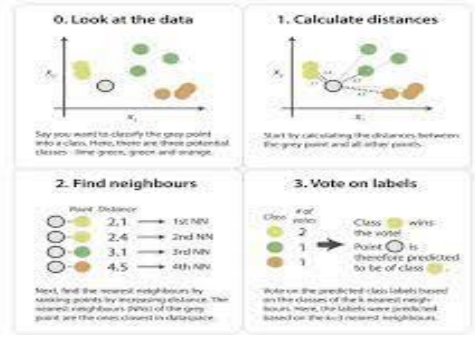
1Fig. 2. Horizontal and vertical transitions

2Fig. 3. Dividing the image and extract the features

K-Nearest Neighbor Algorithm

KNN is a type of instance-based learning, or lazy learning where the function is only approximated locally and all computation is deferred until classification. The k-nearest 14 M. Rashad and N.A. Semary neighbor algorithm is amongst the simplest of all machine learning algorithms: an object is classified by a majority vote of its neighbors, with the object being assigned to the class most common amongst its k nearest neighbors (k is a positive integer, typically small). If k = 1, then the object is simply assigned to the class of its nearest neighbor. The training phase for KNN consists of simply storing all known instances and their class labels. If we want to tune the value of K, n-fold cross-validation can be used on the training dataset. The testing phase for a new instance "T", in a given known set "I" is as follows:

1. *Compute the distance between T and each instance in I*
2. *Sort the distances in increasing numerical order and pick the first K elements*
3. *Compute and return the most frequent class in the K nearest neighbors*



Random forest

Random Forest was introduced in 2001 by Leo Breiman on the basis of bagging The Random Forest algorithm is an ensemble learning method that can be used for both regression and classification machine learning problems.. The Random Forest algorithm has a number of advantages such as immunity to noisy datasets, unimportant features and mislabeled input data. it produces a highly accurate classifier for many data sets and , it also runs efficiently on large databases and it gives estimates of what variables are important in the classification . RFT is composed of some number of decision trees. Each tree is built as follow: —

-Let the number of training objects be D, and the number of features in features vector be F.

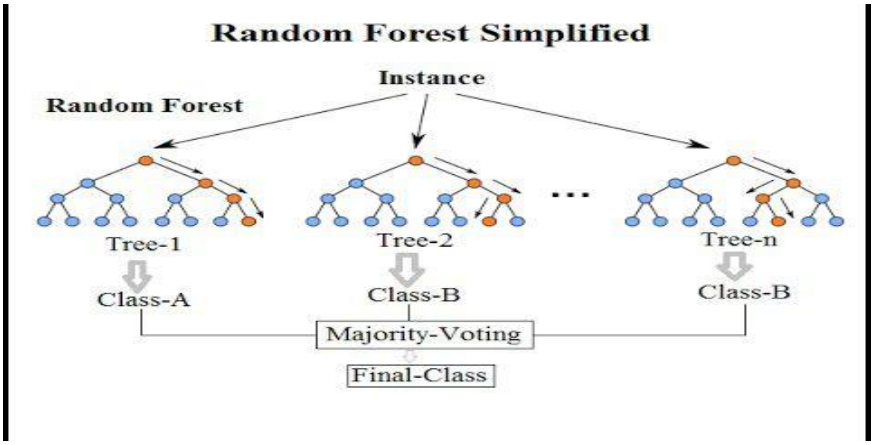
— Training set for each tree is built by choosing D times with replacement from all D available training objects.

— Number f <F is an amount of features in which to base the decision at the node .

These features are randomly chosen for each node.

— Each tree is built to the largest extent possible.

— Each tree gives a classification, which is called voting for that class. The forest chooses the class having the most votes (over all the trees in the forest)



conclusion

we found that RFT performs better. Using the same training and testing set RFT achieves 98% recognition rate compared to 87 % using KNN. On the other hand KNN is very fast in training and testing the datasets compared to RFT. Experimental results also showed that both KNN and RFT are very sensitive to noisy data. In our future works, rough sets-based feature

extraction, rule generation and classification will provide more challenging and may allow us to refine our learning algorithms and/or approaches to the Arabic pattern recognition

Table 2. Results of RFT

Number of trees	10	20	50
Recognition Rate	94%	96.5 %	98 %
Number of features	3	2	3
Training Time	0.77 sec	1.06 sec	3.90 sec
Testing Time	0.01 sec	0.02 sec	0.08 sec

Table 1. Results of KNN

Recognition Rate	87 %
Search Algorithm	KD Tree
Training Time	0.02 sec
Testing Time	0.01 sec

PAPER TWO

DIGIT RECOGNITION WITH FEED FORWARD NEURAL NETWORK

Content

ABSTRACT

The aim of this paper is to design a recognizer to recognize Assamese digits using feed forward neural network. The recognizer crops the individual digits of the image using bounding box method and extracts the feature. In the present study zoning is used to obtain necessary feature vector. This feature is provided as input to the classifier and the network is trained with backpropagation training algorithm with two hidden layer. The recognition rate of printed digits is 98%, including multi size, bold and italics fonts. In case of handwritten digits recognition rate is 70.6%

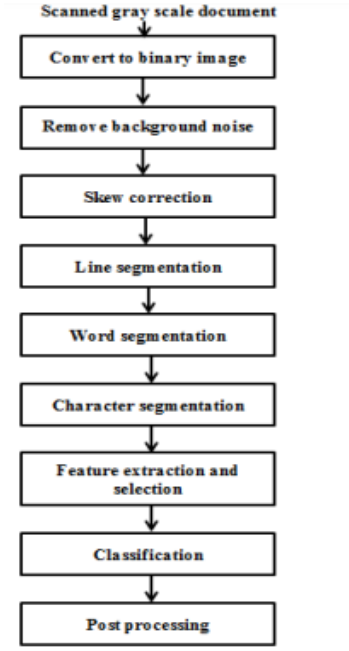
INTRODUCTION

Nowadays one of the most important applications of pattern recognition is to design character recognition system that capable of recognizing human or machine written characters. These systems can produce editable document and thus reduce the human effort. Since the advancement the digitalization age, character recognition by machine remains as an active research field. Although digits are subset of any languages' character sets they are very important to many applications.

In general, there are two categories of character recognition systems; recognize while writing it, called on-line recognition technique or recognize after writing it, called off-line recognition technique. In this paper we discuss about an experiment which uses the latter approach i.e. off-line technique. There are various off-line character recognition systems available which may be specific or generic. The generic systems are capable of processing any type of scanned document, which may contain any font or even graphics also and produce equivalent editable text. Due to its lots of functionality the performance of such type of systems is not always high. The specific recognition systems are used for special tasks only, such as specific font is to be recognized and due to limited font processing accuracy is higher than the generic system when use for specific purpose.

THEORETICAL BACKGROUND

A document is scanned by an optical scanner to produces an image which is not in editable format. In general, character recognition system goes through the following steps as



Block diagram of character recognition system

shown in figure 1.

Noise Removing

Noise can be removed by using linear filtering, medial filtering and adaptive filtering. Morphological openings and closings are also useful for the smoothing of gray-scale images. This technique is used in this recognizer. Here is example of noise removal from binary image by morphologically opening the image in the following figure.

An original binary image a (left) with noise, and image b after noise removal (right).

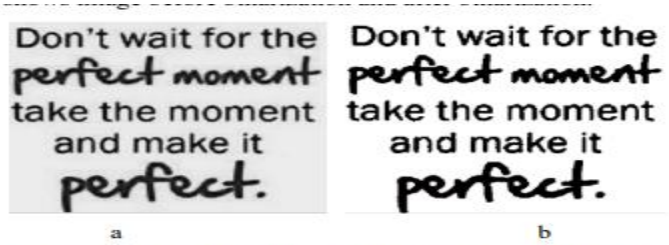
Image Acquisition

The very first task of CR system is to obtain the image patterns. Image may be obtained by scanning document as well as from various standard datasets. In the earlier process, the paper needs to be digitized by scanning it with scanner in good resolution (300dpi or more) and save as any image format. In present study, images are acquired by scanning document in 300dpi. The image obtained through both processes needs to go through lots of preprocessing steps. The basic goal of preprocessing is to produce neat and clean version input image for efficient feature extraction. The following figure shows a portion of scanned image.

Figure 2: A scanned document image

Binarization

It is used to extract text from low quality image background such as poor contrast, non-uniform background, and random noise due to limited sensitivity of sensor. The gray scale images are converted to binary images using appropriate threshold that may be fixed or dynamically calculated from the image.



a b

Figure 3: An original image a (left) and image b after binarization (right).

Skew Detection and Correction

When a document is fed in to the scanner either by man or machine, a few degrees of tilt (skew) is unavoidable. Skew angle is the angle between the lines of text in the digital image and the horizontal direction

Line, Word and Character Segmentation

The segmentation process plays a crucial role in the overall process of recognition of printed and handwritten characters. According to the remark of Marosi a reliable character segmentation method is more important than the recognition performance of the classifier.

Thinning

Thinning is a morphological operation in which a one-pixel width representation of an object is obtained. It successively removes the boundaries of foreground objects as defined, without affecting the connectedness of pixels or the ends of lines.

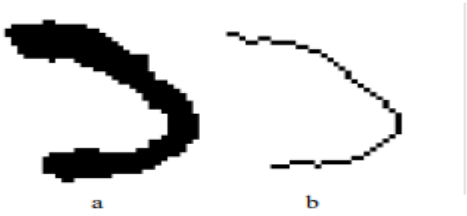


Figure 6: An image (left) before thinning, and (right) after thinning.

Feature Extraction and Selection

Feature extraction is process of simplifying the amount of resources that requires for describing a large set of data accurately. It is another crucial step to design OCR systems with good performance.

Classification

Classification is a process to categories similar object. It is one of the most important components of OCR system. A number of classification methods are used; such as neural networks classifier, tree based classifier, Hidden Markova Model (HMM) based classifier etc. Artificial neural networks can handle non-convex decisions. Multi-layer feed forward neural

network that trained with a back-propagation learning algorithm is the most popular neural network

Post-Processing

It improves the accuracy of system. The outputs of the character recognition classifier unit may have several character candidates. The post processing helps to determine the best combination of characters out of candidate characters

Feed forward Neural Network Classifier

Feed forward neural network is a special type of ANN which can be used for classification of pattern image. It consists of a number of layers 1, 2 … L, where each layer contains many neurons (or nodes). Every neuron in layer l receives input from all the neurons of layer l-1and propagates the output of every neuron of layer l to layer l+1. The layer 1 is called input layer which receives the feature vector of pattern image and the layer L is called output layer which gives the result of classification. All the layers between the two layers are called hidden layers. The input layer consist of as many neurons as the length of input column vector, while number of neurons in output layer is equals to number of pattern class. In this network there is no feedback link from higher layer to lower layers and this is why it is called feed forward neural networks. Each of the links may have a different weight, which indicates the knowledge of a network. Data fed into the input layer as a column vector propagate through the entire hidden layers until it arrives at the outputs layer. An input to the network may be either a raw image or some specific features that are extracted from the pattern image. Features extraction is a crucial phase and it is application dependent. Usually the weights on links are assigned in random way in the beginning Here is a diagram of simple feed forward neural network with one hidden layer

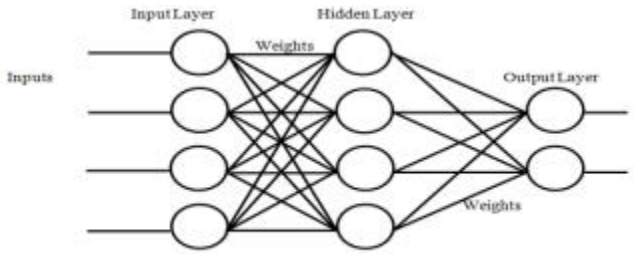


Figure 7: Block diagram of a feed-forward neural network

The hidden layer is the actual processing layer, where a transfer (or activation) functions called sigmoid calculates a layer's output from its net input. There are two sigmoid functions namely log-sigmoid and tan-sigmoid, which are used for pattern recognition. The log-sigmoid function generates outputs between 0 and 1 as the neuron's net input goes from negative to positive infinity.

The output of mathematical calculation by using sigmoid function at every neuron can be defined as follows.

$$ajm = 1\,1+e-Sj\,m \quad (1)$$

Where

$$Sjm = n\,x=0\ wij\,x\,aix \quad (2)$$

Where

ajm *is the activation code of receiving neuron m in layer j,*

Sj is the sum of the products of the activations of all relevant "emitting" nodes (i.e., the nodes in the preceding layer i) by their respective weights, and

wij is the set of all weights between layers i and j that are associated with vectors that feed into neuron m of layer j.

CONCLUSION

In this experiment a feed forward neural network with back propagation is used for the classification of the isolated Assamese numerals. Digits are scanned from document. After preprocessing image are cropped with bounding box to obtain individual digits. Features are extracted from the each zone of the image. The recognition of this proposed system is 98% for printed digits and 70.6% for handwritten digits, which is not close to commercial application. In order to improve the performance of this system, features extraction technique has to be improved and/ or better preprocessing techniques have to be implemented.

PAPER THREE

PATTERN RECOGNITION IN AUTOMATED DIGIT RECOGNITION

Content

Abstract

This paper will take us through pattern recognition in machine learning and explains how it works. It also discusses how it has been applied in various fields and analyzes its future outlook.

Introduction

Pattern recognition is the use of computer algorithms to recognize data regularities and patterns. This type of recognition can be done on various input types, such as biometric recognition, colors, image recognition, and facial recognition. It has been applied in various fields such as image analysis, computer vision, healthcare, and seismic analysis.

What's pattern recognition?

Pattern recognition is the use of machine learning algorithms to identify patterns. It classifies data based on statistical information or knowledge gained from patterns and their representation.

In this technique, labeled training data is used to train pattern recognition systems. A label is attached to a specific input value that is used to produce a pattern-based output. In the absence of labeled data, other computer algorithms may be employed to find unknown patterns.

Pattern recognition features

- *Great precision in recognizing patterns.*
- *Recognition of unfamiliar objects.*
- *Recognition of objects accurately from various angles.*
- *Recovery of patterns in instances of missing data.*
- *Recovery of partially hidden patterns.*

How pattern recognition works

Pattern recognition is achieved by utilizing the concept of learning. Learning enables the pattern recognition system to be trained and to become adaptable to provide more accurate results. A section of the dataset is used for training the system while the rest is used for testing it.

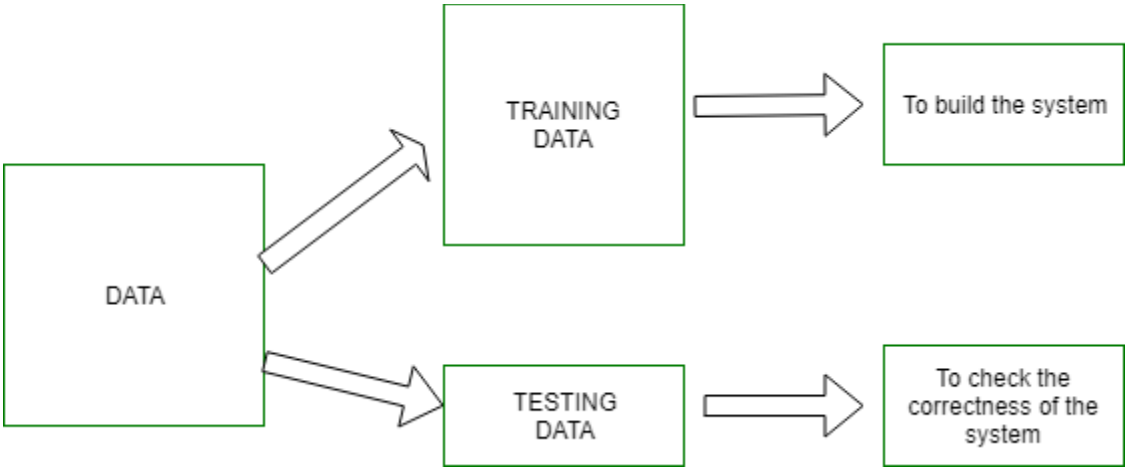


Figure 3image shows how data is used for training and testing.

1. *The training set contains images or data used for training or building the model. Training rules are used to provide the criteria for output decisions.*

2. *Training algorithms are used to match a given input data with a corresponding output decision. The algorithms and rules are then applied to facilitate training. The system uses the information collected from the data to generate results.*

3. *The testing set is used to validate the accuracy of the system. The testing data is used to check whether the accurate output is attained after the system has been trained. This data represents approximately 20% of the entire data in the pattern recognition system.*

Pattern recognition phases

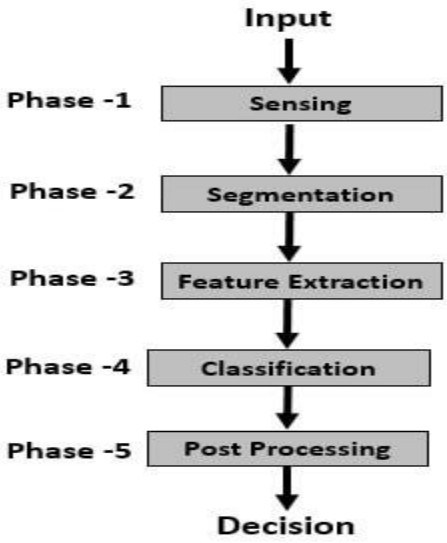
1. Sensing: In this phase, the pattern recognition system converts the input data into analogous data.

2. Segmentation: This phase ensures that the sensed objects are isolated.

3. Feature extraction: This phase computes the features or properties of the objects and sends them for further classification.

4. Classification: In this phase, the sensed objects are categorized or placed in groups or cases.

5. Post-processing: Here, further considerations are made before a decision is made.



Pattern recognition algorithms

Statistical

Algorithm used to build a statistical model. This is a model whose patterns are described using features. The model can predict the probabilistic nature of patterns. The chosen features are used to form clusters. The probability distribution of the pattern is analyzed and the system adapts accordingly. The patterns are subjected to further processing. The model then applies testing patterns to identify patterns.

Structural algorithms

Algorithms are effective when the pattern recognition process is complex. They are important when multi-dimensional entities are used. Patterns are classified into subclasses, thus forming a hierarchical structure. The structural model defines the relationship between elements in the system.

Neural network-based algorithms

These algorithms form a model that consists of parallel structures (neurons). This model is more competent than other pattern recognition models because of its superior learning abilities. A good example of a neural network used in pattern recognition is the Feed-Forward Backpropagation neural network (FFBPNN).

Template matching algorithms

Algorithms are used to build a template matching model, which is a simple pattern recognition model. The model uses two images to establish similarity and the matched pattern is stored in the form of templates. The disadvantage of this model is that it is not efficient in the recognition of distorted patterns.

Fuzzy-based algorithms

Fuzzy-based algorithms apply the concept of fuzzy logic, which utilizes truth values between 0 and 1. In a fuzzy model, some rules may be applied to match a given input with the corresponding output. This model produces good results because it is suited for uncertain domains.

Hybrid algorithms

Hybrid algorithms are used to build a hybrid model, which uses multiple classifiers to recognize patterns. Every specific classifier undergoes training based on feature spaces. A set of combiners and classifiers are used to derive the conclusion. A decision function is used to decide the accuracy of classifiers.

Pattern recognition usage in automated digit recognition

Pattern recognition is used in digital image analysis to automatically study images to gather meaningful information from them. It gives machines the recognition intelligence needed for image processing. Images are needed to be labeled so that the intelligent agent we've programmed can learn from the set of images we've gathered (dataset) and then apply what it has learned by recognizing data from different sets of images (datasets). This can be done in a very accurate manner.

Figure 4 Sample for handwritten Arabic digits dataset

Other applications

- Seismic analysis

  Seismic analysis involves studying how natural events like earthquakes affect rocks, buildings, and soils. Pattern recognition is used for discovering and interpreting patterns in seismic events.

- Healthcare

  Pattern recognition is used in the healthcare sector to improve health services. Data of patients is stored and used by medical practitioners for further analysis. This technique is also used to recognize objects or damages in human bodies.

- Fingerprint identification

  This process is used for identifying fingerprints in computer and smartphone devices. Modern smartphones have a fingerprint identification feature that allows you to gain access to your phone after verifying your fingerprint.

- Computer vision

  It is used in computer applications to extract useful features from image samples. It has been applied in computer vision to perform various tasks such as object recognition and medical imaging.

Conclusion

Pattern recognition is an important technique that enhances the recognition of data regularities and patterns. Pattern recognition has the potential to evolve into a more intelligent process that supports various digital technologies. This technique can be a source of advancements in robotics and automation, especially in the improvement of how humanoid robots are trained.

PAPER FOUR

INTRO TO RANDOM FORESTS

Content

Abstract

A growing interest has been shown in recent years for Multiple Classifier Systems and particularly for Bagging, Boosting and Random Sub- spaces. Those methods aim at inducing an ensemble of classifiers by producing diversity at different levels. In this paper, we'll describe Random Forest principles and review some methods proposed in the literature.

Introduction

Random Forest consists of several independent decision trees arranged in a forest. A majority vote over all trees leads to the final decision. When using Random Forest for detection vast number of negative examples is needed to achieve a robust classifier and a low false positive rate. That leads to a strong inequality between positive and negative class resulting in a Random Forest that focuses on the majority class.

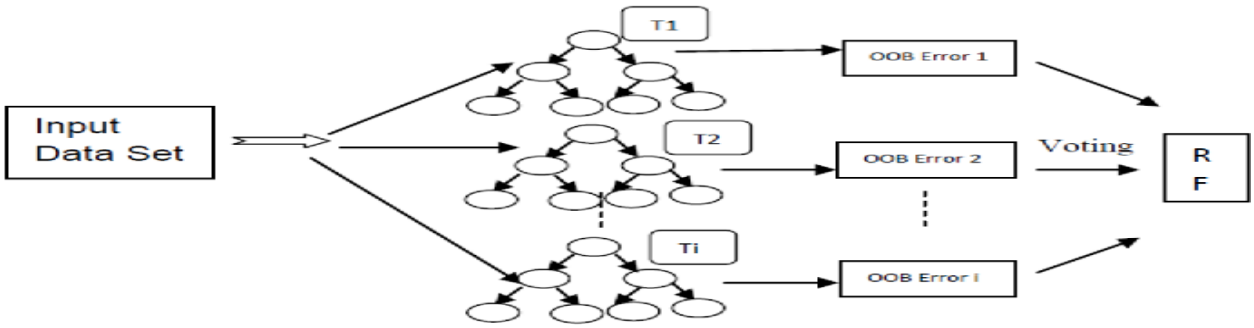


Fig 3: Random forest working methodology

What's random forest

A random forest is a supervised machine learning algorithm that is constructed from decision tree algorithms. This algorithm is applied in various industries such as banking and e-commerce to predict behavior and outcomes.

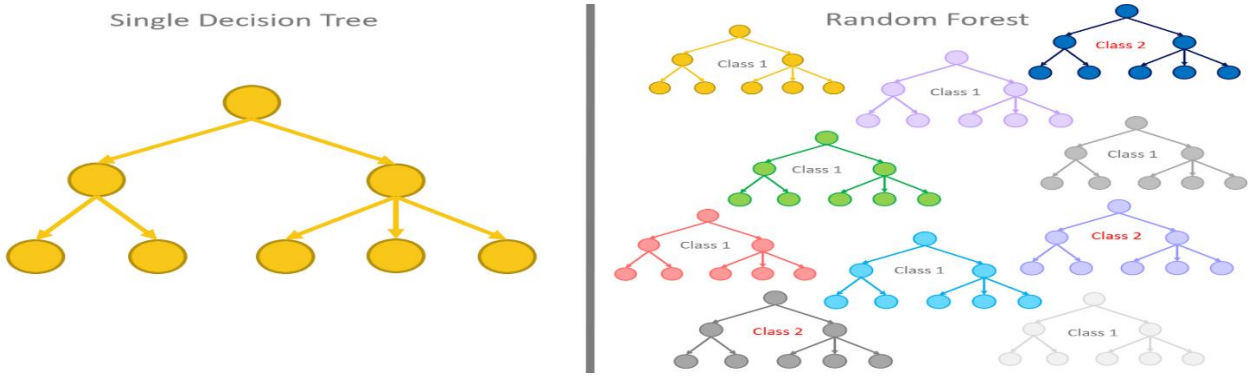
The (random forest) algorithm establishes the outcome based on the predictions of the decision trees. It predicts by taking the average or mean of the output from various trees. Increasing the number of trees increases the precision of the outcome.

A random forest eradicates the limitations of a decision tree algorithm. It reduces the overfitting of datasets and increases precision. It generates predictions without requiring many configurations in packages (like scikit-learn).

The reason why this algorithm is called random is that it offers extra randomness during the creation of the tree structure. When splitting a node, instead of looking for the best attribute directly, it looks for the best attribute in a subset of random attributes. This situation creates more diverse trees

Features of random forest

- *It's more accurate than the decision tree algorithm.*
- *It provides an effective way of handling missing data.*
- *It can produce a reasonable prediction without hyper-parameter tuning.*
- *It solves the issue of overfitting in decision trees.*
- *In every random forest tree, a subset of features is selected randomly at the node's splitting point.*



Classification in random forest

Classification in random forests employs an ensemble methodology to attain the outcome. The training data is fed to train various decision trees. This dataset consists of

observations and features that will be selected randomly during the splitting of nodes. A rain forest system relies on various decision trees. Every decision tree consists of decision nodes, leaf nodes, and a root node. The leaf node of each tree is the final output produced by that specific decision tree. The selection of the final output follows the majority-voting system. In this case, the output chosen by the majority of the decision trees becomes the final output of the rain forest system.

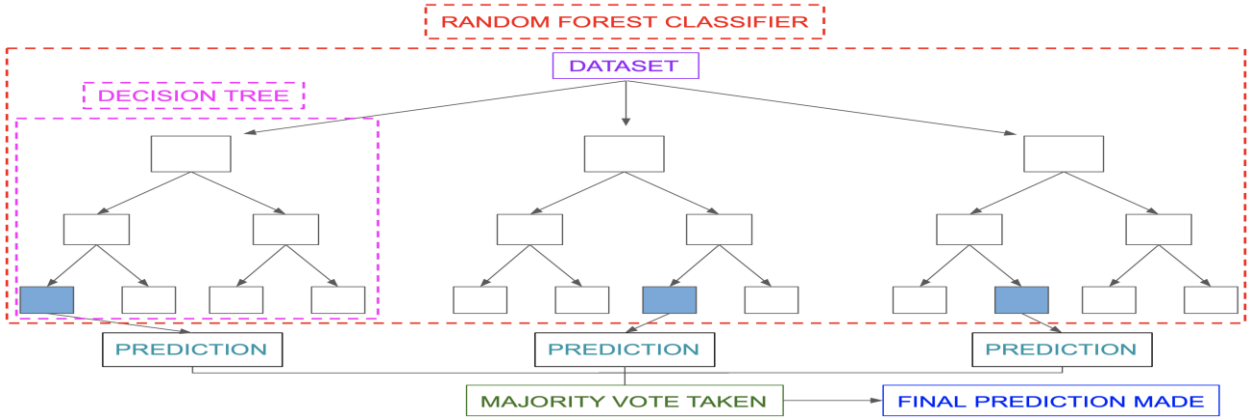


Figure 5 A simple Random Forest classifier

Advantages of random forest

- *It can perform both regression and classification tasks.*
- *A random forest produces good predictions that can be understood easily.*
- *It can handle large datasets efficiently.*
- *The random forest algorithm provides a higher level of accuracy in predicting outcomes over the decision tree algorithm.*
- *The random forest algorithm is not biased, since, there are multiple trees and each tree is trained on a subset of data. Basically, the random forest algorithm relies on the power of "the crowd" therefore, the overall biasedness of the algorithm is reduced.*
- *This algorithm is very stable. Even if a new data point is introduced in the dataset the overall algorithm is not affected much since new data may impact one tree, but it is very hard for it to impact all the trees.*
- *The random forest algorithm works well when you have both categorical and numerical features.*
- *The random forest algorithm also works well when data has missing values or it has not been scaled well (although we have performed feature scaling in this article just for the purpose of demonstration).*

Disadvantages of random forest

- *When using a random forest, more resources are required for computation.*
- *It consumes more time compared to a decision tree algorithm.*

Conclusion

Random forest algorithm is a machine learning algorithm that is flexible and easy to use. It uses ensemble learning, which enables organizations to solve regression and classification problems. This is an ideal algorithm for developers because it solves the problem of overfitting of datasets. It's a very resourceful tool for making accurate predictions needed in strategic decision making in organizations.

PAPER FIVE

REVIEW ON OPTICAL CHARACTER RECOGNITION

Abstract - *Optical Character Recognition is the area of Pattern Recognition that has a topic of studies over the past some decades. Optical character recognition is technique of automatically identifying of different character from a record picture additionally provide full alphanumeric recognition of printed or handwritten characters, text numerical, letters, and symbols in to a computer process able layout including ASCII, Unicode and so forth. Optical character recognition is the bottom for many distinct styles of programs in diverse fields, a lot of which we use in our daily lives. Cost effective and less time consuming, corporations, submit offices, banks, security systems, and even the field of robotics hire this system as the base in their Operations. These days, there are numerous portions of research and making use of OCR technology. These OCR technologies help to examine unique documents written in English, Chinese, Hindu, Arabic, Russian, and others languages. On This paper present review of some researches has been made in English, Arabic and Devanagari characters.*
And explained the methodology they use and challenge they face during development of Optical character recognition.

Key Words: *OCR, optical character recognition, character recognition, handwriting character recognition.*

1. INTRODUCTION

Character recognition, usually abbreviated to optical character recognition or shortened OCR, is the mechanical or electronic translation of images of handwritten, typewritten or printed text (usually captured by a scanner) into machineeditable text [4]. It is a field of research in pattern recognition, artificial intelligence and machine vision. Though academic research in the field continues, the focus on character recognition has shifted to implementation of proven techniques. Optical character recognition technology was invented in the early 1800s, when it was patented as reading aids for the blind. In 1870, C. R. Carey patented an image transmission system using photocells, and in 1890 P.G. Nipkow invented sequential scanning OCR. However, the practical OCR technology used for reading characters was introduced in the early 1950s as a replacement for keypunching system [2]. A year later, D.H. Shephard developed the first commercial OCR for typewritten data. The 1980's saw the emergence of OCR systems intended for use with personal Computers. Nowadays, it is common to find PC-based OCR systems that are commercially available. However, most of these systems are developed to work with Latin-based scripts. Optical character recognition systems for Latin characters have been available for over a decade and perform well on clear typed text. There are research has also been directed at other non-Latin scripts such as Arabic, Japanese, Chinese, Hindu, Tibetan. In order to develop an OCR system it requires the development and integration of many sub systems. The first step is preprocessing such as skew detection and correction, noise detection and removal, binearization, thinning, and normalization. Then segmentation of document images into line, word and characters. This is followed by feature extraction for representing character images and a classification module that label characters to their proper class. Finally, post processing i.e. applying

2. LITERATURE REVIEW

Character recognition technique has been completed through studies on different characters for example, English, Arabic, Chinese, Devanagari, Bangla, Farsi and Kannada and so on. Totally, the complete method is carried out in three phase Preprocessing, Feature extraction and recognition[5]. In this paper only cover the study has been done on English, Arabic and Devanagari scripture.

2.1. Arabic Scripter Character Recognition

In 2002 Majid M. Altuwaijri and Magdy A. Bayoumi They develop system to recognize Arabic text using neural network used set of moment invariants descriptors (under shift, scaling and rotation) and artificial neural network (ANN) used for classification The study has shown 90% of a high accuracy rate [9]. In 2015 Ashraf Abdel Raouf, Colin A. Higgins, Tony Pridmore and Mah-moud I. Khalil Haar studied approach for recognizing Arabic character using Haar Cascade Classifier (HCC) These classifiers were trained and tested on some 2,000 images. To extract feature Haar-like feature extraction used and boosting of a classifier cascade. The system was tested with real text image and produces 87% accuracy rate for Arabic character recognition[10]. In 2017 N. Lamghari, · M. E. H. Charaf and · S. Raghay On this research the data are divided into three parts. From 34,000 character 70% are used for training, 15% for testing phase and 15% for validation. To extract feature hybrid feature extraction used (pixel density, resize, freeman code, structural features, invariant) for recognition used feed forward-back propagation neural network. The system has achieved 98.27% high recognition rate[11]. In 2018 Noor A. Jebrila, Hussein R. Al-Zoubib and Qasem Abu Al-Haijac In addition to the preprocessing step includes in particular three levels. In the primary section, they employed word segmentation to extract characters. In the second one section, Histograms of Oriented Gradient (HOG) are used for feature extraction. The very last phase employed Support Vector Machine (SVM) for classifying characters. They have carried out the proposed method for the recognition of Jordanian metropolis, city, and village names as a case examine, similarly to many other phrases that offers the characters shapes that aren't included with Jordan cites. The set has cautiously been selected to include each Arabic character in its all forms. To the conclusion, they have got constructed their own dataset inclusive of greater than 43.000 handwritten Arabic phrases (30000 used for training and 13000 used for testing stage). Recognition result show 99% rate of accuracy[12].

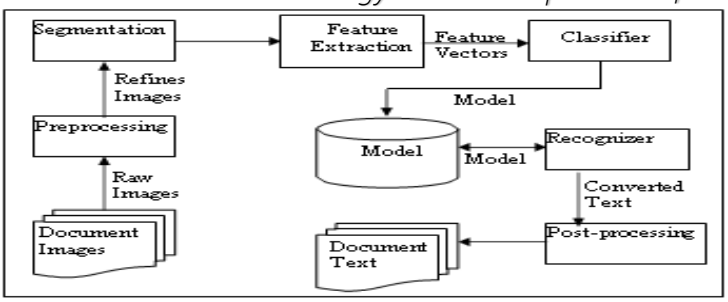
2.2 Devanagari Scripter Character Recognition

In 2011 Gyanendra K. Verma, Shitala Prasad, and Piyush Kumar Curvelet present in approach for Hindi handwritten character recognition using curvelet transformer. The study are used dataset that contain 200 images of character (each image contains all Hindi characters). Feature extract using curvelet transform and for recognition k-nearest neighbor the experiment result show more than 90% accuracy [13]. In
2013 Divakar Yadav, Sonia Sánchez-Cuadrado and Jorge Morato develop optical character recognition system using neural network for Hindi characters and trained with 1000 dataset.

Feature extraction technique is histogram of projection based on mean distance, on pixel values and vertical zero crossing. Then classify using back-propagation neural network with two hidden layers. Experimental result show 98.5% correct recognition[14]. In 2015 Akanksha Gaur and Sunita Yadav this system extract feature using k-means clustering and classified used support vector machine using linear kernel and Euclidean distance. The evaluation show that SVM has better results using linear kernel than Euclidean distance. Maximum achieved using Euclidean distance is 81.7% accuracy. Using linear kernel giving 95.86% result[5]. In 2018 Nikita Singh present system with the title "An Efficient Approach for Handwritten Devanagari Character Recognition Based on Artificial Neural Network" for recognition hind character. For feature extraction they used histogram oriented gradients (HOG) and recognition used artificial neural network (ANN) classifier. The system get 97.06% high accuracy [15].

3. MAJOR STEPS INVOLVE IN CHARACTER RECOGNITION

Building an OCR engine is not an easy thing to do as the main difficulty lies with – identifying each character and word. For making an OCR engine from scratch below are the steps which one can follow to make sure that the OCR meets the desired expectation of character recognition and this is the methodology and the steps most of researchers used.



3.1 Optical Scanning

To start with an OCR, image can be capture by digital camera also but after seeing the challenge been faced in privies work better to use scanner therefor consider first need putting together a good optical scanner. With the help of this scanner, an image of original file or document is captured. It is commanding to select scanner with a good sensing tool and transport mechanism.

3.2 Pre-processing:

Preprocessing is performing different operation on the scanned or input image. It helps to remove noise from image make character clear and It basically enhances the image rendering suitable for segmentation. Preprocessing has various task are such as converting gray scale, binarization, thinning, skewing and normalization.

3.3 Segmentation:

Once the preprocessing produces noise free clean character image, it's then segmented into several subcomponents. There are three steps of segmentation first line segmentation divide the character in image horizontal second word segmentation the divide words from line sentence last character segmentation divide the characters from word. Finally we get segmented characters those character help for feature extraction and recognition.

3.4 Feature extraction:

This is one of the riskiest components in an OCR development. The main aim is to extract important patter
from characteristics. The selected features are expected to contain pattern that differentiate one character from other and relevant information from the input data, so that the classification can be performed by using those patter extract from segmented character this instead of the complete original data.

3.5 Training and recognition:

Investigation of OCR's pattern recognition can be done via template matching, statistical technique, syntactic or structural techniques, and artificial neural networks. The system also have to be learn in such a way that the problem associated to incomplete vocabulary is solved.

3.6 Post-processing:

In this final process, activities like grouping, error detection and correction take place. During grouping, symbols in the text are associated with strings. However, it's impossible to reach 100% accurate identification of characters, only some of the errors can be detected and deleted as per the context.

4. CHALLENGES OF OPTICAL CHARACTER RECOGNITION

For better and high character recognition accuracy there are so many OCR techniques but still difficult to achieve 100% correct recognition especially for character that has similarity. The challenges I observe during review is many of them related to the data collection and preprocessing if we can identify and rid of those challenges we can get high correct recognition. The following issues created due to collecting input data using digitals camera. Instead of using camera to capture characters or scripts prefer to scan the document but let's see what those challenges are.

4.1 Scene Complexity

Input data taken with camera may have other object is also for example building, homes, panting and other objects to separate those objects from text or character is very tough. The data that content non textual contents make preprocessing difficult there for affect the character recognition process.

4.2 Conditions of Uneven Lighting

Many times image taken from road or outdoor affected by light and shadows. This is another challenge for optical character recognition. It make difficult to detect and segment characters. This kind of issues makes scanning document more preferable than capturing it by camera. Camera light flash also may help for additional lighting and create shadows in images.

4.3 Skewness (Rotation)

Image taking using camera also disturb by this issue. The angle of the image incorrect therefor when we fed this data to optical character recognition system the outcome will be incorrect. But there are techniques to solve this problem like Fourier transformer, projection profile, Hough transform and so on.

4.4 Blurring and Degradation

This also caused by image taken with camera. This happen when images are taken from distance, trying to capture on movements and Lack of focusing. Image taking on this and other circumstance face blurring and degradation. For segmentation and accurate recognition sharpness of characters is needed.

4.5 Fonts and style

Characters that are connected each other like Arabic, Hindi and fonts style like Italic and other overlap each other this make difficult for optical character recognition system during segmentation process hard to detect and divide words in to character.

4.6 Multilingual Environments

Characters that have multi environment such as, language that has large number of character Ethiopian, Korean, Chinese, Japanese and other. Characters that written connectedly with each other Arabic language. Ethiopian language Amharic alphabet similarity of characters it's difficult for computer to see the difference between most of them. Therefor this kind of multi environmental characters are challenges for OCR to divide and extract individual characters and recognize correctly.

4.7 Damage documents

When the input document are very old and damage whether we take it in camera or scanned will be very difficult to observe the character, content many noise when we try to remove those noise sometime the data or image lose it necessary content or characters.

5. CONCLUSION

In the research works revised in this paper, character recognition system use different approaches and many of them get good accuracy. What we can understand from this paper is feature extraction techniques should be choose according to the character you working because each scripts or alphabets has its own nature therefor need to find techniques which fit or suitable for characters. The better able to extract features from character more we can detect and recognize characters in highest accuracy.

ACKNOWLEDGMENT

First and foremost I thank my supervisor Dr. Ali Imam Abidi for his guidance in my research work. He gave me the best advice and helping me with provide me necessary document for my researches" Review on Optical Character Recognition" ,Who also helped me in the survey of the related work of different authors .I came to know about so many new things I am really thankful to them for his care and support.

REFERENCES:

[1] J. Cowell and F. Hussain, "Amharic character recognition using a fast signature based algorithm," *Proc. Int. Conf. Inf. Vis.*, vol. 2003–Janua, pp. 384–389, 2003.
[2] I. Stoianov, "Optical Character Recognition of Historical Documents," *Clover.Slavic.Pitt.Edu*, 1995.
[3] V. Patil and S. Shimpi, "Handwritten English character recognition using neural network," *Elixir Comp. Sci. Engg*, vol. 41, no. 3, pp. 5587–5591, 2011.
[4] K. A. Okrah, "Nyansapo (the wisdom knot): Toward an African philosophy of education," *Nyansapo (The Wisdom Knot) Towar. an African Philos. Educ.*, no. 224, pp. 1–121, 2003.
[5] A. Gaur and S. Yadav, "Handwritten Hindi character recognition using k-means clustering and SVM," *2015 4th Int. Symp. Emerg. Trends Technol. Libr. Inf. Serv. ETTLIS 2015 - Proc.*, pp. 65–70, 2015.
[6] N. M. Noor, M. Razaz, and P. Manley-Cooke, "Global geometry extraction for fuzzy logic based handwritten character recognition," *Proc. - Int. Conf. Pattern Recognit.*, vol. 2, pp. 513–516, 2004.
[7] D. Nasien, H. Haron, and S. S. Yuhaniz, "Support Vector Machine (SVM) for english handwritten character recognition," *2010 2nd Int. Conf. Comput. Eng. Appl. ICCEA 2010*, vol. 1, pp. 249–252, 2010.
[8] M. S. Sonawane and C. A. Dhawale, "Evaluation of Character Recognisers: Artificial Neural Network and Nearest Neighbour Approach," *2015 IEEE Int. Conf. Comput. Intell. Commun. Technol.*, pp. 129–132, 2015.
[9] M. M. Altuwaijri and M. A. Bayoumi, "Arabic text recognition using neural networks," pp. 415–418, 2002.
[10] A. AbdelRaouf, C. A. Higgins, T. Pridmore, and M. I. Khalil, "Arabic character recognition using a Haar cascade classifier approach (HCC)," *Pattern Anal. Appl.*, vol. 19, no. 2, pp. 411–426, 2016.
[11] N. Lamghari, M. E. H. Charaf, and S. Raghay, "Hybrid Feature Vector for the Recognition of Arabic Handwritten Characters Using Feed-Forward Neural Network," *Arab. J. Sci. Eng.*, vol. 43, no. 12, pp. 7031–7039, 2018.
[12] N. A. Jebril, H. R. Al-Zoubi, and Q. Abu Al-Haija, "Recognition of Handwritten Arabic Characters using Histograms of Oriented Gradient (HOG)," *Pattern Recognit. Image Anal.*, vol. 28, no. 2, pp. 321–345, 2018.
[13] R. Rani, R. Dhir, and G. S. Lehal, "Information Systems for Indian Languages," *Commun. Comput. Inf. Sci.*, vol. 139, no. January 2016, pp. 174–179, 2011.

5/THE DATASET EMPLOYED

the data set used in our project consists of 60k training and 10 k testing images

the images are given as pixels and as 4 files

2 files for training (pixels -labels) 60 k each

2 files for testing (pixels -labels) 10 k each

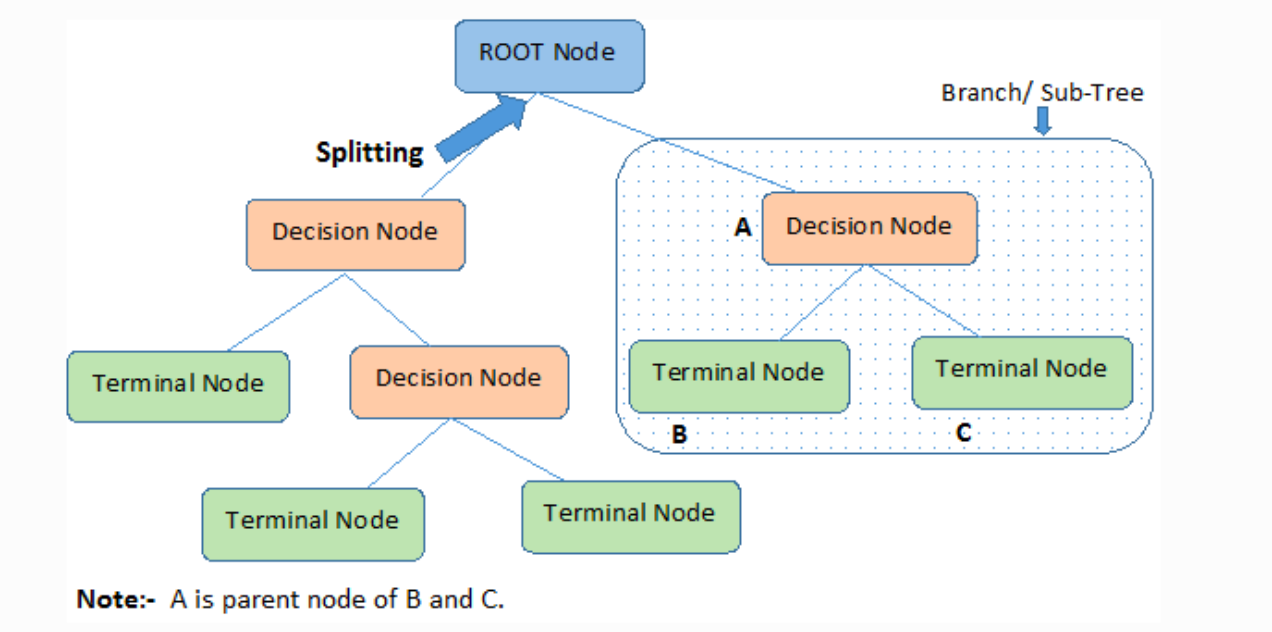
Link: Arabic Handwritten Digits Dataset | Kaggle: https://www.kaggle.com/mloey1/ahdd1

6/DETAILS OF THE ALGORITHM(S)/APPROACH(ES) USED AND THE RESULTS OF THE EXPERIMENTS

-DECISION TREES

A hierarchical data structure that represents data by implementing a divide and conquer strategy. Decision Trees are a type of Supervised Machine Learning where the data is continuously split according to a certain parameter. The tree can be explained by two entities, namely decision nodes and leaves. The leaves are the decisions or the final outcomes. And the decision nodes are where the data is split.

The approach most used for solving the classification and regression problems. The main objective is creating the training model which basically predicts the class or targeted values of learning decision rules that inferred from training and past data. The core object of the model is creating a small tree that can classify the unknown class or instance by determining the list of the rules the handwritten digits dataset. Multiple different tools are used by DT classifier to extract information, digit recognition, text predictive and machine learning.



-RANDOM FORESTS

A random forest is a supervised machine learning algorithm that consists of many decision trees. The 'forest' generated by the random forest algorithm is trained through bagging.

Bagging is an ensemble meta-algorithm that improves the accuracy of machine learning algorithms. It utilizes ensemble learning, which is a technique that combines many classifiers to provide solutions to complex problems.

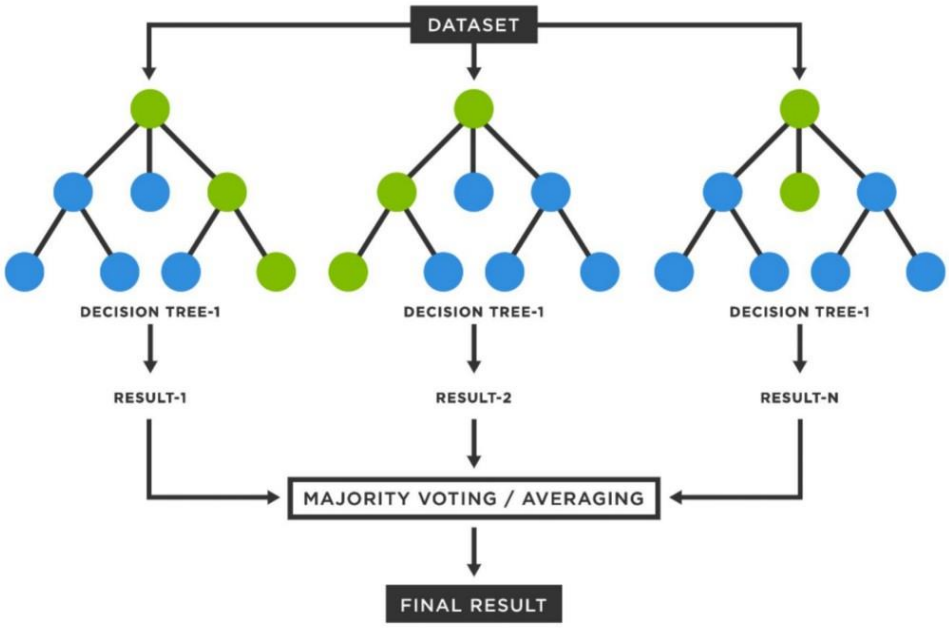
The random forest algorithm establishes the outcome based on the predictions of the decision trees. It predicts by taking the average or mean of the output from various trees. Increasing the number of trees increases the precision of the outcome.

The steps used for making the random forest are:

First, start with the selection of random samples from a given dataset.

Second, the algorithm will construct a decision tree for every sample. Then it will get the prediction result from every decision tree

Third, *the algorithm will use the vote to decide the prediction for the image. Final, take the most numbers of votes as the result for the image prediction.*



Features of a Random Forest Algorithm

- *It's more accurate than the decision tree algorithm.*
- *It provides an effective way of handling missing data.*
- *It can produce a reasonable prediction without hyper-parameter tuning.*
- *It solves the issue of overfitting in decision trees.*
- *In every random forest tree, a subset of features is selected randomly at the node's splitting point.*

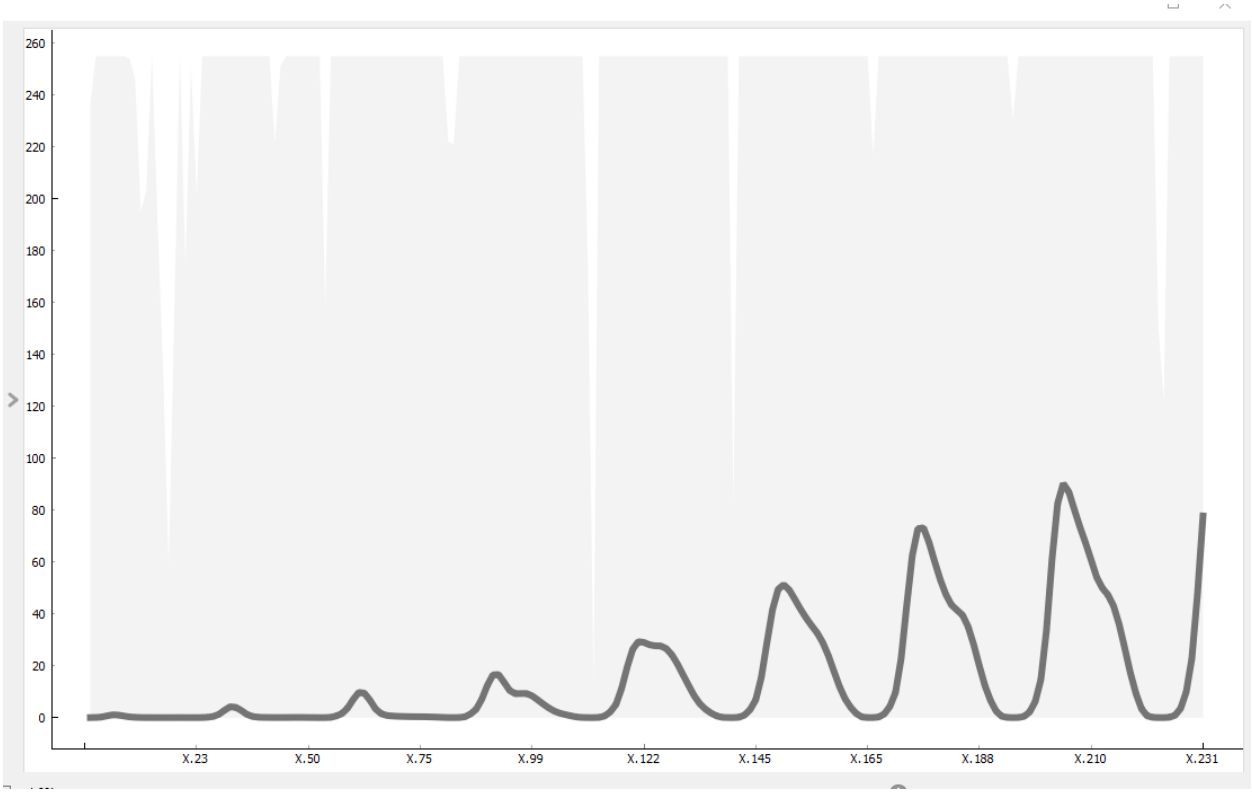
7/ DEVELOPMENT PLATFORM

Used libraries:

```
import numpy as np
import matplotlib.pyplot as pt
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score
from sklearn.model_selection import GridSearchCV
from PIL import Image,ImageOps,ImageGrab
import cv2 as cv
import tkinter as tk
from tkinter import *
from win32 import win32gui
from sklearn.tree import DecisionTreeClassifier
from sklearn import metrics
```

Programming language used: python

Tools- Enviroment : JetBrains PyCharm Community Edition 2019.2.3


260
240
220
200
180
160
140
120
100
80
60
40
20
0
X.23
X.50
X.75
X.99
X.122
X.145
X.165
X.188
X.210
X.231