CHAPTER-3

DATA COLLECTION AND FRAMEWORK FOR GENDER CLASSIFICATION AND WRITER IDENTIFICATION SYSTEM

Gender classification and writer identification systems are the two prevailing, stimulating, and motivational perspectives of handwriting based research developments. Data collection and generation of corpus have always been the cornerstone for the development of the system. In this chapter, the novel contribution, sample collection process and the generation of corpus have been presented in section 3.1. The collection of data samples from the male and female writers has also been presented with images. In section 3.2, the framework for the proposed systems has been elaborated along with the methodologies required for implementing the phases. Pre-processing methods, implementation of feature extraction techniques, hybridization of feature extraction and classification techniques and evaluating performance metrics have been discussed in detail, in the sub-sections followed by section 3.3, that presents the chapter summary at the end of the chapter.

3.1 CORPUS GENERATION

The generation of a dataset involves collecting, analyzing, measuring, and validating the data using standard techniques. Data collection must be done with great care as it is the principal and salient phase in every novel research development. The format of data collection may vary as per the nature of research and application to be developed, but the collection, storage, and maintenance of the dataset is the mandated responsibility of the researcher. It is the prime target of every researcher to maintain the eminence of the data samples because the quality of the sample is directly proportional to the accuracy of the experiments performed (Singh and Sachan, 2015). If the quality of data varies or degrades, then it will surely contradict the performance of the system. For the successful and efficient corpus generation, numerous factors to be considered are the size of the corpus, choice of the script and participants, number of participants, sampling and selection, manner of input, approach to cleaning are the necessary phases that require extreme attention of the researcher. Fogel *et al.* (2020) developed scrabbleGAN, a semi-supervised approach that can generate images of words to boost up the accuracy. Dash and Ramamoorthy (2019) presented the method

of generating a corpus and overcoming the pitfalls faced during the generation of the corpus and emphasized the linguistic and statistical factors and issues to control the generation process.

3.1.1 Generating Corpus for Gender Classification System

In this section, the method of generating the dataset for the gender classification system is clearly described. Corpus has been generated by collecting offline handwritten samples of the writers in the Gurumukhi script followed by the Pre-processing activities such as binarization, normalization, and thinning which are explained in the next section. For our proposed work, data samples of 200 writers with 100 female writers and 100 male writers have been collected. Each writer has written 10 copies of 35 primary characters of the Gurumukhi script. Therefore, in total, we have 100×35×10= 35,000 Gurumukhi samples collected from 100 female writers and 35,000 samples collected from 100 male writers. For the output, we have only two classes for gender classification i.e., male and female class. Table 3.1 shows a detailed view of the dataset for the gender classification system.

Table 3.1. Dataset Description for Gender Classification System

Nature of System	**Number of Classes**	**Number of Writers**	**Number of documents written by each writer**	**Number of Samples**
Gender Classification System	2	100 male writers	10	35000 female Gurumukhi characters
		100 female writers		35000 male Gurumukhi characters

3.1.2 Generating Corpus for Writer Identification System

For the development of the writer identification system, we have taken samples of 200 writers, similar to the gender classification process and every writer has written 10 times each primary character (total 35) of the Gurumukhi script, so in total, we have 70000 samples with 200 classes. Table 3.2 shows the dataset description for a writer identification system.

Table 3.2. Dataset Description for Writer Identification System

Nature of System	Number of Classes	Number of Writers	Number of documents written by each writer	Number of Samples
Writer Identification System	200	100 male writers and 100 female writers	10	70,000 Gurumukhi characters

3.1.3 Samples of Offline Male and Female Data

In this section, we are presenting samples of offline data collected from the female and male writers, collected at the commencing stage. Figure 3.1 shows the data collected from a male writer. Figure 3.2 shows the sample of female writer. Samples of data were collected on paper with the help of a pen, followed by the scanning of the document at 300 dpi. Then on the scanned document, Pre-processing, feature extraction and classification phases are the next incredible phases to execute.

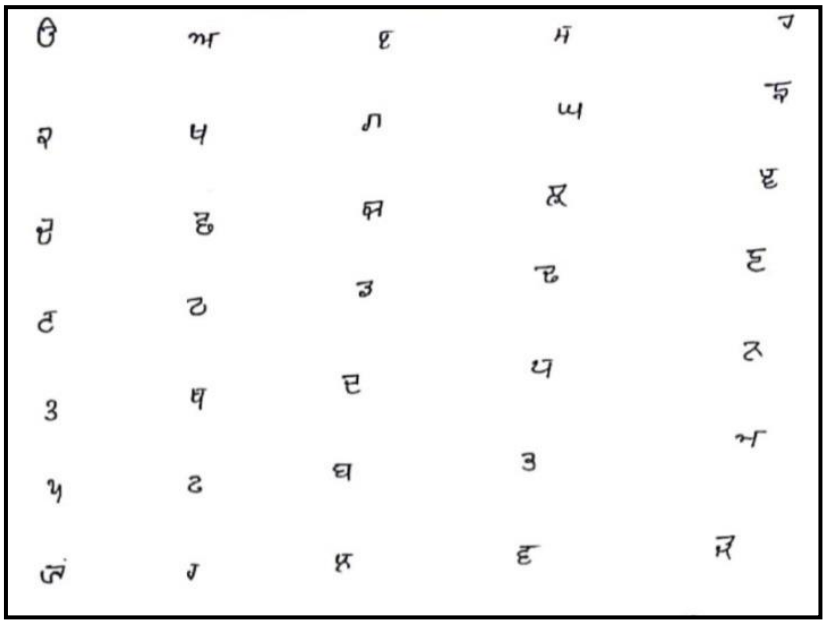


Figure 3.1. Offline Handwritten Sample of a male writer

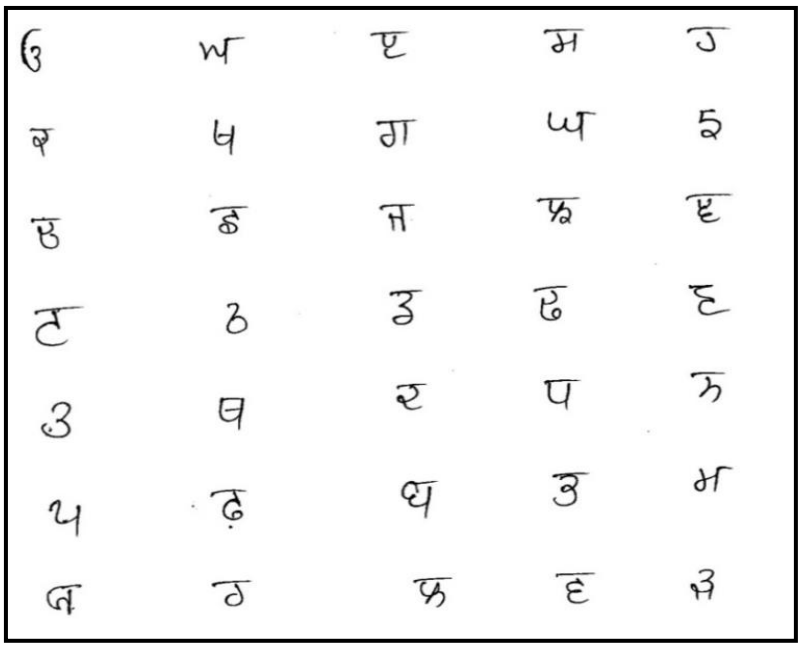


Figure 3.2. Offline Handwritten Sample of a female writer

3.1.4 Novel Contribution and Achievement

The development of gender classification system in the Gurumukhi script is a novel endeavor and achievement for the handwriting based researchers and to create a corpus for the gender classification for a huge dataset i.e. for 200 writers with 100 female writers and 100 male writers, with 70, 000 Gurumukhi characters, is again an accomplishment, as the benchmark dataset is not publicly available in the Gurumukhi script for gender classification and writer identification with a sufficiently large number of samples.

For the development of writer identification system, storing data with the name of 200 writers with 10 copies of each writer is obviously a tedious, challenging, and useful activity in the Gurumukhi script which has not been available. Therefore, storing the dataset of 200 writers in the form of 200 classes for writer identification and two classes for gender classification is the real attempt of this research work.

3.2 FRAMEWORK FOR GENDER CLASSIFICATION AND WRITER IDENTIFICATION SYSTEM

The framework for both gender classification and writer identification provides the establishment for developing the system. It facilitates us with the standard structure to achieve the target and consisting of directional flow with the phases. The framework of gender classification and writer identification will share many common phases but

the major difference lies in the classification and data maintenance phase. Therefore, it depends upon the nature of the dataset i.e., two classes for gender classification and 200 classes for writer identification system. Figure 3.3 presents the detailed framework with all phases such as data collection and digitization, Pre-processing which further includes normalization and thinning, followed by feature extraction and classification phase.

Dimensionality reduction and hybridization of various techniques are also used to optimize the outcomes attained by the proposed system. In this proposed work, we are going to implement hybridization of feature extraction techniques, dimensionality reduction, and classification methods along with hybridization of classifiers. For dimensionality reduction, one can have many options such as Principal Component Analysis (PCA), high correlation filter, low covariance matrix, backward feature elimination, and forward feature construction. In our proposed work, the hybridization of classification techniques using, a majority voting scheme has been implemented. So, here for the experiments to be executed properly and efficiently, the study of phases, their concept, working, techniques and algorithms are explained thoroughly.

3.2.1 Digitization

Digitization helps in defining the bit depth and resolution that will further help in preservation, cost, and production flow. Digital files are the master files that contain a facsimile of the original document. Digitized images cannot be altered or enhanced and can be stored for a long time and for different experimental evaluations. It is the art of converting data i.e., text, image, sound in the digital form so that it can be processed by the machine. Digitization will generate a series of numbers and a discrete set of points for further processing. The process of conversion takes place with the help of physical objects such as scanners, cameras, and other electronic devices. For our proposed system, scanning is completely done at 300 dots per inch (dpi) which is considered a standard value.

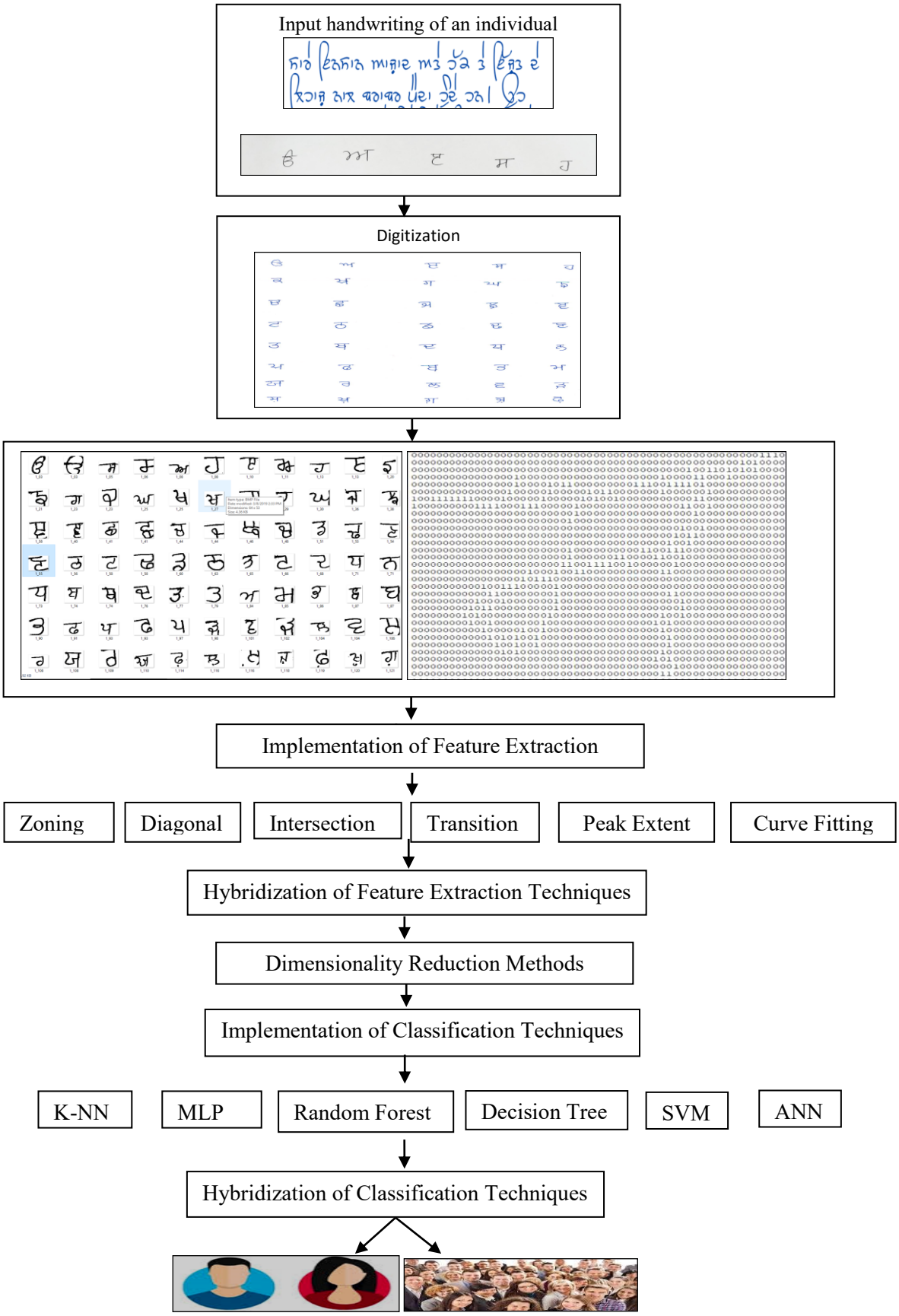
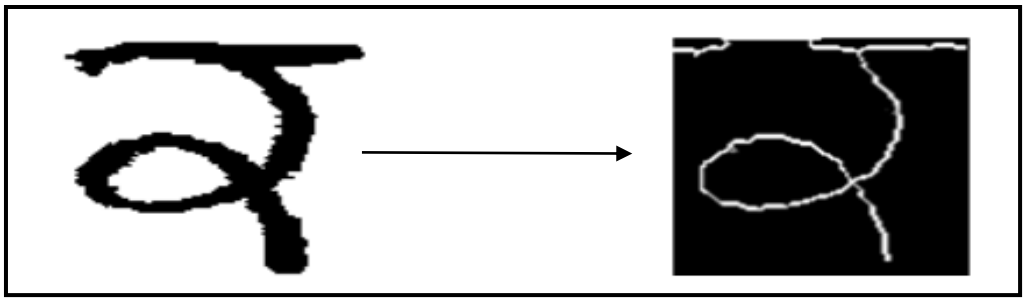


Figure 3.3. Framework for Gender Classification and Writer Identification System

3.2.2 Pre-processing

This is the phase in which scanned data samples are transformed to a format compatible with execution and implementation by the machine as shown in Figure 3.4. Making the data suitable for the algorithmic execution and the extraction of features are the fundamental requirement for the implementation. It is also considered as the process of converting the raw data into an easy and understandable format by machine learning model. Many processes such as slicing, normalization, and thinning are the necessary phases of Pre-processing. In the next subsections, these mandatory sub-phases are discussed with the figures.



Digitized Image Thinned Image

Figure 3.4. Transformation of Digitized to Thinned Image

3.2.2.1 Slicing

In this phase, initially, the threshold value is adjusted for the black and white value to convert it into a binary image followed by the slicing. So, the Gurumukhi characters written by the writers on a sheet have been scanned and then sliced and stored separately in a folder with name for further processing. Because processing will be done on an individual character, therefore this phase will transform a one-page document into individual set of characters, removing blank spaces and stored as per the format as shown in Figure 3.5.

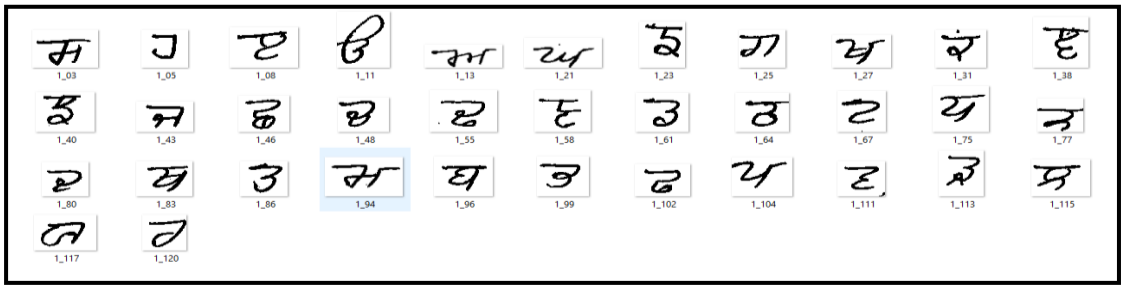


Figure 3.5. Sliced Data

3.2.2.2 Normalization

Normalization is another process of data preparation, which includes altering the values of numeric columns in a dataset to a general common scale without losing the essential information. It is also defined as a scaling method in which values are rescaled and shifted from the original image to, between the range of 0 and 1. In this phase, characters are normalized into a window of size 64×64 with the intention of providing uniformity to the characters or having similar distributions. Next is the conversion of the .gif image to .bit map images with the help of online tools has been performed, will be done in this phase as shown in Figure 3.6.

Figure 3.6. Bitmap images of characters

3.2.2.3 Thinning

This is the process of compressing the data and enhancing the feature extraction process in the subsequent phases. It transforms the width of the character stroke from several pixels to a single-pixel eliminating local noise without introducing distortion of the data. It will reduce the binary image region into lines to maintain the significant features of a pattern. In our experimental works, the parallel thinning algorithm proposed by Zhang and Suen (1984) has been implemented for the thinning process. An example of the thinned image is depicted in Figure 3.7.

Figure 3.7. Sample of a Thinned Image

3.2.3 Feature Extraction

Features are the quantitative measurements that are used to extract the hidden characteristics from the images for recognition (Lee *et al.*, 2013; Morera *et al.*, 2018). This is the process of extracting features from the digitized image of a character using feature extraction techniques, which transforms an image into a vector that is fed for classification. Structural and Statistical methods are the kinds of feature extraction techniques that have been exploited for extracting features. Statistical features deal with the calculation of pixels, discrete patterns based on statistical techniques such as decision and probability theory, transformation method, and filtering method. Structural features are those that can be extracted using relation description, formal grammar, decision tree, etc.

3.2.3.1 Zoning Features

This is one of the most prominent methods of feature extraction that divides the thinned image of a character into a number of equal-sized zones as shown in Figure 3.8. The hierarchical pattern of division is shown in Figure 3.9. Now, the number of foreground pixels in each zone is calculated. These numbers p1, p2, …, pn, obtained for all n zones, are normalized to [0, 1] resulting into a feature set of n elements.

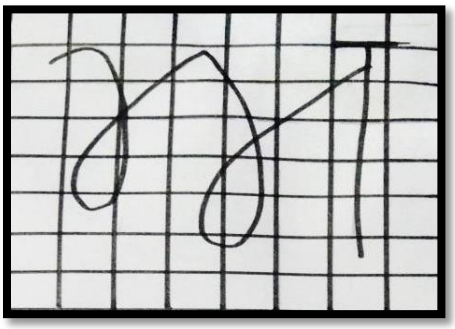


Figure 3.8. Extracting Features with Zoning Method

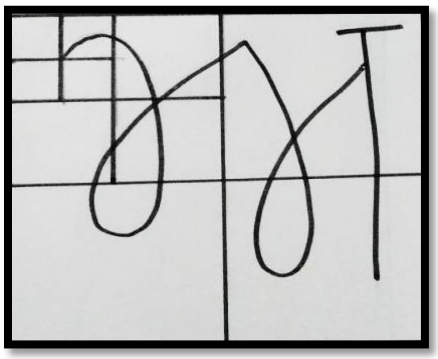


Figure 3.9. Hierarchical Pattern of Division

3.2.3.2 Diagonal Features

This is a method of feature extraction that will count the number of foreground pixels at the diagonals which are generated after implementing the zoning method.

The steps that have been used to extract these features are:

Step I: Divide the thinned image into *n* number of zones.

Step II: Now foreground pixels present along each diagonal are summed up in order to get a single sub-feature.

Step III: These sub-feature values are averaged to form a single value which is then placed in the corresponding zone as its feature.

Step IV: Corresponding to the zones whose diagonals do not have a foreground pixel, the feature value is taken as zero.

These steps will again give a feature set with *n* elements.

Therefore, corresponding to the zoning process, the diagonal method also produces 85 diagonal features for 85 zones as shown in Figure 3.10.

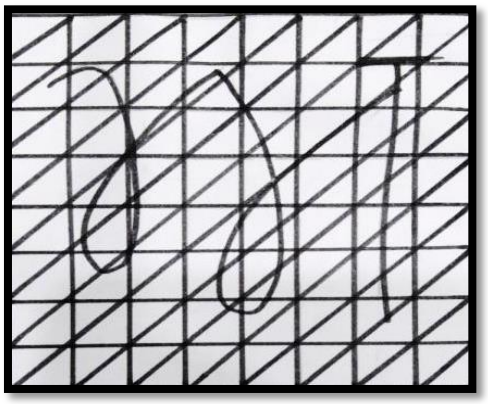


Figure 3.10. Diagonal method of Feature Extraction

3.2.3.3 Transition Features

The transition method of extracting features will calculate the transitions i.e., from foreground to background pixels and background to foreground pixels in both left to right and top to bottom, called horizontal and vertical transition respectively. Again, the transition method generates 2*n* feature values for a character image. Transition

features in horizontal and vertical directions are described in Figure 3.11(a) and 3.11(b).

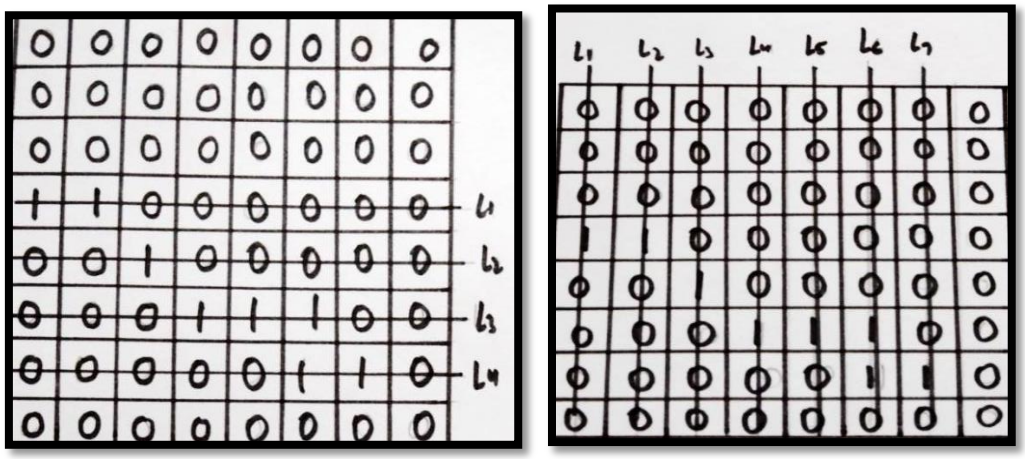


Figure 3.11. (a) Transition in Horizontal Direction (b) Transition in Vertical Direction

Following steps have been followed in the Transition method for extracting features.

Step I: Divide the thinned image of a character into *n* zones.

Step II: Calculate the number of transitions in horizontal and vertical directions for each zone.

3.2.3.4 Peak Extent Based Features

It is a method of feature extraction that will calculate the peak extents i.e., continuous 1's in the horizontal and in the vertical direction both, and that will be equal to the aggregate of successive foreground pixels, replacing with the peak extent as shown in Figure 3.12.

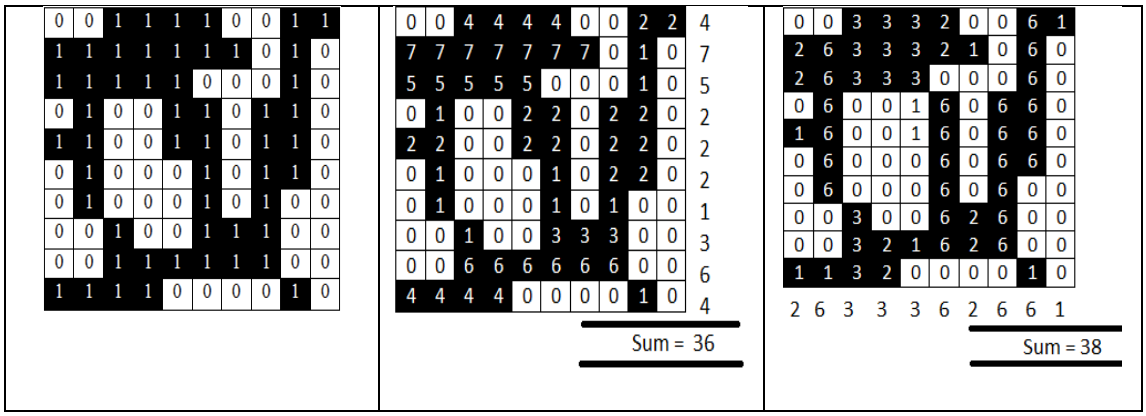


Figure 3.12. (a) Peak Extent based features (b) Horizontal Peak Extent(c) Vertical Peak Extent

- In this method, features are extracted by considering the sum of the peak extents that fit consecutive black pixels along each zone both horizontally and vertically.
- In the horizontal peak extent features, Figure 3.12. (b) Take the sum of the peak extents that fit consecutive black pixels horizontally in each row of a zone.
- In vertical peak extent features, take the sum of the peak extents that fit successive black pixels vertically in each column of a zone as shown in figure 3.12 (c).
- Replace the value by using the sum of all peak extents. The addition of 10 peak extent values will give the feature of that character.
- Normalize the feature value by dividing each element of the feature vector by the largest value in the feature vector in both horizontal and vertical directions.
- After completing the process, we will get 2*n* features for a given character.

3.2.3.5 Curve Fitting Features

In the curve fitting-based feature extraction method, we have divided the thinned image of a character into *n* zones. A parabola is then fitted to the series of ON pixels (foreground pixels) in each zone using the Least Square Method (LSM) as shown in Figure 3.13. This will give 3*n* features for a given character image.

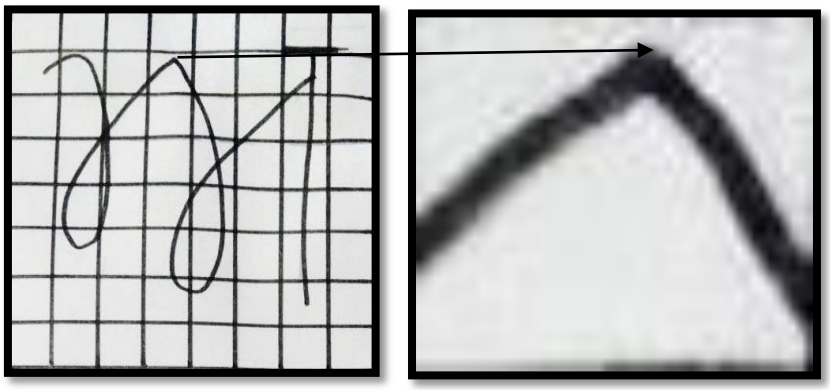


Figure 3.13. Curve Fitting Based Method of Feature Extraction

Step I: Divide the thinned image into *n* number of equal sized zones.

Step II: For each zone, fit a parabola using the least square method and calculate the values of *a*, *b* and *c*.

Step III: Corresponding to the zones that do not have a foreground pixel, set the values of *a*, *b* and *c* as zero.

Step IV: Normalize the feature values in the scale [0, 1] as follows

$$Normalized\ Feature\ Value = \frac{Actual\ Feature\ Value - of\ Actual\ Feature\ Values}{Max\ of\ Actual\ Feature\ Values - Min\ of Actual\ Feature\ Values}$$

3.2.3.6 Intersection and Open-End Point Features

An intersection point is the pixel that has more than one pixel in its neighborhood and an open-end point is a pixel that has only one pixel in its neighborhood.

Step I: Divide the thinned image of a character into *n* zones.

Step II:Calculate the number of intersections and open-end points for each zone. This will give 2*n* features for a character image as shown in Figure 3.14.

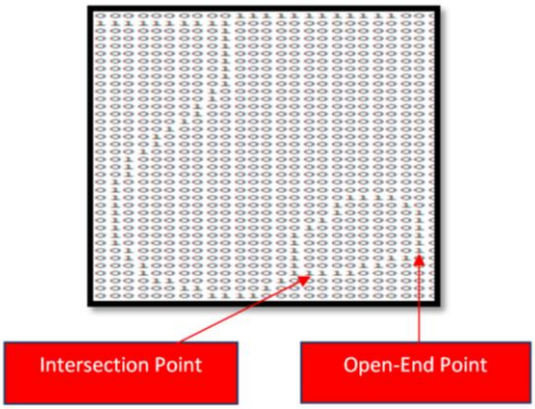


Figure 3.14. Intersection and Open-End Point Method of Feature Extraction

3.2.4 Dimensionality Reduction

This is the process used in statistical pattern recognition for reducing the dimensions of the feature vector i.e., after extracting the feature values, the goal is to select the most prominent features and help in reducing the redundant or irrelevant features. It will reduce the feature values, enhance the generalization, and help in achieving higher accuracy rates with less time, less cost, and less complexity without any loss of meaningful information. For dimensionality reduction, we are implementing Principal

Component Analysis (PCA), which is a method of selecting the **principal components.** As these principal components are ***orthogonal*** to each other, hence these are statistically independent which provides benefits to the dimensionality reduction. Few important characteristics of PCA are:

- It is a tool to reduce multidimensional data to lower dimensions while retaining most of the information.
- Reduction in the number of features within a large dataset without losing its significance (= variance).
- Reduction in computation power, storage, and time needed for modelling.
- Model will be more likely to overfitting on the training examples.
- To achieve the representatives' samples, well-distributed within the whole "universe" of possibilities as the number of features increases.
- PCA is widely used for exploratory data analysis and development of predictive models.

Wang and Chang (2006) proposed a component analysis-based dimensionality reduction method for hyperspectral image analysis. Nie *et al.* (2014) presented a novel robust principal analysis-based dimensionality reduction and its strength and weakness and also removing means optimally. Durou *et al.* (2017) proposed the development of a writer identification system based on dimensionality reduction methods such as Kernel Principal Component Analysis (KPCA), Isomap, Locally Linear Embedding (LLE), Laplacian Eigenmaps on the features extracted using Oriented Basic Image and graphemes and obtained good accuracy results.

3.2.5 Classification Techniques

Classification is the process of recognizing, understanding, and grouping ideas, patterns, and objects into predefined classes according to established criteria e.g., using training datasets, the machine learning program uses a variety of algorithms for classification. Classification algorithms are the predictive calculations used for assigning data into predefined categories or sub-populations. So, mapping of input data to a specific class is called classification. These methods will work of categorizing unsorted data to specific classes as per the training.

The working of classification algorithms is based on many parameters such as Euclidean distance, creating hyperplane, non-linear relationships, and activation functions, etc between input and output. So here we are discussing classification algorithms used for the proposed experimental work.

3.2.5.1 Adaptive Boosting (AdaBoost)

It is an ensemble learning method also named meta-learning which is used to increase the efficiency of binary classifiers. AdaBoost stands for 'Adaptive Boosting' which transforms weak learners or predictors into strong predictors or strong learners in order to classify the data [w15] as shown in Figure 3.15(a) and AdaBoost Classifier is shown in Figure 3.15(b).

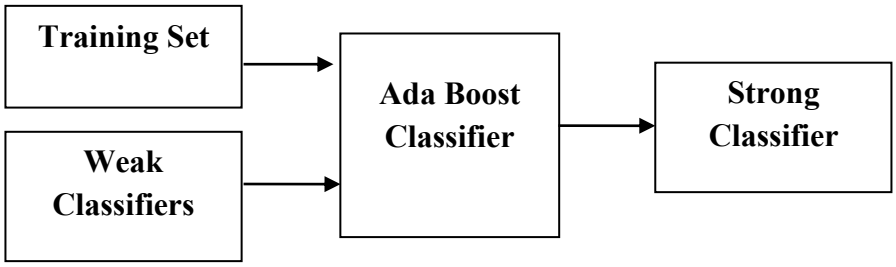


Figure 3.15. (a) Conversion of weak to strong classifier

It uses an iterative approach that learns from the issues of weak classifiers and converts them into the strong classifier. Sequential learners and parallel learners are the kinds of learners that work on the basis of empirical evidence and not comfortable with noisy data and suffer from over-fitting and low margin.

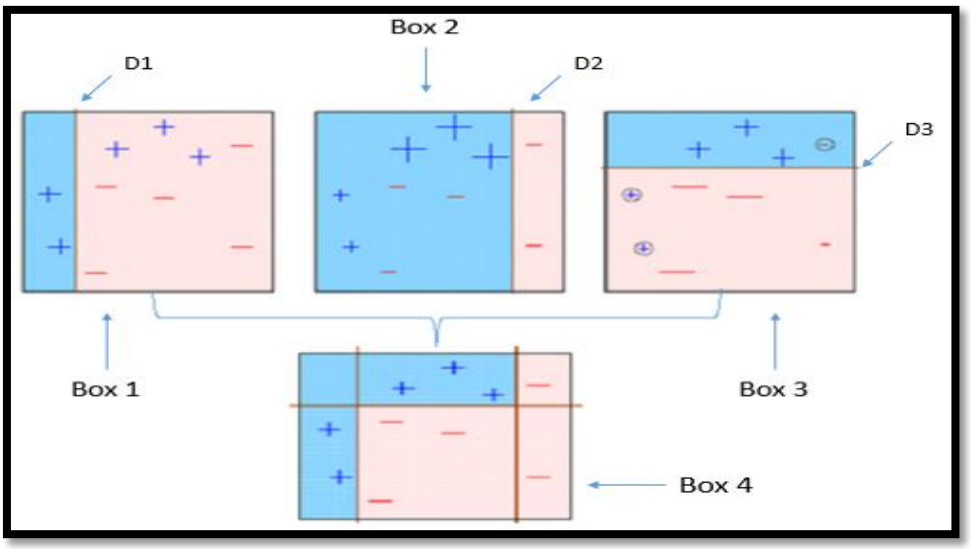


Figure 3.15. (b) AdaBoost Classification Method

3.2.5.2 Artificial Neural Network (ANN)

- ANN is a computational model which is based on the functioning and structure of a biological neural network consisting of 1000 billion neurons called amazing parallel processors in the human brain [w16]. It works on the non-linear statistical data and then processes the non-linear relationship between input and output in parallel as shown in Figure 3.16.
- Artificial neurons are the atomic or elementary units in an artificial neural network.
- Strengths of ANN involve parallel processing capability, capability to work with incomplete knowledge, storing data on the entire network, having a memory distribution, fault tolerance.

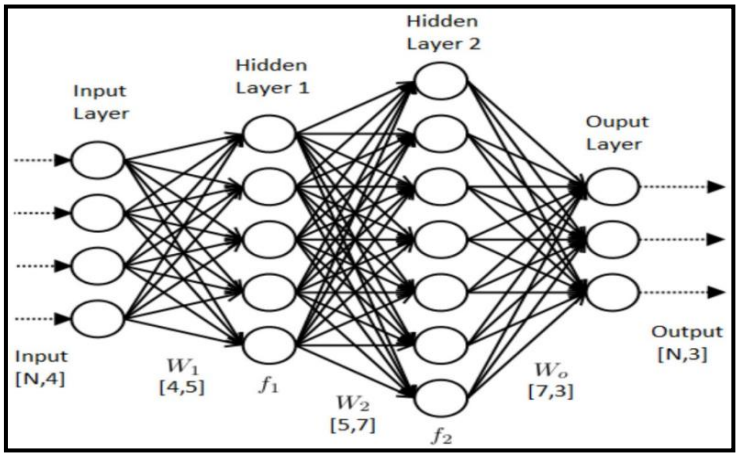


Figure 3.16. Artificial Neural Network

3.2.5.3 Decision Tree (DT)

The key method used in the decision tree is ID3, which is based on features of entropy and information gain. It was given by J. R. Quinlan, which uses a top-down greedy search and having no backtracking as shown in Figure 3.17. For classification and regression, the decision tree has always been the best choice. It is a supervised machine learning method in which data is continuously categorized and matched according to a given parameter [w17].

It has many choices of decision and finally, a leaf node reaches a classification. The topmost decision node in a tree is called the best predictor and the root node. It will be used for both categorical and numerical data. Decision trees are categorized as classification trees and regression trees. The classification tree includes

yes and no or fit and unfit trees and the decision variable is discrete or categorical. A regression tree is one in which the target variable can take continuous values i.e., real numbers.

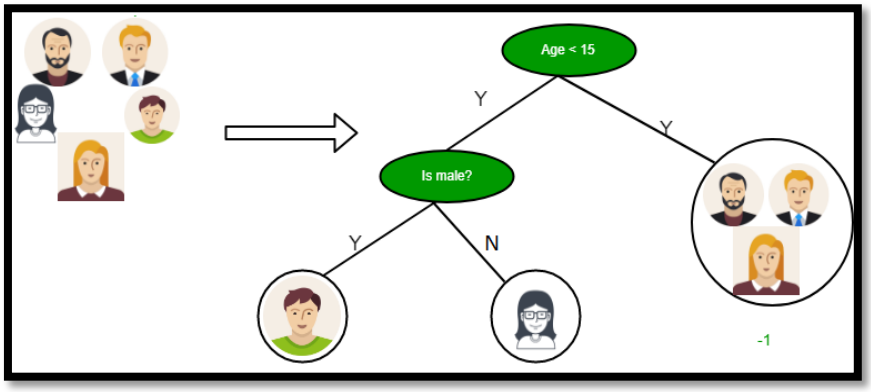


Figure 3.17. Decision Tree

3.2.5.4 K-Nearest Neighbour (K-NN)

- ✓ K-NN is also named as a non-parametric and lazy learning algorithm that records all available cases and classifies the new tested one based on measures such as a distance function as shown in Figure 3.18. It is selected as the best method when there is less or no prior knowledge of the data. K-NN is used in a variety of applications such as speech, video recognition, healthcare, finance, handwriting detection, image recognition and any type of pattern recognition, political science, and statistical estimations [w18].
- ✓ It does not use any training data points to make generalizations which means there is no explicit training phase and has no prior assumptions. Since the algorithm is based upon the distance so the normalization of the training data can improve the accuracy rates dramatically.
- ✓ In simple terms, K represents the number of neighbours and is based on the estimation of the Euclidean distance between the candidate vector and the stored vector. If k = 1, then the case is simply assigned to the class of its nearest neighbour.
- ✓ This method is easy to implement with no training required and new data can be added seamlessly and hence can be used for classification and regression (Kumat *et al*., 2011; Kumar *et al*., 2020).

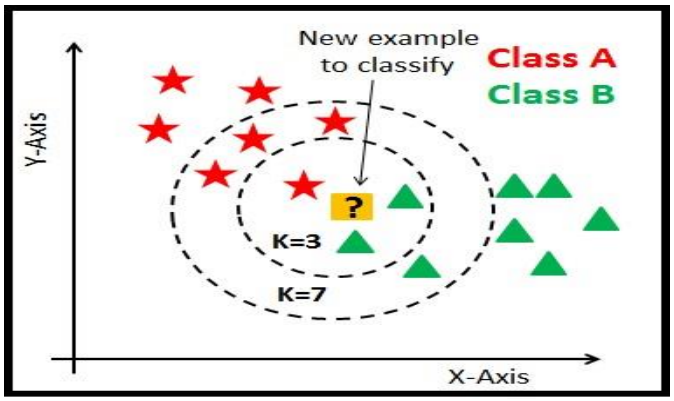


Figure 3.18. K-NN

- ✓ Here, *N* is the total number of features in the feature set, is the library stored feature vector, and is called candidate feature vector.
- ✓ K-NN does not work well with high dimensions, large datasets, poor feature scaling and sensitive to noisy data, incomplete or missing data.

3.2.5.5 Random Forest (RF)

This is an example of the ensemble of algorithms and supervised learning methods used for classification and regression. The working is based on majority voting methodology i.e., it generates a multitude of decision trees at training time and then retrieving an output class that is equal to the mode of classes or mean prediction of individual trees. It uses feature randomness and bagging methodology when creating the individual trees and finally for the uncorrelated forests of trees. It produces the best one by searching over a random $\sqrt{n}$ features, where n is the number of features and finally produces a result based on the average of predictions [w19] as shown in Figure 3.19.

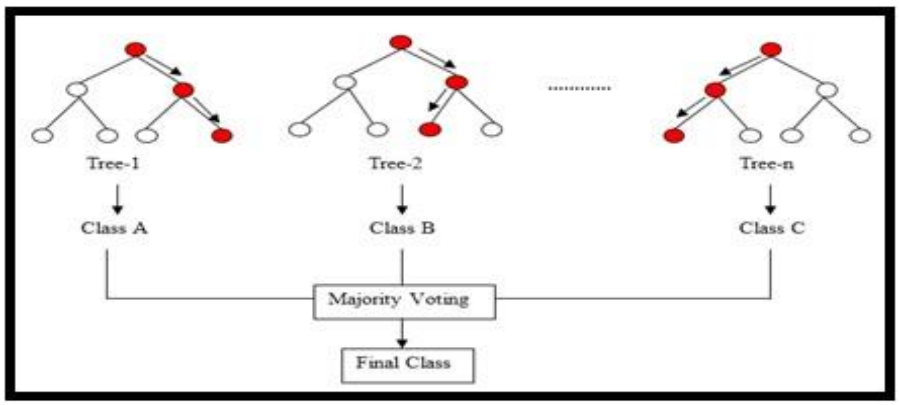


Figure 3.19. Random Forest

3.2.5.6 Multi-Layer Perceptron (MLP)

- ✓ MLP utilizes a supervised learning technique called backpropagation for training.
- ✓ It is based on the non-linear activation function that defines the most complicated architecture of artificial neural networks and can handle linear or non-linear separable data [w20].
- ✓ It does not contain a single perceptron with multiple layers. Rather the network composed of multiple layers of perceptron (with threshold activation) i.e., deals with additional perceptron in layers for handling complex data as shown in Figure 3.20.
- ✓ MLP networks are usually used for the supervised learning format. A typical learning algorithm for MLP networks is also called the backpropagation algorithm.
- ✓ MLPs are the universal function approximators and are widely used for regression and classification in handwriting-based applications, speech recognition, speech recognition, image recognition, etc.

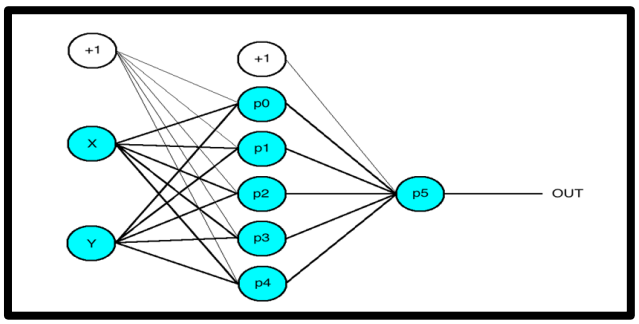


Figure 3.20. Multi-Layer Perceptron

3.2.5.7 Support Vector Machine (SVM)

- ✓ SVMs are based on the statistical learning theory that uses supervised learning. In supervised learning, a machine is trained instead of being programmed to perform a given task on a number of inputs/outputs pairs.
- ✓ The classifier classifies data by selecting appropriate kernel function for the linear kernel, polynomial kernel, and RBF kernel and can be used for linear and nonlinear classification as shown in Figure 3.21.

- ✓ SVM works very well when there is a clear separation of data i.e., the margin is clear, high dimensional data, and semi-structured and unstructured data [w21].
- ✓ It is based on convex optimization and the risk of overfitting is less in SVM.
- ✓ The weakness of SVM is the selection of kernel function is difficult, more time required for training, no probabilistic description, desired classes overlap with noisy data and difficult to interpret the model.
- ✓ It is a very useful technique for text and hypertext categorization, classification of satellite data, handwriting recognition, biological science, breast cancer diagnosis, facial expression classification, protein fold detection, etc.

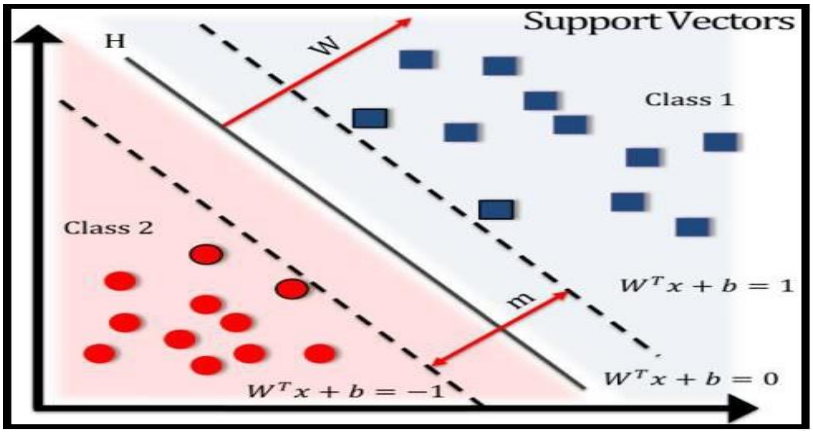


Figure 3.21. Support Vector Machine

3.2.6 Hybridization of Feature Extraction and Classification Techniques

This is the phase that deals with the hybridization of both feature values and classification algorithms. Hybridization of feature values will generate a large set of feature values so that the classifier will generate a more accurate final classification after exploring numerous features of a character. So, the idea is to integrate the capabilities of feature extraction techniques e.g. feature extraction method F1 generates say 85 feature values, and F2 produces 70 feature values so F1+F2 generates 155 feature values. Therefore, the classifier now will produce the best classification result out of 155 values as compared to 85 or 70 individually. Similarly, the hybridization of classification algorithms will strengthen the power of classifiers.

There are so many techniques available for hybridizing the results of classification techniques e.g., are majority voting scheme, bagging, boosting, and stacking. Woznaik (2014) presented various combination rules, topology, and ensemble learning for the hybridization of classification algorithms. Mousavi and Eftekhari (2015) presented an ensemble learning methodology for the hybridization of classification methods. Demidova *et al.* (2019) presented a hybrid approach for improving the object classification results. Hybridization is done between the SVM and random forest and k-NN classifiers and it proves the expediency of hybridization.

3.2.7 Evaluating Performance Metrics

The last step is to evaluate the performance metrics such as accuracy, precision, true positive rate, false-positive rate, CPU elapsed time, root mean square error, the area under the curve. Jiao and Du (2016) discussed performance measures such as positive predictive value, false discovery rate, Jaccard index, True positive, false positive, etc. for evaluating the success of the experiments (Singh *et al*., 2015; Botchkarev, 2019). Various performance metrics are:

- ✓ **True Positive Rate (TPR):** TPR defines how many correct positive results occur between all positive samples available during the test.
- ✓ **False Positive Rate (FPR):** FPR defines how many incorrect positive results occur between all negative samples available during the test.
- ✓ **True Positive (TP):** It is an outcome where the model correctly predicts the **positive** class.
- ✓ **True Negative (TN)**: It is an outcome where the model correctly predicts the negative class.
- ✓ **False Positive (FP)**: It is an outcome where the model gets positive results when it should have received negative results.
- ✓ **False Negative (FN)**: It is an outcome where the model got negative results when it should have received positive results.
- ✓ **Accuracy**: It is defined as the degree to which the result of experiments, calculations and measurements conforms to the standard value. Figure 3.22 shows the difference between accuracy and precision.

$$Accuracy = \frac{True\ Negative + True\ Positive}{True\ Positive + False\ Posiive + True\ Negative + False\ Negative}$$

- ✓ **Precision Rate**: It is defined as the measure of quality. It means our method retrieves more relevant results than irrelevant ones. It gives us the closeness of the measurements to each other on the other hand accuracy is the closeness of the measurements to a specific value [w22].

$$Precision = \frac{True\ Positive}{True\ Positive + False\ Positive}$$

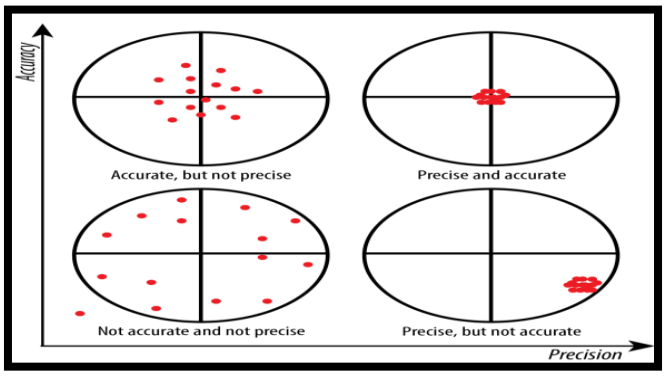


Figure 3.22. Accuracy and Precision

- ✓ **Area under Curve (AUC):** It helps in selecting a compound or an area that provides the highest exposure levels as shown in Figure 3.23 [w23].

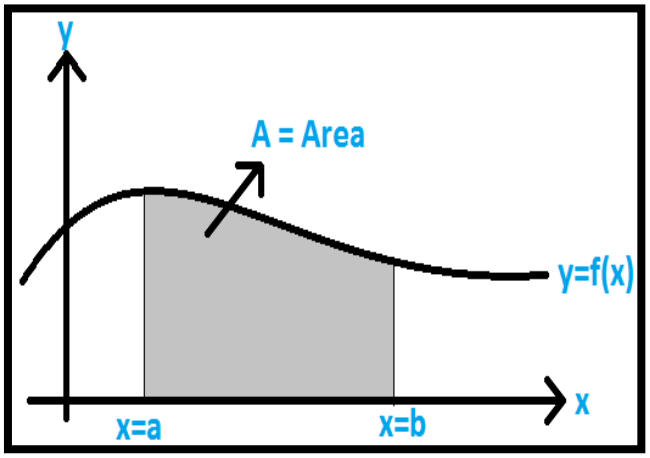


Figure 3.23 Area under curve

- ✓ **CPU Elapsed Time:** CPU time refers to the actual time spent on the CPU and elapsed time refers to the total time taken for the completion of the parse and compile.

- ✓ **Root Mean Square Error (RMSE):** The root-mean-square deviation (RMSD) or root-mean-square error (RMSE) is a frequently used measure of the differences between values (sample or population values) predicted by a model or an estimator and the values observed. RMSE is the square root of the average of squared errors.

3.3 CHAPTER SUMMARY

In this chapter, we presented the keystone of research without which the research is incomplete i.e., data collection and generation of the corpus. The process of data collection, selection of subject for data collection, quality of data, and maintenance of data are elaborated. The next section describes the framework that represents the workflow with the detailed working of the phases. All phases such as Pre-processing, feature extraction, dimensionality reduction, classification and hybridization phase have been deeply discussed with necessary images. Performance metrics are also evaluated at the end of this chapter to explore the quality of experiment results revealed.