COMPARATIVE EVALUATION OF CLASSIFIER MODELS FOR AUTOMATIC TARGET RECOGNITION IN SYNTHETIC APERTURE RADAR IMAGE

A Project report submitted in partial fulfillment of the requirements for the award of the degree of

BACHELOR OF TECHNOLOGY

IN

ELECTRONICS AND COMMUNICATION ENGINEERING

A.Pratyusha (317126512001)   K.Abhiram (317126512023)

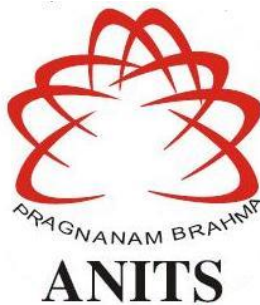R.Balaji Kanaka Sai Kumar (317126512049)   U.Bhargav (317126512054)

Under the guidance of

Mrs. D. Nagamani

M.Tech (Ph.D.)

Assistant Professor

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

ANIL NEERUKONDA INSTITUTE OF TECHNOLOGY AND SCIENCES
(UGC AUTONOMOUS)
(*Permanently Affiliated to AU, Approved by AICTE and Accredited by NBA & NAAC with 'A' Grade*)
Sangivalasa, bheemili mandal, Visakhapatnam dist.(A.P)
2020-2021

ANIL NEERUKONDA INSTITUTE OF TECHNOLOGY AND SCIENCES

(Permanently Affiliated to AU, Approved by AICTE and Accredited by NBA & NAAC with 'A' Grade)

Sangivalasa, Bheemili Mandal, Visakhapatnam dist.(A.P)-531162

PRAGNANAM BRAHMA

ANITS

CERTIFICATE

This is to certify that the project report entitled "**Comparative Evaluation of Classifier models for Automatic Target Recognition in Synthetic Aperture Radar Image**" submitted by **A.Pratyusha (317126512001), K.Abhiram (317126512023), R.Balaji Kanaka Sai Kumar (317126512049), U.Bhargav (317126512054)** in partial fulfillment of the requirements for the award of the degree of **Bachelor of Technology** in **Electronics & Communication Engineering** of Andhra University, Visakhapatnam is a record of bonafide work carried out under my guidance and supervision.

ProjectGuide

Ms. D. Nagamani
M.Tech (Ph.D.)
Assistant Professor
Department of E.C.E
ANITS

Assistant Professor
Department of E.C.E.
Anil Neerukonda
Institute of Technology & Scien
Sangivalasa, Visakhapatnam-53

Head of theDepartment

Dr. V.Rajyalakshmi
M.E,Ph.D,MIEEE,MIE,MIETE,MHRM
Department of E.C.E
ANITS

Head of the Department
Department of E C E
Anil Neerukonda Institute of Technology & Sciences
Sangivalasa - 531 162

ACKNOWLEDGEMENT

PROJECT STUDENTS
A.Pratyusha (317126512001

K.Abhiram (317126512023)

R.Balaji Kanaka Sai Kumar (317126512049)

 U.Bhargav (317126512054)

ABSTRACT

SAR works in the Microwave band to produce highly accurate output images. These SAR images are used in military applications, observation of the earth and its climatic conditions, etc. So the object detection in the image should be very accurate. This problem can be sorted out by using an effective Machine learning technique. Several algorithms have been developed like KNN, SVM, and CNN. In this project, SVM based CNN technique is proposed and compared with other classification techniques (SVM, CNN, K-NN) on MSTAR (Man- portable Surveillance and Target Acquisition Radar) dataset. Predictive analysis of these classification techniques has been evaluated in terms of accuracy, precision, recall, F1-score. The proposed SVM based CNN technique gives better accuracy compared to other Classification techniques.

CONTENTS

CHAPTER-7 RESULTS AND CONCLUSION

List of Figures

CHAPTER-1

INTRODUCTION

Image classification is the process of categorizing and labeling groups of pixels or vectors within an image based on specific rules. The categorization law can be devised using one or more spectral or textural characteristics. Image classification plays an important role in remote sensing images and is used for various applications such as environmental change, agriculture, land use/land planning, urban planning, surveillance, geographic mapping, disaster control, and object detection and also it has become a hot research topic in the remote sensing community. The remote sensing image data can be obtained from various resources like satellites, airplanes, and aerial vehicles. In this work classification method for automatic target recognition in synthetic aperture radar (SAR) images is proposed. Due to its advantages of all-day, all-weather, and strong penetrating capability, synthetic aperture radar (SAR) has been widely used in military and civil fields.

SAR is a kind of active microwave imaging radar, which can obtain two-dimensional (2-D) images with high resolution. The automatic target recognition (ATR) is for SAR images to extract stable and iconic features based on SAR images, and determine its category attribute and confirm its particular copies of the same class, which can be applied to battlefield monitoring, guidance attack, attack effect assessment, marine resource detection, environmental geomorphology detection, and natural disaster assessment, and has vital research significance.

1.1 MOTIVATION OF THE PROJECT

Human visual performance greatly exceeds computer capabilities, probably because of superior high-level image understanding, contextual knowledge, and massively parallel processing. Human capabilities deteriorate drastically in a low-visibility environment or after an extended period of surveillance, and certain

working environments are either inaccessible or too hazardous for human beings. For these reasons, automatic recognition systems are developed for various military and civilian applications.

The development of robust automatic target recognition (ATR) systems must still overcome a number of well-known challenges: for example, the large number of target classes and aspects, long viewing range, obscured targets, high-clutter background, different geographic and weather conditions, sensor noise, and variations caused by translation, rotation, and scaling of the targets. Inconsistencies in the signature of targets, similarities between the signatures of different targets, limited training and testing data, camouflaged targets, non-repeatability of target signatures, and difficulty using available contextual information makes the recognition problem even more challenging.

1.2 LITERATURE SURVEY

There has been a considerable amount of research over the last few years in improvising the image classification techniques for Synthetic Aperture Radar (SAR)Automatic Target Recognition (ATR).

Dan E. Dudgeon and Richard T. Lacoss wrote about ATR Technology, Detection Theory and Pattern Recognition in which they mentioned that the efforts are continuing to accelerate the development of recognition algorithms. Because of the amount of data required to develop and test ATR systems for robustness, and the cost of mounting realistic data collection efforts, the development of accurate synthetic data-generation tools is also being addressed. Just as the speech-recognition problem required decades of serious research before simple applications could be brought to market, the ATR problem needs a wide range of research before the practical systems. Most people believe the ATR problem will not be solved by a single brilliant idea. The solution will probably require a combination of improved sensors, faster computers, and better algorithms.

Chih-Chung Chang and Chih-Jen Lin have discussed about the library being developed for the Support Vector Machines(SVM's) .They have been actively developing this package since the year 2000. Their goal is to help users to easily apply SVM to their applications. Issues such as solving SVM optimization problems, theoretical convergence, multi-class classification, probability estimates, and parameter selection are discussed by them in detail.

Lijuan Zhou, Linshuang Wang, Xuebin Ge and Qian Shi gave an overview about the K-NN algorithm for uneven data. KNN classifier may have a problem when training samples are uneven. The problem is that KNN classifier may decrease the precision of classification because of the uneven density of training data. To solve the problem, they gave a new clustering-based KNN method. It preprocesses training data by using clustering, then classify with a new KNN algorithm, which adopts a dynamic adjustment in each iteration for the neighborhood number K. This method would avoid the uneven classification phenomenon and reduce the misjudgment of the boundary testing samples.

Shengna WU, Ke WANG and Yiwen OUYANG have proposed transfer-based Convolutional Neural Network (CNN) small samples SAR images ground object recognition detection method. First of all, the source domain sufficient samples were used to pre-train CNN and then CNN was fine-tuned by using the transfer of small samples of the target domain SAR images to obtain a new CNN. The network finally realizes ground object recognition detection of small samples SAR images.

Ievgen M. Gorovyi and Dmytro S. Sharapov have discussed about the, the classification step is ATR. They implemented SVM and CNN techniques for MSTAR Database which comprises of 10 classes with testing and training samples and demonstrated that two different machine learning instruments can provide very high classification accuracy.

1.3 PROBLEM ANALYSIS AND PROPOSED APPROACH

A strategy is introduced for achieving high accuracy in synthetic aperture radar (SAR) automatic target recognition (ATR) tasks. SAR ATR involves a sequence of processes, such as some type of preprocessing, feature extraction, classifier construction, and finally target classification. The preprocessing stage may involve multiple types of processing that aims at facilitating the efficiency of image interpretation and analysis in the subsequent stages, for example, by suppressing the clutter reflections that obscure the contrast between the target of interest and the clutter. Moreover, SAR images are resized, shifted, and rotated to predefined standards. The so-called resizing is normally implemented by cropping out part of the image. The shifting and rotating processes are also known as image registration and pose rectification, respectively.

Feature extraction is another essential stage which extracts effective discriminant features for improving recognition accuracy. Several features have already been exploited in SAR ATR. Methods such as Support Vector Machines (SVM), Convolutional Neural Networks (CNN), and K-Nearest Neighbors (KNN) are all vastly exploited in classification step of SAR ATR.

1.4 ORGANISATION OF PROJECT REPORT

The project report is organized in seven chapters. It starts from the introduction. The current chapter introduces about the project and gives a brief description on image classification and SAR ATR. Chapter 2 provides information about image classification using supervised learning, its applications and advantages. Chapter 3 gives detailed explanation about SVM and K-NN algorithms, its advantages and applications. Chapter 4 explains about CNN and its architecture and algorithm. Chapter 5 explains about the software description. Chapter 6 presents the project description and the performance measures of the models. Chapter 7 contains the results and gives the conclusion and references for the project.

CHAPTER-2

IMAGE CLASSIFICATION

2.1 INTRODUCTION

Digital Image processing is the numerical computation of digital images and includes preprocessing, enhancement and classification. Preprocessing refers to the initial processing of the raw data to calculate the image radiometry, correct geometric distortion, and remove noise. Image enhancement produces a new enhanced image that is displayed on Cathode Ray Tube (CRT). This enhancement image may be easier to interpret than the original image in different ways. Image classification carriers the digital processing a step further and attempts to replace the visual interpretation step with Quantitative decision making. Digital image classification is the use of machine learning algorithms to perform image classification on digital images.

The 2D continuous image is divided into N rows and M columns. The intersection of a row and a column is called a pixel. The image can also be a function other variables including depth, color, and time. An image given in the form of a transparency, slide, photograph or an X-ray is first digitized and stored as a matrix of binary digits in computer memory. This digitized image can then be processed and/or displayed on a high-resolution television monitor. For display, the image is stored in a rapid-access buffer memory, which refreshes the monitor at a rate of 25 frames per second to produce a visually continuous display.

2.1.1 DEFINITION OF AN IMAGE

The term image refers to a two-dimensional light intensity $f(x, y)$. Where x and y denote spatial coordinates and the value of f at any point (x, y) is proportional to the brightness (Or grey level) of the image at the point.

Figure 2.1 IMAGE

2.1.2 DIGITAL IMAGE

The term digital images refer to an image f (x, y) that has been discretized both in spatial coordinates and brightness. A digital image can be considered a matrix whose row and column indices identify a point in the image and the corresponding matrix element value identifies the grey level at that point. The elements of such a digital array are called picture elements, or more commonly pixels. The images width is represented by the variable N the images height with the variable M as shown below.

$$f(x,y) = \begin{bmatrix} f(0,0) & f(0,1) & f(0,2) & \cdots & f(0,N-1) \\ f(1,0) & f(1,1) & f(1,2) & \cdots & f(1,N-1) \\ \cdot & \cdot & \cdot & & \cdot \\ \cdot & \cdot & \cdot & & \cdot \\ \cdot & \cdot & \cdot & & \cdot \\ f(M-1,0) & f(M-1,1) & f(M-1,2) & \cdots & f(M-1,N-1) \end{bmatrix}$$

2.1.3 TYPES OF IMAGES

There are several ways of encoding the information in an image.

1. Binary image

2. Gray scale image

 3. Indexed image

4. True color or RGB image

BINARY IMAGE

Each pixel is just black or white. Since there are only two possible values for each pixel (0, 1), only one bit per pixel is needed.

GRAYSCALE IMAGE

Each pixel is a shade of gray, normally from 0 (black) to 255(white). This range means that each pixel can be represented by eight bits, or exactly one byte. Other gray scale ranges are used, but generally they are a power of 2.

INDEXED IMAGE

An indexed image consists of an array and a color map matrix. The pixel values in the array are direct indices into a color map. By convention, this documentation uses the variable name X to refer to the array and map to refer to the color map.

TRUE COLOR OR RGB IMAGE

Each pixel has a particular color; that color is described by the amount of red, green and blue in it. If each of these components has a range 0–255, this gives a total of 256^3 different possible colors. Such an image is a "stack" of three matrices; representing the red, green and blue values for each pixel. This means that for every pixel there correspond 3 values.

2.2 CONCEPT OF IMAGE CLASSIFICATION

Classification is the process of assigning spectral classes into information classes. Spectral classes are groups of pixels that are uniform with respect to their brightness values in the different spectral channels of data. Information classes are categories of interest that an analyst attempts to identify in the image on the basis of his knowledge and experience about the area. For example, a remote sensing image contains spectral signatures of several features present on the ground in terms of pixels of different values.

An interpreter or analyst identifies homogeneous groups of pixels having similar values and labels the groups as information classes such as water, agriculture, forest, etc. while generating a thematic map. When this thematic information is extracted with the help of software, it is known as digital image classification.

It is important to note that there could be many spectral classes within the information class depending upon the nature of features the image represents or the purpose of the classification. In other words, different spectral classes may be grouped under one information class. In short, image classification is defined as a process of assigning all pixels in the image to particular classes or themes based on spectral information represented by the digital numbers (DNs). The classified image comprises a mosaic of pixels, each of which belong to a particular theme and is a thematic map of the original image.

2.2.1 Approaches to Classification

There are two general approaches to image classification:

1. Supervised Classification:

It is the process of identification of classes within a remote sensing data with inputs from and as directed by the user in the form of training data.

2. Unsupervised Classification:

It is the process of automatic identification of natural groups or structures within a remote sensing data.

Both the classification approaches differ in the way the classification is performed. In the case of supervised classification, specific land cover types are delineated based on statistical characterization of data drawn from known examples in the image (known as training sites). In unsupervised classification, however, clustering algorithms are used to uncover the commonly occurring land cover types, with the analyst providing interpretations of those cover types at a later stage.

In this study **supervised machine learning** is used technique for the classification step in SAR-ATR.

2.2.2 SUPERVISED CLASSIFICATION

Supervised learning is the machine learning task of inferring a function from labeled training data. The training data consist of a set of training examples. In supervised learning, each example is a pair consisting of an input object (typically a vector) and a desired output value (also called the supervisory signal). A supervised learning algorithm analyzes the training data and produces an inferred function, which can be used for mapping new examples. An optimal scenario will allow for the algorithm to correctly determine the class labels for unseen instances. This requires the learning algorithm to generalize from the training data to unseen situations in a "reasonable" way.

In order to solve a given problem of supervised learning, one has to perform the following steps:

1. Determine the type of training examples. Before doing anything else, the user should decide what kind of data is to be used as a training set. In the case of handwriting analysis, for example, this might be a single handwritten character, an entire handwritten word, or an entire line of handwriting.

2. Gather a training set. The training set needs to be representative of the real-world use of the function. Thus, a set of input objects is gathered and corresponding outputs are also gathered, either from human experts or from measurements.

3. Determine the input feature representation of the learned function. The accuracy of the learned function depends strongly on how the input object is represented. Typically, the input object is transformed into a feature vector, which contains a number of features that are descriptive of the object. The number of features should not be too large, because of the curse of dimensionality; but should contain enough information to accurately predict the output.

4. Determine the structure of the learned function and corresponding learning algorithm. For example, the engineer may choose to use support vector machines or K-nearest neighbors.

5. Complete the design. Run the learning algorithm on the gathered training set. Some supervised learning algorithms require the user to determine certain control

parameters. These parameters may be adjusted by optimizing performance on a subset (called a validation set) of the training set, or via cross-validation.

6. Evaluate the accuracy of the learned function. After parameter adjustment and learning, the performance of the resulting function should be measured on a test set that is separate from the training set. A wide range of supervised learning algorithms is available, each with its strengths and weaknesses. There is no single learning algorithm that works best on all supervised learning problems.

2.3 ADVANTAGES AND DISADVANTAGES OF SUPERVISED CLASSIFICATION ALGORITHM

ADVANTAGES

1. Supervised learning allows collecting data or producing a data output from the previous experience.
2. Helps to optimize performance criteria using experience.
3. Supervised machine learning helps to solve various types of real-world computation problems.
4. Analyst has control over the classification.
5. Processing is tied to specific areas of known identity.
6. Errors can be detected and often rectified

DISADVANTAGES

1. Decision boundary might be over trained if the training set which doesn't have examples that wanted to have in a class.
2. Need to select lots of good examples from each class while training the classifier.
3. Classifying big data can be a real challenge.
4. Training for supervised learning needs a lot of computation time.

2.4 TRAINING, TESTING AND VALIDATING

In a perfect world everyone could perform a test on data that your machine learning algorithm has never learned from before. However, waiting for fresh data isn't always feasible in terms of time and costs. As a first simple remedy, it can randomly split our data into training and test sets. The common split is from 25 to 30 percent for testing and the remaining 75 to 70 percent for training. Split the data consisting of our response and features at the same time, keeping correspondence between each response and its features.

The second remedy occurs when there is need to tune the learning algorithm. In this case, the test split data isn't a good practice because it causes another kind of over fitting called snooping. To overcome snooping, just need a third split is needed, called a validation set. A suggested split is to have your examples partitioned in thirds: 70 percent for training, 20 percent for validation, and 10 percent for testing. The split should perform randomly, that is, regardless of the initial ordering of the data. Otherwise, the test won't be reliable, because ordering could cause overestimation or underestimation. As a solution, one must ensure that the test set distribution isn't very different from the training distribution, and that sequential ordering occurs in the split data. For example, check whether identification numbers, when available, are continuous in the taken sets. Sometimes, even if it is strictly abide by random sampling, it always cannot obtain similar distributions among sets, especially when your number of examples is small.

CHAPTER -3

SUPPORT VECTOR MACHINES (SVM) AND K-NEAREST NEIGHBORS (K-NN)

3.1 INTRODUCTION

SVM AND K-NN are the supervised classification algorithms used for the classification for this study. In supervised learning, algorithms learn from labeled data. After understanding the data, the algorithm determines which label should be given to new data by associating patterns to the unlabeled new data K-NN algorithm is one of the simplest classification algorithms and it is used to identify the data points that are separated into several classes to predict the classification of a new sample point. K-NN is a non-parametric, lazy learning algorithm. It classifies new cases based on a similarity measure (i.e., distance functions).Support vector is used for both regression and classification. It is based on the concept of decision planes that define decision boundaries. A decision plane (hyper plane) is one that separates between a set of objects having different class memberships.

3.2 FEATURE EXTRACTION - PRINCIPAL COMPONENT ANALYSIS (PCA)

Feature extraction is a common step of Support Vector Machines (SVM) K-Nearest Neighbors machine (NN) learning technique. PCA is used for feature extraction in this project. PCA reduces the dimensions of the datasets and converts large datasets into small datasets. This will make it easy for algorithms to analyse data. The variables of the datasets are reduced in such a way that the accuracy of the datasets is not affected. Standardization of dataset variables has to be done so that each variable can be analysed accurately. The various datasets can be normalized with the formula Z=value-mean/Standard Deviation.

Correlation gives the relation between the data variables. A square n*n matrix, where n is number of Dimensions. This matrix is symmetric. Here the values of variables are

range from -1 to +1. If the variables are not related to each other then they are called orthogonal and their values are zero. An example of a 3*3 covariance matrix is:

$$\begin{bmatrix} Cov(x,x) & Cov(x,y) & Cov(x,z) \\ Cov(y,x) & Cov(y,y) & Cov(y,z) \\ Cov(z,x) & Cov(z,y) & Cov(z,z) \end{bmatrix}$$

Eigenvalues and Eigenvectors are determined for the above matrix and principal components of data are determined. The largest possible variance of all principal components is calculated until it reaches the n principal components. A feature vector is a column matrix, formed by the resulting Eigenvectors of each dataset variable. Hence the dimension of the datasets reduces. Finally, a principal component analysis is performed to plot back to the original image. It can be obtained by multiplying the transpose of the feature vector and the transpose of the original dataset.

3.3 SUPPORT VECTOR MACHINES (SVM)

3.3.1 DEFINITION

A support vector machine constructs a hyper plane or set of hyper planes in a high- or infinite dimensional space, which can be used for classification, regression, or other tasks. Intuitively, a good separation is achieved by the hyper plane that has the largest distance to the nearest training-data point of any class (so-called functional margin), since in general the larger the margin the lower the generalization error of the classifier. Whereas the original problem may be stated in a finite dimensional space, it often happens that the sets to discriminate are not linearly separable in that space. For this reason, it was proposed that the original finite-dimensional space be mapped into a much higher-dimensional space, presumably making the separation easier in that space. To keep the computational load reasonable, the mappings used by SVM schemes are designed to ensure that dot products may be computed easily in terms of the variables in the original space, by defining them in terms of a kernel function k(x, y) selected to suit the problem.

3.3.2 HYPER PLANE, SUPPORT VECTORS AND KERNEL

HYPER PLANE:

There can be multiple lines/decision boundaries to segregate the classes in n-dimensional space, but the best decision boundary is needed that helps to classify the data points. This best boundary is known as the hyper plane of SVM. The dimensions of the hyper plane depend on the features present in the dataset, which means if there are 2 features then hyper plane will be a straight line. And if there are 3 features, then hyper plane will be a 2-dimension plane. Always a hyper plane is created that has a maximum margin, which means the maximum distance between the data points.

SUPPORT VECTORS:

The data points or vectors that are the closest to the hyper plane and which affect the position of the hyper plane are termed as Support Vector. Since these vectors support the hyper plane, hence called a Support vector.

KERNEL

Kernel in the SVM is responsible for transforming the input data into the required format. Different SVM algorithms use different types of kernel functions. These functions can be different types. For example linear, non-linear, polynomial, radial basis function (RBF) and Sigmoid. The most used type of kernel function is **RBF.** Because it has localized and finite response along the entire x axis. For complex applications, one should use more advanced kernels to separate classes that are nonlinear in nature. With this transformation, one can obtain accurate classifiers.

The kernel functions return the inner product between two points in a suitable feature space. Thus by defining a notion of similarity, with little computational cost even in very high dimensional spaces.

3.3.3 TYPES OF SVM

SVM can be of two types:

1. **Linear SVM:** Linear SVM is used for linearly separable data, which means if a dataset can be classified into two classes by using a single straight line, then such data is termed as linearly separable data, and classifier is used called as Linear SVM classifier.
2. **Non-linear SVM:** Non-Linear SVM is used for non-linearly separated data, which means if a dataset cannot be classified by using a straight line, then such data is termed as non-linear data and classifier used is called as Non-linear SVM classifier.

3.3.4 WORKING OF SVM

Linear SVM:

The working of the SVM algorithm can be understood by using an example. Suppose the dataset that has two tags (green and blue), and the dataset has two features x1 and x2. A classifier is needed that can classify the pair(x1, x2) of coordinates in either green or blue. Consider the below image.
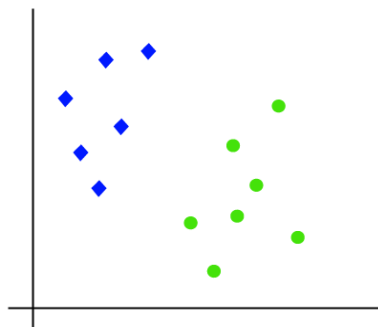

FIGURE 3.1 DATASETS WITH TWO TAGS (GREEN AND BLUE)

So as it is 2-d space so by just using a straight line it can easily separate these two classes. But there can be multiple lines that can separate these classes.
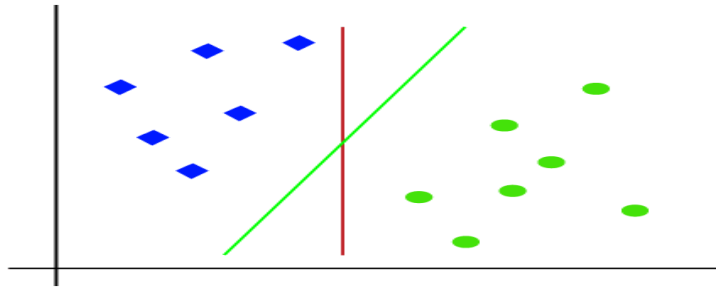
Consider the below image

FIGURE 3.2 SEPERATION OF CLASSES USING STRAIGHT LINES

Hence, the SVM algorithm helps to find the best line or decision boundary; this best boundary or region is called as a **hyper plane**. SVM algorithm finds the closest point of the lines from both the classes. These points are called support vectors. The distance between the vectors and the hyper plane is called as **margin**. And the goal of SVM is to maximize this margin. The hyper plane with maximum margin is called the **optimal hyper plane**.
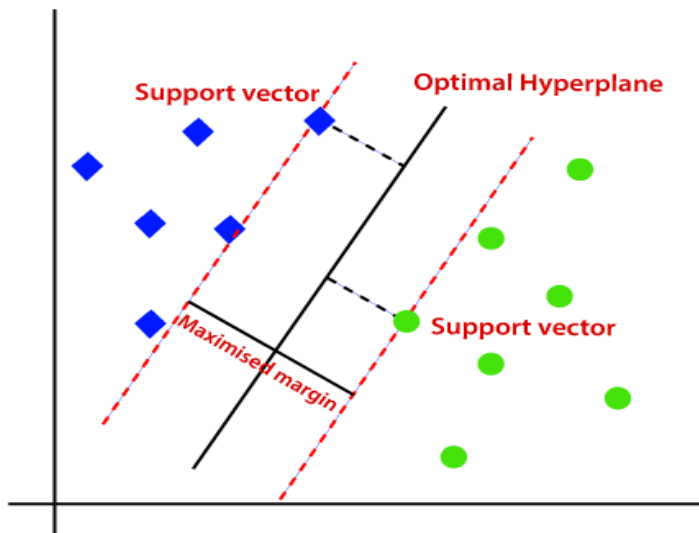

FIGURE 3.3 SVM MODEL WITH OPTIMAL HYPERPLANE

Non-Linear SVM:

If data is linearly arranged, it can be separated by using a straight line, but for non-linear data, a single straight line cannot be drawn. Consider the below image:
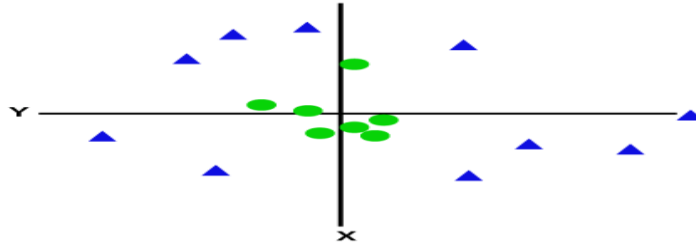

FIGURE 3.4 NON-LINEAR DATA ARRANGMENT

So to separate these data points, one more dimension should be added. For linear data, two dimensions x and y are used, so for non-linear data, a third dimension z is added. It can be calculated as $z=x^2 +y^2$.

By adding the third dimension, the sample space will become as below image:
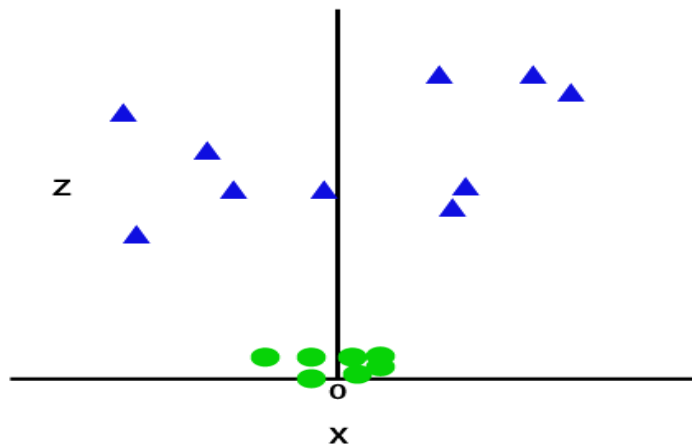
FIGURE 3.5 SEPERATING CLASSES USING 3RD DIMENSION

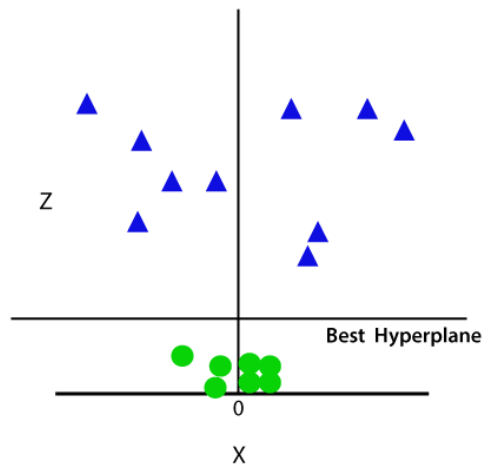So now, SVM will divide the datasets into classes in the following way. Consider the below image:

FIGURE 3.6 BEST HYPERPLANE FOR NON-LINEAR SVM

Since it is in 3-d Space, hence it is looking like a plane parallel to the x-axis. If it ids converted into 2d space with z=1, then it will become as:
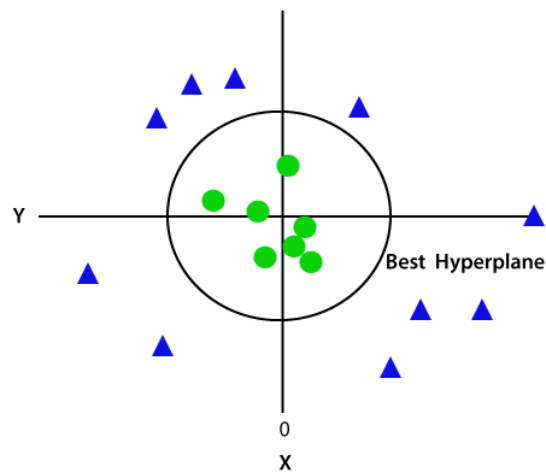

FIGURE 3.7 HYPERPLANE IN 2-D PLANE

Hence, a circumference of radius 1 in case of non-linear data is obtained.

3.3.5 ADVANTAGES AND DISADVANTGES OF SVM

ADVANTAGES

1. Support vector machine is very effective even with high dimensional data.
2. When the data set is in the form where number of features is more than the number of rows of data, SVM can perform in that case as well.
3. When classes in the data are points are well separated SVM works really well.
4. SVM can be used for both regression and classification problem.
5. And last but not the least SVM can work well with image data as well.

DISADVANTAGES

1. When classes in the data are points are not well separated, which means overlapping classes are there, SVM does not perform well.
2. Choosing an optimal kernel for SVM manually is a difficult task.
3. SVM on large data set comparatively takes more time to train.
4. SVM or Support vector machine is not a probabilistic model so it cannot be explained this classification in terms of probability.
5. It is difficult to understand and interpret the SVM model compared to Decision tree as SVM is more complex.

3.3.6 APPLICATIONS OF SVM

- SVMs are helpful in text and hypertext categorization as their application can significantly reduce the need for labeled training instances in both the standard inductive and transductive settings.
- Classification of images can also be performed using SVMs. Experimental results show that SVMs achieve significantly higher search accuracy than traditional query refinement schemes after just three to four rounds of relevance feedback.
- SVMs are also useful in medical science to classify proteins with up to 90% of the compounds classified correctly.
- Hand-written characters can be recognized using SVM.

3.4 K-NEAREST NEIGHBORS (K- NN)
3.4.1 DEFINITION

K nearest neighbors (KNN) is a supervised machine learning algorithm. A supervised machine learning algorithm's goal is to learn a function such that $f(X) = Y$ where X is the input, and Y is the output. KNN can be used both for classification as well as regression. The properties of KNN are that it is a lazy learning algorithm and a non-parametric method.

Lazy learning means the algorithm takes almost zero time to learn because it only stores the data of the training part (no learning of a function). The stored data will then be used for the evaluation of a new query point. The non-parametric method refers to a method that does not assume any distribution. Therefore, KNN does not have to find any parameter for the distribution. While in the parametric method, the model finds new parameters, which in turn will be used for the prediction purpose. The only hyper parameter (provided by the user to the model) KNN has is K, which is the number of points that needs to be considered for comparison purpose.
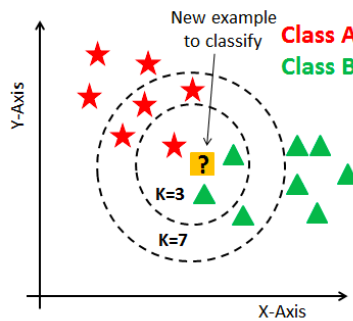
3.4.2 ALGORITHM

FIGURE 3.8 K-NN Example

In the above image, yellow is the query point, to know which class it belongs to (red or green), with K=3, the 3 nearest neighbors of the yellow point are considered, and the class is assigned to the query point based on the majority (e.g., 2 green and 1 red — then it is of green class).

Similarly, for K=5, 5 nearest neighbors are considered for the comparison, and the majority will decide which class the query point belongs to. One thing to notice here, if the value of K is even, it might create problems when taking a majority vote because the data has an even number of classes (i.e., 2). Therefore, choose K as an odd number when the data has an even number of classes and even number when the data has an odd number of classes.

3.4.3 WORKING OF K-NN

In the training phase, the model will store the data points. In the testing phase, the distance from the query point to the points from the training phase is calculated to classify each point in the test dataset. Various distances can be calculated, but the most popular one is the Euclidean distance (for smaller dimension data).

The K-NN working can be explained on the basis of the below algorithm:

1. Select the number K of the neighbors
2. Calculate the Euclidean distance of K number of neighbors
3. Take the K nearest neighbors as per the calculated Euclidean distance.
4. Among these k neighbors, count the number of the data points in each category.
5. Assign the new data points to that category for which the number of the neighbor is maximum and the model is ready.

Suppose a new data point is added and to put the point in the required category. Consider the below image:
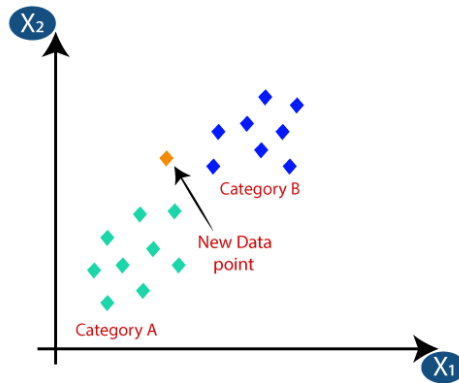

FIGURE 3.9 NEW DATA POINT

Firstly, choose the number of neighbors as k=5.

Next, calculate the **Euclidean distance** between the data points. The Euclidean distance is the distance between two points, which is already studied in geometry. It can be calculated as:

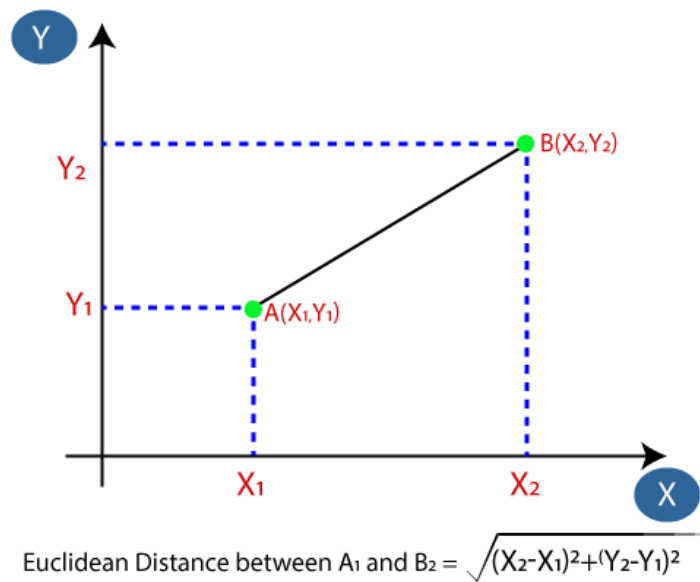Euclidean Distance between A_1 and $B_2 = \sqrt{(X_2-X_1)^2+(Y_2-Y_1)^2}$

FIGURE 3.10 CALCULATION OF EUCLIDEAN DISTANCE

By calculating the Euclidean distance, the nearest neighbors are calculated as three nearest neighbors in category A and two nearest neighbors in category B. Consider the below image:

FIGURE 3.11 DETERMINING CATEGORY OF NEW DATA POINT

As there are 3 nearest neighbors are from category A, hence this new data point must belong to category A.

3.4.4 PARAMETER SELECTION

The best choice of k depends upon the data. Generally, larger values of k reduce the effect of noise on the classification, but make boundaries between classes less distinct. A good k can be selected by various heuristic techniques. The special case where the class is predicted to be the class of the closest training sample (i.e. when k = 1) is called the nearest neighbor algorithm.

The accuracy of the k-NN algorithm can be severely degraded by the presence of noisy or irrelevant features, or if the feature scales are not consistent with their importance. Much research effort has been put into selecting or scaling features to improve classification. A particularly popular approach is the use of evolutionary algorithms to optimize feature scaling. Another popular approach is to scale features by the mutual information of the training data with the training classes.

In binary (two class) classification problems, it is helpful to choose k to be an odd number as this avoids tied votes. One popular way of choosing the empirically optimal k in this setting is via bootstrap method.

3.4.5 ADVANTAGES AND DISADVANTAGES OF K-NN

ADVANTAGES

- It is simple to implement.
- It is robust to the noisy training data
- It can be more effective if the training data is large.

DISADVANTAGES

- Always needs to determine the value of K which may be complex some time.
- The computation cost is high because of calculating the distance between the data points for all the training samples.

3.4.6 APPLICATIONS

1. KNN can be used for Recommendation Systems. Although in the real world, more sophisticated algorithms are used for the recommendation system. KNN is not suitable for high dimensional data, but KNN is an excellent baseline approach for the systems. Many companies make a personalized recommendation for its consumers, such as Netflix, Amazon, YouTube, and many more.

2. KNN can search for semantically similar documents. Each document is considered as a vector. If documents are close to each other, that means the documents contain identical topics.

3. KNN can be effectively used in detecting outliers. One such example is Credit Card fraud detection.

CHAPTER IV

CONVOLUTIONAL NEURAL NETWORKS (CNN)

4.1 INTRODUCTION

In the last few years of the IT industry, there has been a huge demand for once particular skill set known as Deep Learning. Deep learning is a subset of Machine Learning which consists of algorithms that are inspired by the functioning of the human brain or the neural networks. These structures are called as Neural Networks. It teaches the computer to do what naturally comes to humans. Deep learning, there are several types of models such as the Artificial Neural Networks (ANN), Auto encoders, Recurrent Neural Networks (RNN) and Reinforcement Learning. But there has been one particular model that has contributed a lot in the field of computer vision and image analysis which is the Convolutional Neural Networks (CNN) or the ConvNets.

CNNs are a class of Deep Neural Networks that can recognize and classify particular features from images and are widely used for analysing visual images. Their applications range from image and video recognition, image classification, medical image analysis, computer vision and natural language processing.

The term 'Convolution" in CNN denotes the mathematical function of convolution which is a special kind of linear operation wherein two functions are multiplied to produce a third function which expresses how the shape of one function is modified by the other. In simple terms, two images which can be represented as matrices are multiplied to give an output that is used to extract features from the image.

4.2 DEFINITION

Convolutional Neural Networks are a type of Deep Learning Algorithm that take the image as an input and learn the various features of the image through filters. This allows them to learn the important objects present in the image, allowing them to discern one image from the other.

For example, the convolutional network will learn the specific features of cats that differentiate from the dogs so that when if input image is provided it can easily differentiate between the two. One important feature of Convolutional Neural Network that sets it apart from other Machine Learning algorithms is its ability to pre-process the data by itself. Thus, spending time for a lot of resources in data pre-processing is not needed. During cold-start, the filters may require hand engineering but with progress in training, they are able to adapt to the learned features and develop filters of their own. Therefore, CNN is continuously evolving with growth in the data.

4.3 CNN ARCHITECTURE

There are two main parts to CNN architecture

- A convolution tool that separates and identifies the various features of the image for analysis in a process called as Feature Extraction
- A fully connected layer that utilizes the output from the convolution process and predicts the class of the image based on the features extracted in previous stages.

There are three types of layers that make up the CNN which are the convolutional layers, pooling layers, and fully-connected (FC) layers. When these layers are stacked, CNN architecture will be formed. In addition to these three layers, there are two more important parameters which are the dropout layer and the activation function which are defined below.
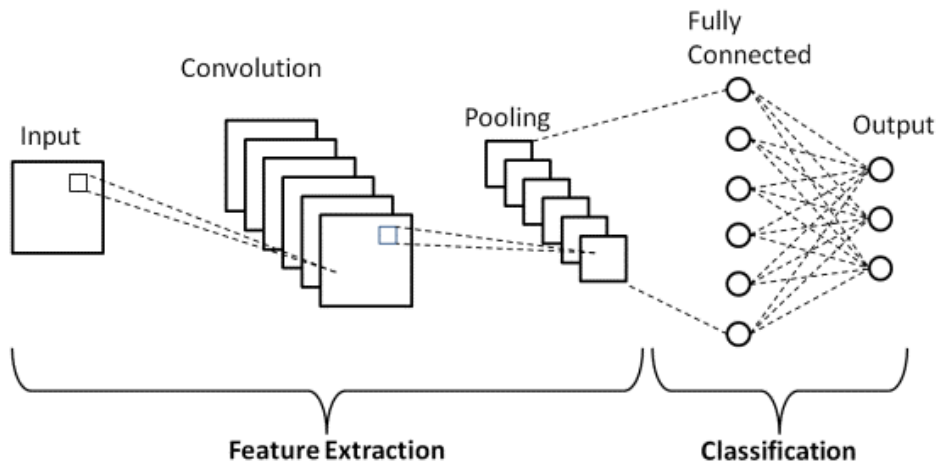
FIGURE 4.1 BASIC ARCHITECTURE OF CNN

4.3.1 CONVOLUTIONAL LAYER

This layer is the first layer that is used to extract the various features from the input images. In this layer, the mathematical operation of convolution is performed between the input image and a filter of a particular size MxM. By sliding the filter over the input image, the dot product is taken between the filter and the parts of the input image with respect to the size of the filter (MxM).

The output is termed as the Feature map which gives us information about the image such as the corners and edges. Later, this feature map is fed to other layers to learn several other features of the input image.
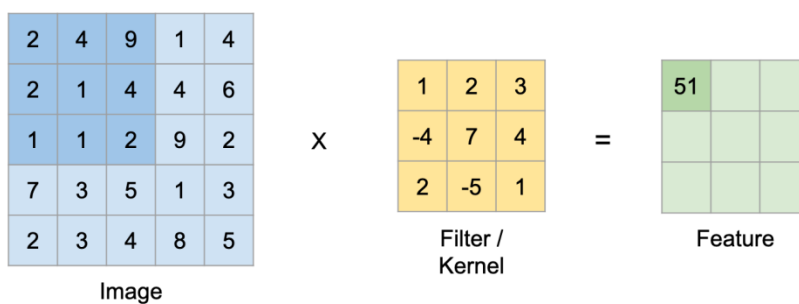

FIGURE 4.2 CONVOLUTIONAL LAYER

There a few things to learn from layer 1 that is striding and padding.

PADDING

While applying convolutions it will not obtain the output dimensions same as the input and the data will be lost over borders. So append a border of zerosand recalculate the convolution covering all the input values.
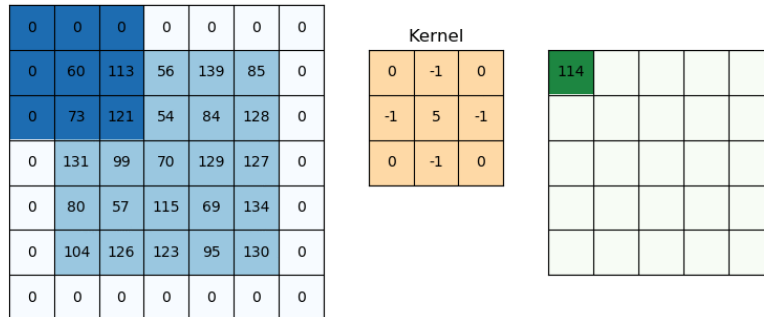

FIGURE 4.3 PADDING

STRIDING

Sometimes there is no need to capture all the data or information available so skip some neighbouring cells and visualise it,
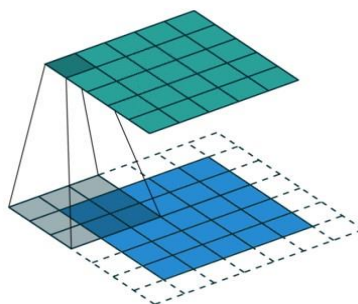

FIGURE 4.4 STRIDING

4.3.2 POOLING

In most cases, a Convolutional Layer is followed by a Pooling Layer. The primary aim of this layer is to decrease the size of the convolved feature map to reduce the computational costs. This is performed by decreasing the connections between layers and independently operates on each feature map. Depending upon method used, there are several types of pooling operations.

In Max Pooling, the largest element is taken from feature map. Average Pooling calculates the average of the elements in a predefined sized Image section. The total sum of the elements in the predefined section is computed in Sum Pooling. The Pooling Layer usually serves as a bridge between the Convolutional Layer and the FC Layer.
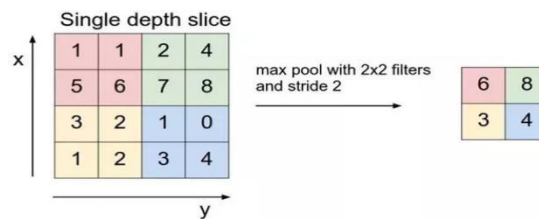
FIGURE 4.5 POOLING

4.3.3 FULLY CONNECTED LAYER

The Fully Connected (FC) layer consists of the weights and biases along with the neurons and is used to connect the neurons between two different layers. These layers are usually placed before the output layer and form the last few layers of a CNN Architecture.

In this, the input image from the previous layers are flattened and fed to the FC layer. The flattened vector then undergoes few more FC layers where the mathematical functions operations usually take place. In this stage, the classification process begins to take place.

4.3.4 DROPOUT

Usually, when all the features are connected to the FC layer, it can cause over fitting in the training dataset. Over fitting occurs when a particular model works so well on the training data causing a negative impact in the model's performance when used on a new data.

To overcome this problem, a dropout layer is utilised wherein a few neurons are dropped from the neural network during training process resulting in reduced size of the model. On passing a dropout of 0.3, 30% of the nodes are dropped out randomly from the neural network.

4.3.5 ACTIVATION FUNCTIONS

Finally, one of the most important parameters of the CNN model is the activation function. They are used to learn and approximate any kind of continuous and complex relationship between variables of the network. In simple words, it decides which information of the model should fire in the forward direction and which ones should not at the end of the network.

It adds non-linearity to the network. There are several commonly used activation functions such as the ReLU, Softmax, tanH and the Sigmoid functions. Each of these functions has a specific usage. For a binary classification CNN model, sigmoid and Softmax functions are preferred and for a multi-class classification, generally Softmax us used.

4.4 ADVANTAGES AND DISADVANTAGES
ADVANTAGES
- Very High accuracy in image recognition problems.
- Automatically detects the important features without any human supervision.
- Weight sharing.

DISADVANTAGES

- CNN do not encode the position and orientation of object.

- Lack of ability to be spatially invariant to the input data.

- Lots of training data is required

4.5 APPLICATIONS

- Convolutional Neural Networks or CNNs were developed for image recognition and therefore, are mostly in the field of computer vision where they are used for classifying images, segmenting them and also performing localization on the images.

- Videos are different from images in the sense that they have a temporal dimension. While more complicated than images, they can tweak these CNNs to accommodate these types of streaming visual inputs. Most frequently, convolutional networks are becoming highly popular with other algorithms like LSTM and Boltzmann Machines for boosting their performance at handling video input.

- Apart from visual inputs, CNNs are also being utilized in the field of Natural Language Processing for semantic parsing, sentence modeling, prediction as well as classification.

- More recently, companies like Google have been using CNNs alongside Recurrent Neural Networks and LSTMs for speech recognition.

- CNNs are also being used in drug discovery where they prove to be an efficient tool for identifying the interaction between the molecules and the biological proteins for the identification of potential treatments.

4.6 SVM BASED CNN

In this section the architecture is based on CNN and SVM, where in CNN is considered as a deep learning algorithm, on which the dropout technique has been applied during training. Our proposed system was tailored by altering the trainable classifier of the CNN with an SVM classifier. Our target is to mix the CNN respective capacities and the SVM to obtain a new effectual recognition system inspired by the two formalisms. Firstly, the first layer welcomes raw image pixels as input. Secondly, the second and fourth layer of the network is convolution layers

alternator with sub-sampling layers, which take the pooled maps as input. Consequently, they are able to extract features that are more and more invariant to local transformations of the input layer. FCL is the sixth layer which consists of N neurons. The final layer was substituted by SVM with a linear kernel for classification. Because of using a huge number of data and parameters, over-fitting can occur. So to prevent our network from this problem and to improve it, dropout is applied. This technique consists of temporarily removing a unit from the network. This removed unit is randomly selected only during the training. Dropout is applied only at FCL layer and for more precisely, it is applied to feed-forward connections (perceptron). This choice is based on the fact that since the convolutional layers don't have a lot of parameters, over-fitting is not a problem and therefore dropout would not have much effect. The outputs from the hidden units are taken by the SVM as a feature vector for the training process. After that, the training stage continues till realizing good trained. Finally, classification on the test set was performed by the SVM classifier with such automatically extracted features.

CHAPTER V

SOFTWARE TOOL USED

5.1 INTRODUCTION

Google Colab is a great platform for deep learning enthusiasts, and it can also be used to test basic machine learning models, gain experience, and develop an intuition about deep learning aspects such as hyper parameter tuning, preprocessing data, model complexity, over fitting and more.

Colaboratory by Google (Google Colab in short) is a Jupyter notebook based runtime environment which allows everyone to run code entirely on the cloud. This is necessary because it means that all can train large scale ML and DL models even if anyone don't have access to a powerful machine or a high speed internet access.

Google Colab supports both GPU and TPU instances, which makes it a perfect tool for deep learning and data analytics enthusiasts because of computational limitations on local machines. Since a Colab notebook can be accessed remotely from any machine through a browser, it's well suited for commercial purposes as well.

5.2 DOWNLOADING A DATASET

When training a machine learning model on the local machine, where likely to have trouble with the storage and bandwidth costs that come with downloading and storing the dataset required for training a model. Deep learning datasets can be massive in size, ranging between 20 to 50 Gb. Downloading them is most challenging in a developing country, where getting high-speed internet isn't possible.

The most efficient way to use datasets is to use a cloud interface to download them, rather than manually uploading the dataset from a local machine. Thankfully, Colab gives us a variety of ways to download the dataset from common data hosting platforms.

5.3 PYTHON

Python is an interpreted high-level, general-purpose programming language. Python's design philosophy emphasizes code readability with its notable use of significant indentation. Its language constructs and object-oriented approach aim to help programmers write clear, logical codes for small and large-scale projects. Python is dynamically typed and garbage-collected. It supports multiple programming paradigms, including structured (mainly procedural), object-oriented and functional programming. Python is often described as a "batteries included" language due to its comprehensive standard library.

Guido van Rossum began working on Python in the late 1980s as a successor to the ABC programming language and first released it in 1991 as Python 0.9.0. Python 2.0 was released in 2000 and introduced new features, such as list comprehensions and a garbage 7 collection system using reference counting. Python 3.0 was released in 2008 and was a significant revision of the language that is not entirely backwards compatible, and much Python 2 code does not run unmodified on Python 3. Python 2 was discontinued version 2.7.18 in 2020. Python is a multi-paradigm programming language. Object-oriented programming and structured programming are fully supported, and many of its features support functional programming and aspect-oriented programming (including meta programming and meta objects (magic methods)). Many other paradigms are supported via extensions, including design by contract and logic programming.

5.4 FEAUTURES OF COLAB

- Write and execute code in Python.
- Document your code that supports mathematical equations.
- Create/Upload/Share notebooks.
- Import/Save notebooks from/to Google Drive.
- Import/Publish notebooks from GitHub.
- Import external datasets e.g. from Kaggle.
- Integrate PyTorch, Tensor Flow, Keras, and OpenCV.
- Free Cloud service with free GPU.

One of the main **benefits** of using **Colab** is that it has most of the common libraries that are needed for machine learning like **Tensor Flow**, Keras, Scikit Learn, OpenCV, numpy, pandas, etc. pre-installed.

Machine learning is a complex discipline. But implementing machine learning models is far less daunting and difficult than it used to be, thanks to machine learning framework such as Google's Tensor Flow that ease the process of acquiring data, training models, serving predictions, and refining future results.

Created by the Google Brain team, Tensor Flow is an open source library for numerical computation and large-scale machine learning. Tensor Flow bundles together a slew of machine learning and deep learning (aka neural networking) models and algorithms and makes them useful by way of a common metaphor.

Tensor Flow can train and run deep neural networks for handwritten digit classification, image recognition, word embedding's, recurrent neural networks, sequence-to-sequence models for machine translation, natural language processing, and PDE (partial differential equation) based simulations. Best of all, Tensor Flow supports production prediction at scale, with the same models used for training.

With Colab all can import an image dataset, train an image classifier on it, and evaluate the model, all in just a few lines of code. Colab notebooks execute code on

Google's cloud servers, meaning all can leverage the power of Google hardware, including GPUs and TPUs, regardless of the power of your machine. All that need is just a browser.

Colab is used extensively in the machine learning community with applications including:

- Getting started with Tensor Flow
- Developing and training neural networks
- Experimenting with TPUs
- Disseminating AI research
- Creating tutorials

5.4.1 CHARTING IN COLABORATORY

A common use for notebooks is data visualization using charts. Colaboratory makes this easy with several charting tools available as Python imports. Matplotlib is the most common charting package.

5.4.2 GOOGLE DRIVE

Files can be accessed in Drive in a number of ways, including:

- Mounting your Google Drive in the runtime's virtual machine
- Using a wrapper around the API such as PyDrive
- Using the native REST API.

5.4.3 EASY SHARING

Just like other Google services, all can share the results of the work with others just like using Google Docs. But it should be noted, that it cannot be edited by more than 1 person on one notebook at the same time. And one more thing, Google will provide additional RAM if the current RAM is not enough.

Google Colab has a 'maximum lifetime' limit of running notebooks that is 12 hours with the browser open, and the 'Idle' notebook instance is interrupted after 90 minutes. In addition to that all can have a maximum of 2 notebooks running simultaneously. If the notebook window is closed and opened again while the instance is still running, the output and cell variables will still be present. But if the notebook instance has been recycled, output and cell variables are no longer available.

CHAPTER VI

PROJECT DESCRIPTION

In this project MSTAR Database images which comprise images of 10 different classes has taken. First the images are divided into training and testing datasets. Next the datasets are imported into Google drive to use the images in Google Colab. For studying the classification step different algorithms were used which are Support Vector Machines (SVM), K-Nearest Neighbors (K-NN) and Convolutional Neural Networks (CNN) algorithms along with SVM based CNN. SVM and K-NN are implemented and the parameters are evaluated for both the techniques. Next CNN and SVM based CNN are implemented and the results are noted. At last all the performance measures are compared and the maximum accuracy is obtained for SVM based CNN technique for MSTAR Database.

6.1 DATASET FEAUTURES

In this study, the moving and stationary target acquisition and recognition (MSTAR) database is used. This is a public dataset of SAR images of 0.3m by 0.3m resolution. The images were taken over 360 degrees covering various target orientations. Dataset was obtained in two acquisitions with 2536 and 2636 samples respectively (10 classes). Fig. 6.2 illustrates corresponding sample examples from each class. Image samples have 128*128 sizes with centred targets.
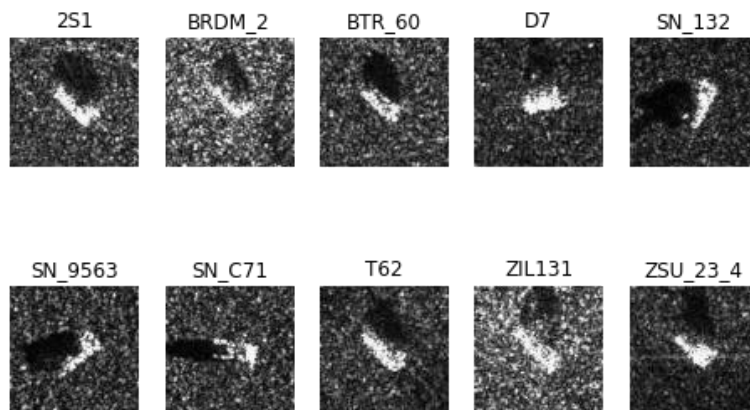

FIGURE 6.1 Images of 10 targets from MSTAR database.

6.2 ALGORITHM:

- STEP-1: Read the training dataset and testing dataset into the notebook from Google Drive.

- STEP-2: Install and import the required libraries and packages for the classification.

- STEP-3: Resize the images to required height and width.

- STEP -4: Shuffle the images in the database.

- STEP-5: Define the models for SVM, CNN and K-NN.

- STEP-6: Compile all the models, for CNN and SVM based CNN run the models up to 200 epochs.

- STEP-7: Generate training accuracies for all the models.

- STEP-8: Generate Confusion matrix for all the models.

- STEP-9: Generate the Classification report for all the models.

- STEP-10: Generate the estimated image and true images for plotting error and true predictions and the performance characteristics are observed.
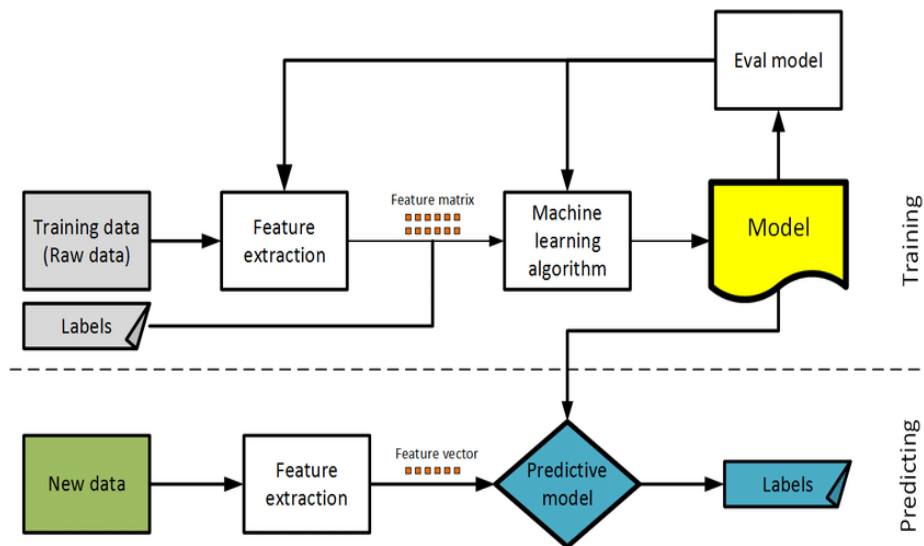
6.3 FLOW CHARTS

FIGURE 6.2 FLOW CHART FOR SUPERVISED LEARNING
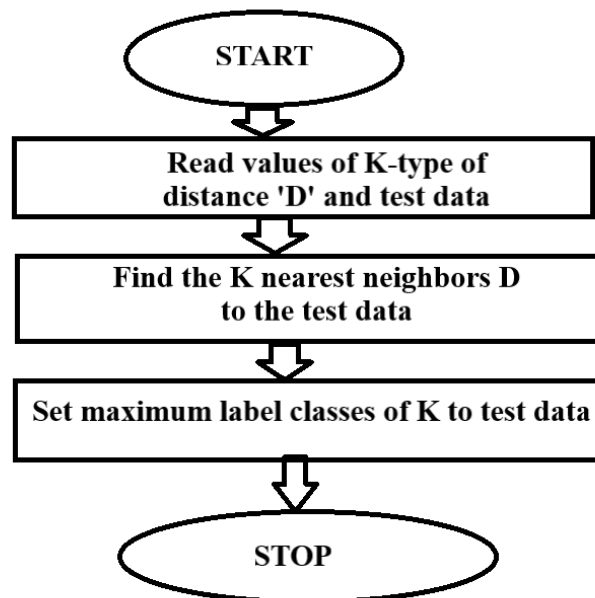

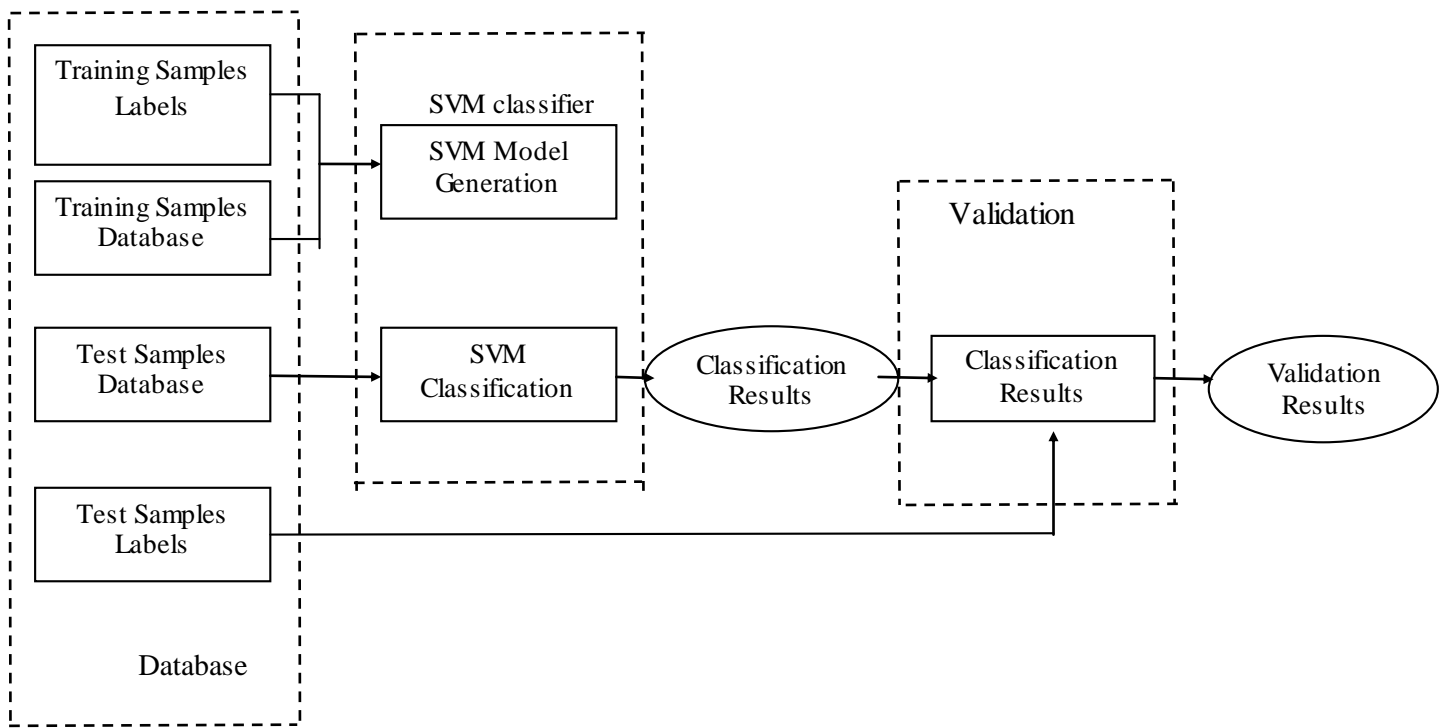FIGURE 6.3 K-NEAREST NEIGHBORS (K-NN) FLOW CHART

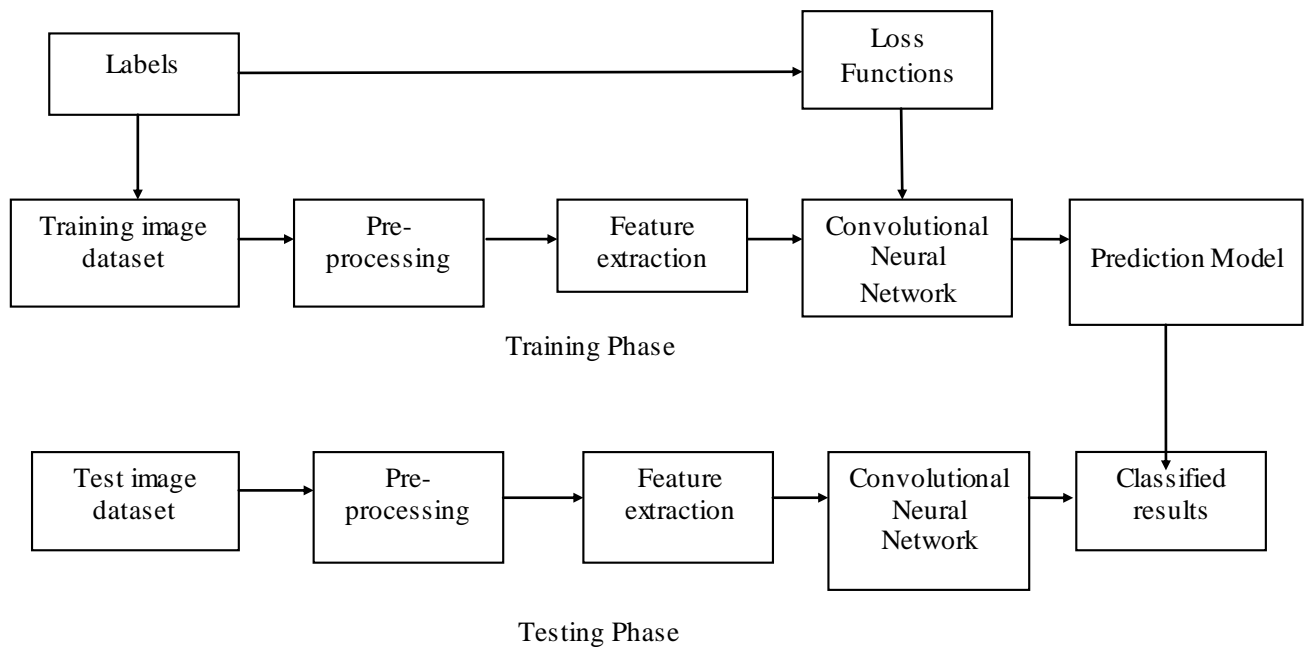FIGURE 6.4 SUPPORT VECTOR MACHINES (SVM) FLOW CHART

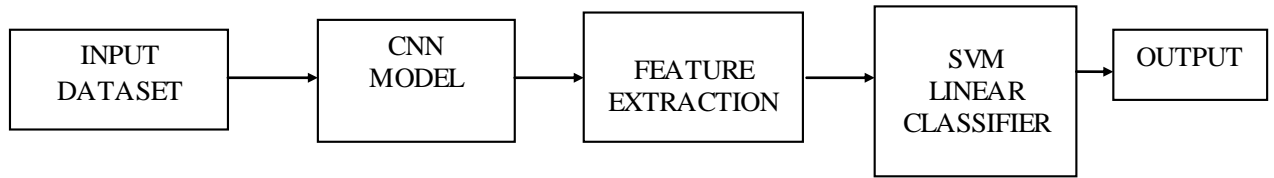FIGURE 6.5 CONVOLUTIONAL NEURAL NETWORK (CNN) FLOW CHART

FIGURE 6.6 SVM based CNN FLOW CHART

6.4 PERFPRMANCE MEASURES

6.4.1 CONFUSION MATRIX

. A confusion matrix is a table that is often used to describe the performance of a classification model on a set of test data for which the true values are known.

FIGURE 6.7 CONFUSION MATRIX

TERMS ASSOCIATES WITH CONFUSION MATRIX

True Positives (TP) - These are the correctly predicted positive values which mean that the value of actual class is yes and the value of predicted class is also yes. E.g. if actual class value indicates that this passenger survived and predicted class tells the same thing.

True Negatives (TN) - These are the correctly predicted negative values which means that the value of actual class is no and value of predicted class is also no.

E.g. if actual class says this passenger did not survive and predicted class tells the same thing.

False positives and false negatives, these values occur when your actual class contradicts with the predicted class.

False Positives (FP) – When actual class is no and predicted class is yes. E.g. if actual class says this passenger did not survive but predicted class tells that this passenger will survive.

False Negatives (FN) – When actual class is yes but predicted class in no. E.g. if actual class value indicates that this passenger survived and predicted class tells that passenger will die.

After understanding these four parameters calculating Accuracy, Precision, Recall and F1 score becomes easy.

6.4.2 Accuracy

Accuracy is the most intuitive performance measure and it is simply a ratio of correctly predicted observation to the total observations. One may think that, if high accuracy is attained then our model is best. Yes, accuracy is a great measure but only when there are symmetric datasets where values of false positive and false negatives are almost same. Therefore other parameters are to be considered to evaluate the performance of taken model.

Accuracy = TP+TN/TP+FP+FN+TN

6.4.3 PRECISION

Precision is the ratio of correctly predicted positive observations to the total predicted positive observations.

Precision = TP/TP+FP

6.4.4 RECALL (SENSITIVITY)

Recall is the ratio of correctly predicted positive observations to the all observations in actual class - yes.

Recall = TP/TP+FN

6.4.5 F1 SCORE

F1 Score is the weighted average of Precision and Recall. Therefore, this score takes both false positives and false negatives into account. Intuitively it is not as easy to understand as accuracy, but F1 is usually more useful than accuracy, especially if there is an uneven class distribution. Accuracy works best if false positives and false negatives have similar cost. If the cost of false positives and false negatives are very different, it's better to look at both Precision and Recall.

F1 Score = 2*(Recall * Precision) / (Recall + Precision)

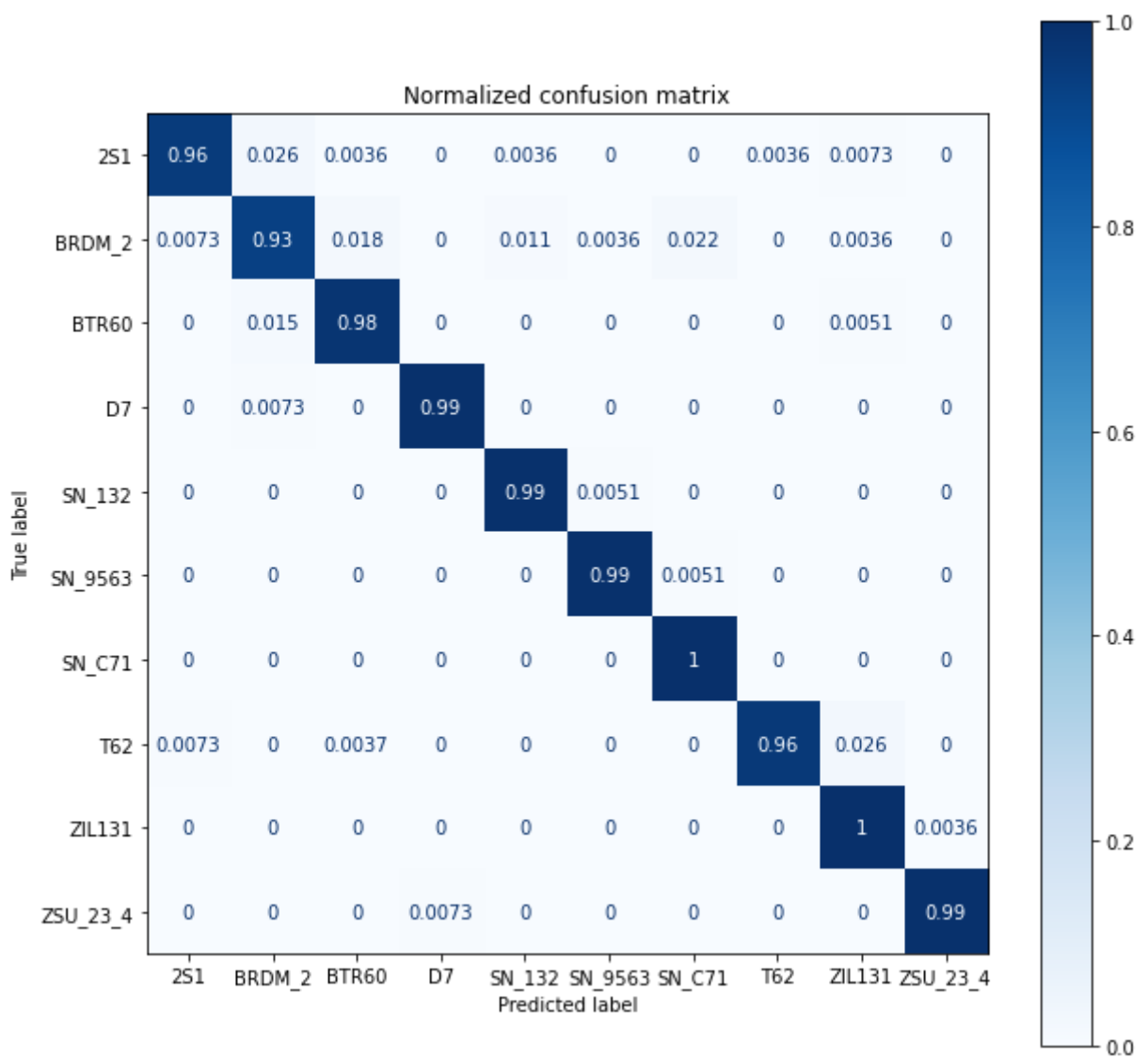# CHAPTER VII

RESULTS AND CONCLUSION

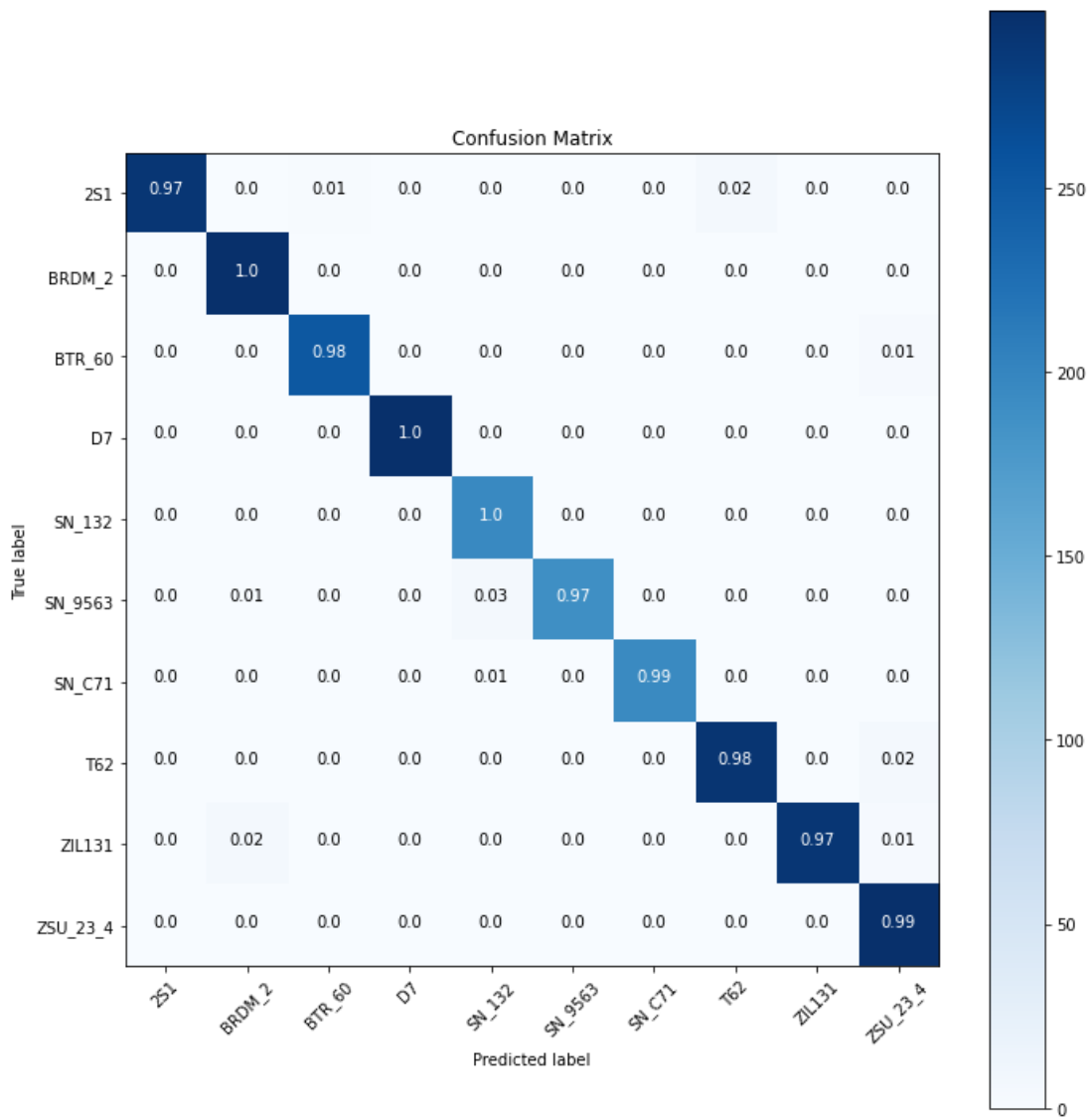## 7.1 CONFUSION MATRIX
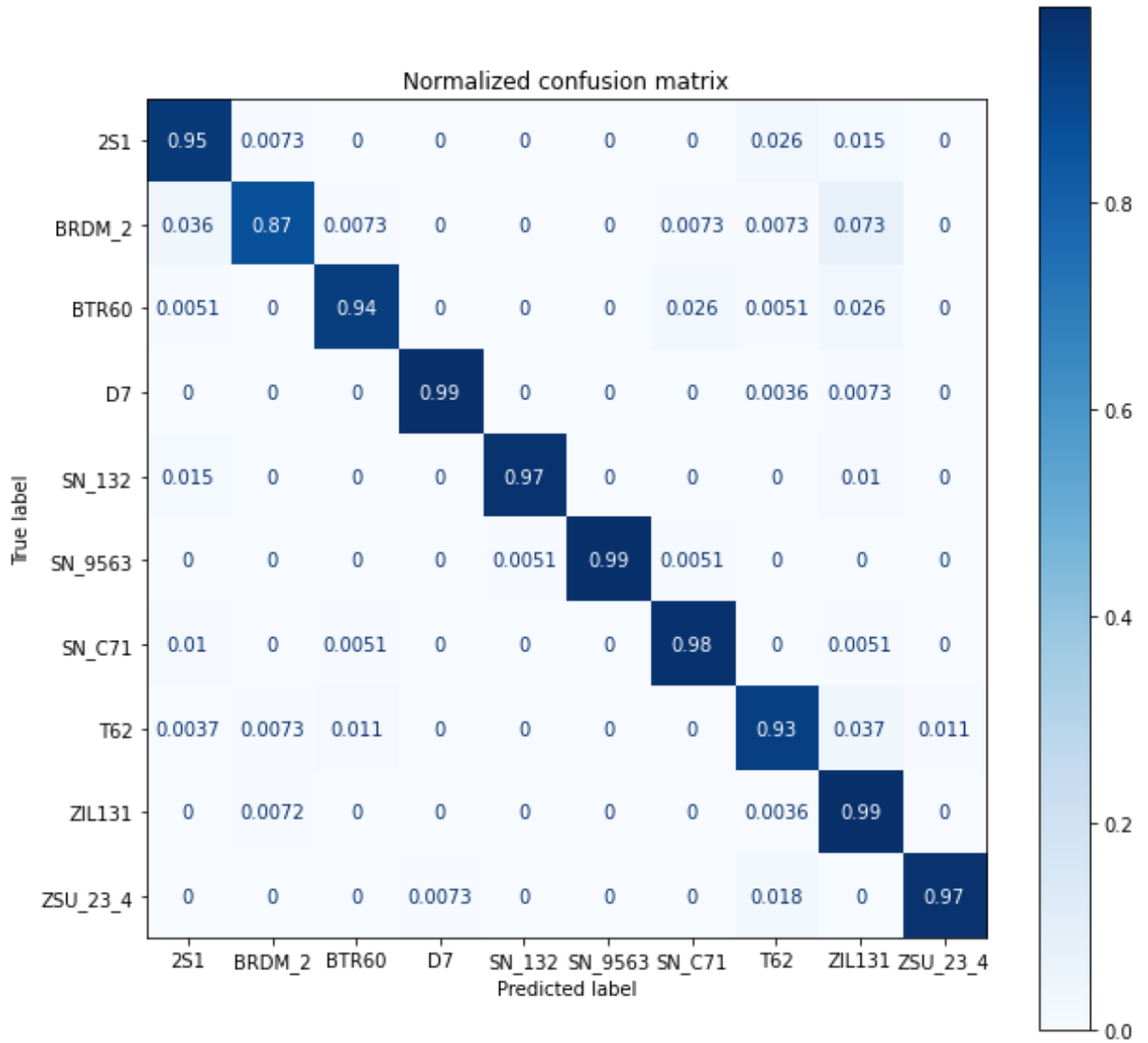
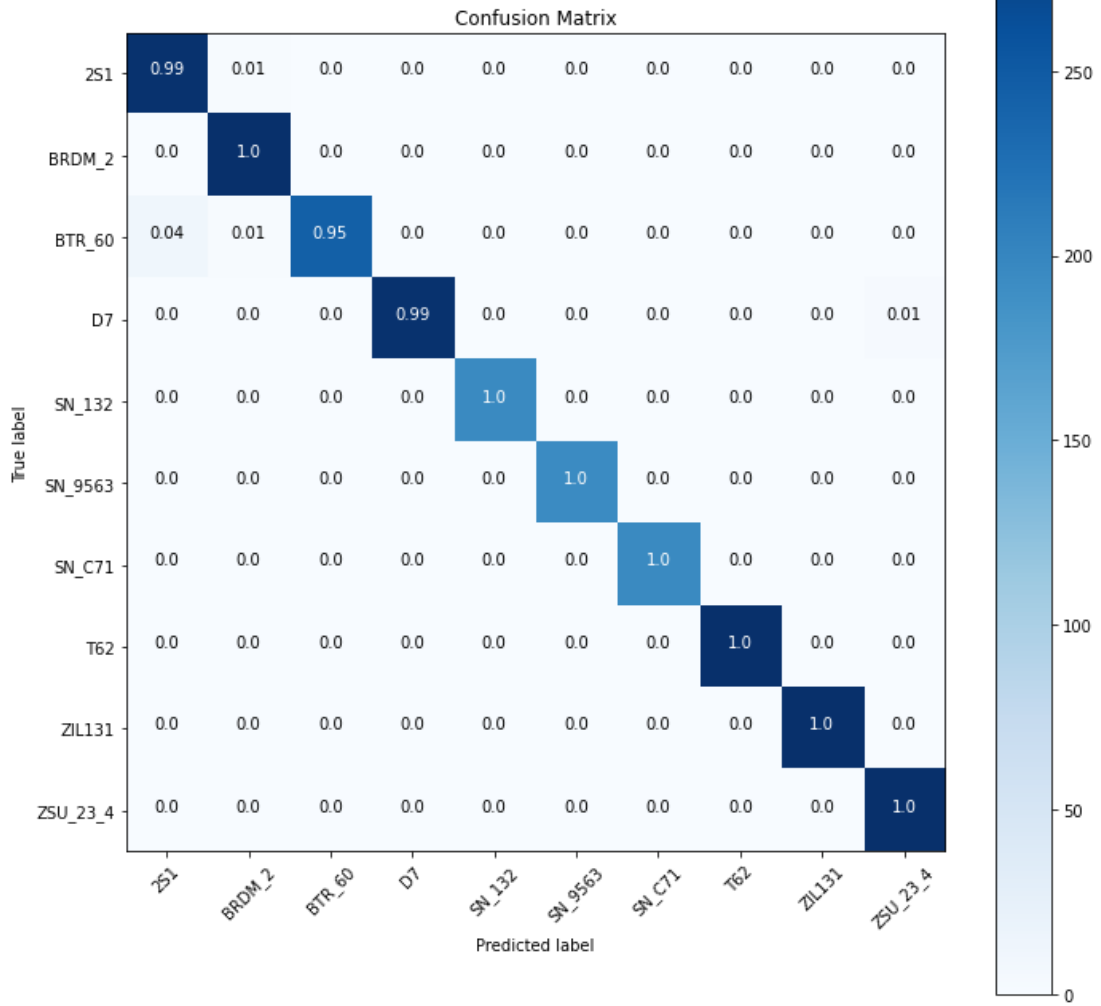K-NEAREST NEIGHBORS (K-NN) CONFUSION MATRIX

Normalized confusion matrix

SUPPORT VECTOR MACHINES (SVM) CONFUSION MATRIX
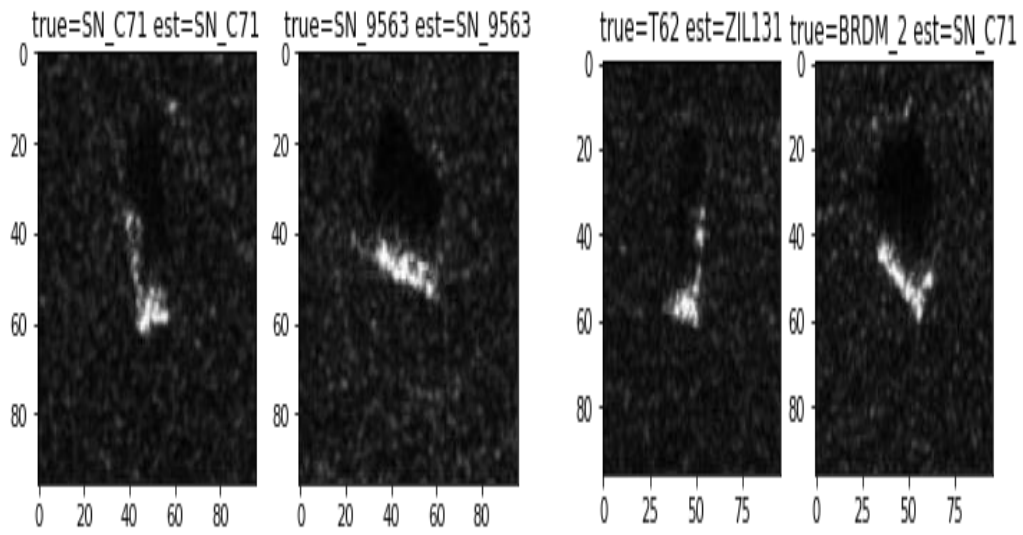
Confusion Matrix

CONVOLUTIONAL NEURAL NETWORKS (CNN) CONFUSION MATRIX

Confusion Matrix

SVM BASED CNN CONFUSION MATRIX
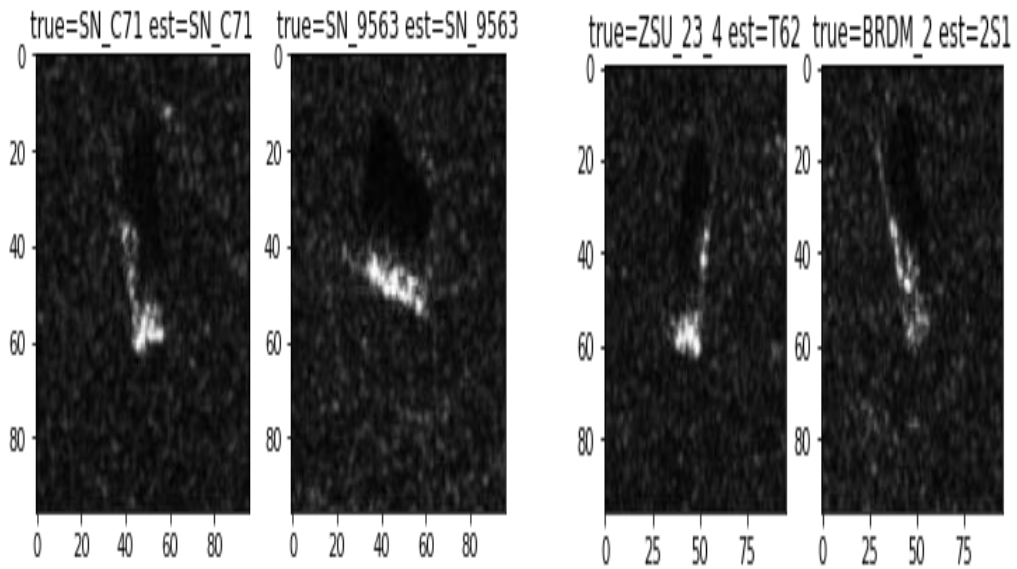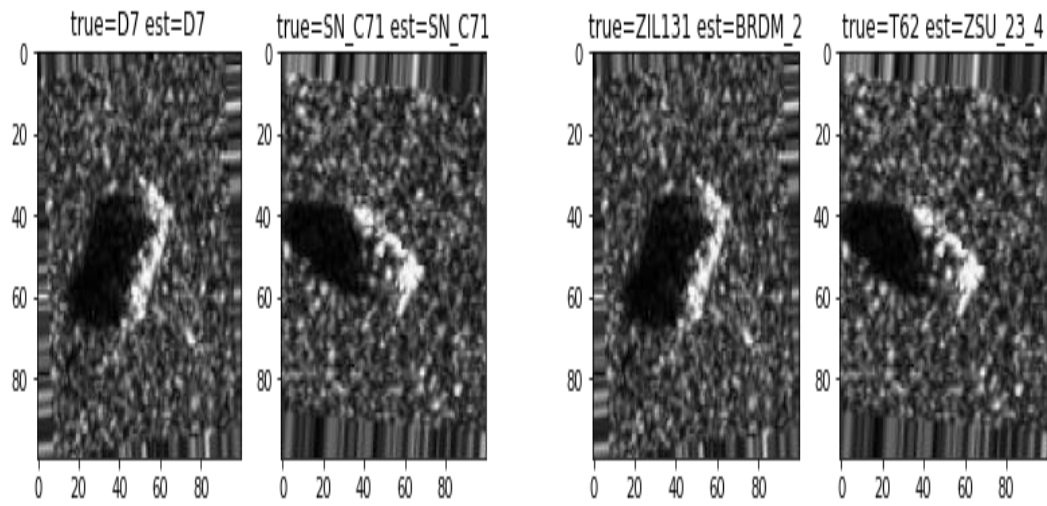
7.2 ESTIMATED AND TRUE IMAGES

SUPPORT VECTOR MACHINES (SVM) OUTPUT IMAGES

K- NEAREST NEOGHBORS (K-NN) OUTPUT MAGES

true=D7 est=D7   true=SN_C71 est=SN_C71   true=ZIL131 est=BRDM_2   true=T62 est=ZSU_23_4

CONVOLUTIONAL NEURAL NETWORKS (CNN) OUTPUT IMAGES

true=ZIL131 est=ZIL131   true=T62 est=T62   true=BTR_60 est=2S1   true=BTR_60 est=2S1
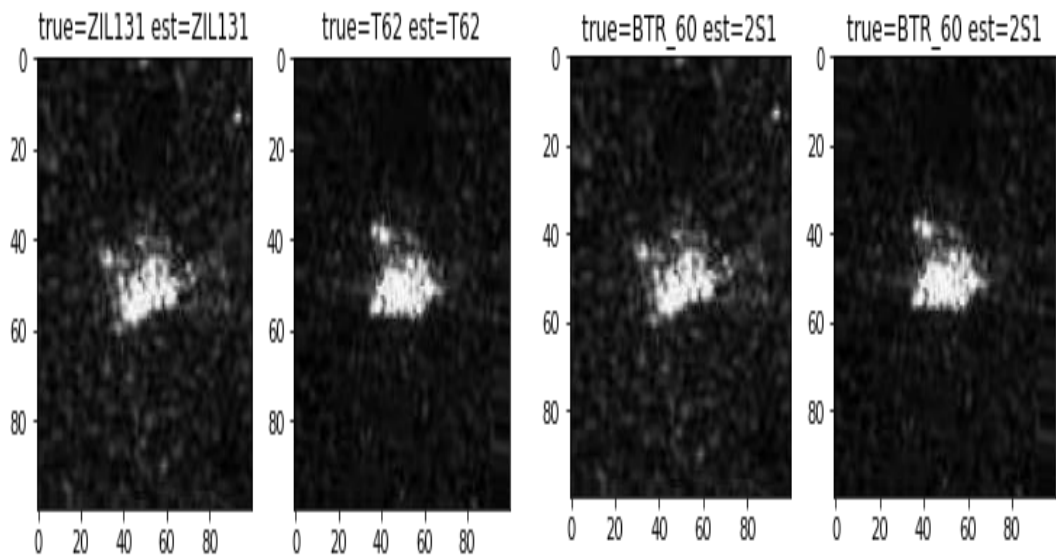
SVM BASEDCNN OUTPUT IMAGES

7.3 OUTPUT COMPARISION TABLE

Technique	SVM BASED CNN	CONVOLUTIONAL NEURAL NETWORKS(CNN)	SUPPORT VECTOR MACHINES(SVM)	K-NEAREST NEIGHBORS (K-NN)
Accuracy	99.62	99.22	97.9	95.7
Precision	0.99	0.99	0.98	0.96
Recall	0.99	0.99	0.98	0.96
F1-Score	0.99	0.99	0.98	0.96

CONCLUSION

In this paper, K-Nearest Neighbors (K-NN), Support Vector Machines (SVM), Convolutional Neural Networks (CNN) and SVM based CNN are implemented on MSTAR dataset. KNN output completely relies on nearest neighbors, which may not be ideal in all cases. SVM output depends on the hyper plane in which the data is divided into classes. On the other hand, CNN extract the features from the input data, which are very helpful for analysis. SVM based CNN uses both activation function and kernels which results in high accuracy. From the simulation results, SVM based CNN has the highest accuracy (99.62%) compared to CNN (99.2%), SVM (97.9%) KNN (95.7 %.).

REFERENCES

[1] Comparative Analysis of Convolutional Neural Networks and Support Vector Machines for Automatic Target Recognition Ievgen M. Gorovyi, Dmytro S. Sharapov, Proceedings of the 2017 Microwaves, Radar and Remote Sensing Symposium (August 29-31, Kiev, Ukraine), pp.63-66 (2017)

[2] D. Dugeon and R. Lacoss, "An overview of automatic target recognition". Lincoln Laboratory Journal, Vol. 6, No.1, pp. 3–9, 1993.

[3] Maha Al Mufti, Esra Al Hadhrami Emirates Technology and Innovation Center (ETIC) Khalifa University of Science and Technology Abu Dhabi, United Arab Emirates, "Automatic Target Recognition in SAR Images, Comparison Between Pre-trained CNNs in a Transfer Learning Based Approach", 2018 International conference on Artificial Intelligence on big data.

[4] A. Eryildirim, Methods for automatic target classification in radar. Master thesis, Ankara, Bilkent University, 76 p, 2009.

[5] J. Yue, W. Zhao, S. Mao, and H. Liu, "Spectral–spatial classification of hyper spectral images using deep convolutional neural networks," Remote Sens. Lett., vol. 6, no. 6, pp. 468–477, 2015.

[6] Khalid El-Darymli∗, "Automatic Target Recognition inSynthetic Aperture Radar Imagery: A State-of-the-Art Review", Member, IEEE, Eric W. Gill† , Senior Member, IEEE, Peter McGuire‡ , Desmond Power§ , Member, IEEE, Cecilia Moloney¶ , Member, IEEE.

[7] Y. Yang, Y. Qiu and C. Lu, "Automatic target classification experiments on the MSTAR SAR images". Proceedings of the 6th International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing and 1st ACIS International Workshop on Self-Assembling Wireless Networks; pp. 2–7, 2005.

[8] Adugna G. Mullissa, Claudio Persello, Valentyn Tolpekin, Faculty of Geoinformation Science and Earth Observation of the University of TwenteEnschede, the Netherlands "FULLY CONVOLUTIONAL NETWORKS FOR MULTI-TEMPORAL SAR IMAGE CLASSIFICATION".

[9] Priyanka G, Gandhi Raj R, Somar K. P " Target Recognition in SAR using Deep Learning" 2019 2nd International Conference on Intelligent Computing, Instrumentation and Control Technologies (ICICICT).

[10] ShwetaTaneja, Charu Gupta, SakshiAggarwal, Veni Jindal, IT Department, BhagwanParshuram Institute of Technology, Guru Gobind Singh Indraprastha University New Delhi, India "MFZ-KNN-A Modified Fuzzy Based K Nearest Neighbor Algorithm".

[11] 2010 2nd International Asia Conference on Informatics in Control, Automation and Robotics, "A Clustering-Based KNN Improved Algorithm CLKNN for Text Classification".

[12] Jing Chai *, Hongwei Liu, ZhengBao "A W-KNN CLASSIFIER TO IMPROVE RADAR OUTLIER REJECTION PERFORMANCE"

[13] Shengna WU, Ke WANG*, "Study on Small Samples SAR Image Recognition Detection Method Based on CNN", Yiwen OUYANG School of Information Science and Engineering Henan University of Technology Zhengzhou, China.

[14] Yuan Zhang, Yang Song*, Yanping Wang, Hongquan Qu North China University of Technology Beijing, China. "A fast training method for SAR based on CNN for targets recognition", 2018 11th International Congress on Image and Signal Processing, Biomedical Engineering and Informatics (CISP-BMEI 2018)

[15] C.-C. Chang and C.-J. Lin. LIBSVM: a library for support vector machines. ACM Transactions on Intelligent Systems and Technology, 2:27:1--27:27, 2011.