Methodology

3. METHODOLOGY

The primary objective of the present research work is to classify TIHC images using multiple classifiers. Three classifiers were selected, namely, Neural Networks, Support Vector Machine and K Nearest Neighbor. The working of individual classifiers and the fusion mechanism followed are explained in this chapter.

As mentioned earlier, in multiple-classifier systems the outputs of a set of classifiers are combined to produce the final classification decision. The accuracy of the classifier combination depends on the following properties.

1) Classifier Details
    a) The number of individual classifiers used.
    b) The type of the individual classifiers. Some combination scheme use classifiers of the same types, e.g., Neural Networks, linear classifiers, Nearest Neighbor classifiers and other schemes use sets of different classifier models.
2) The feature vectors used by the individual classifiers.
3) Techniques used during fusion classification
    a) Partitioning method (Training and Testing sets)
    b) The aggregation method
4) Type of training

Usually the individual classifiers are chosen ad hoc on the basis of their accuracy (the higher the better) and application. Each of the above properties is discussed in the following sections.

3.1. CLASSIFIERS

The present research work performs fusion classification in two ways. The first method uses two classifiers and the second method combines three classifiers. Three classifiers are considered, namely, Neural Networks, Support Vector Machines and K Nearest Neighbor techniques.

3.1.1. Artificial Neural Networks

An Artificial Neural Network (ANN) is an information processing paradigm that is inspired by the way biological nervous systems, such as the brain, process information. The key element of the ANN paradigm is the novel structure of the information processing system. It is composed of a large number of highly interconnected processing elements (neurons) working in union to solve specific problems (Principe *et al.*, 2009). ANNs, like people, learn by example. An ANN is configured for a specific application, such as pattern recognition or data classification, through a learning process. Learning in biological systems involves adjustments to the synaptic connections that exist between the neurons. This is true of ANNs as well.

A Neural Network (NN), in the case of artificial neurons called Artificial Neural Network (ANN) or Simulated Neural Network (SNN), is an interconnected group of natural or artificial neurons that uses a mathematical or computational model for information processing based on a connectionist approach to computation. In most cases, an NN is an adaptive system that changes its structure based on external or internal information that flows through the network. In more practical terms Neural Networks are non-linear statistical data modelings or decision making tools. They can be used to model complex relationships between inputs and outputs or to find patterns in data. However, the paradigm of Neural Networks - i.e., implicit, and not explicit learning is stressed - seems more to correspond to some kind of natural intelligence than to the traditional Artificial

Intelligence, which would stress, instead, rule-based learning. Traditionally, the term Neural Network had been used to refer to a network or circuit of biological neurons (Michael, 1995). The modern usage of the term often refers to artificial Neural Networks, which are composed of artificial neurons or nodes.

A Neural Network is a powerful data modeling tool that is able to capture and represent complex input/output relationships. The motivation for the development of Neural Network technology stemmed from the desire to develop an artificial system that could perform "intelligent" tasks similar to those performed by the human brain. Neural Networks resemble the human brain in the following two ways: they acquire knowledge through learning, and the knowledge is stored within inter-neuron connection strengths known as synaptic weights (NeuroIntelligence-Alyuda Research, 2010). The true power and advantage of Neural Networks lies in their ability to represent both linear and non-linear relationships and in their ability to learn these relationships directly from the data being modeled. Traditional linear models are simply inadequate when it comes to modeling data that contains non-linear characteristics.

The most common Neural Network model is known as a Supervised Network because it requires a desired output in order to learn. The goal of this network type is to create a model that maps the input to the output using historical data so that the model can then be used to produce the output when the desired output is unknown.

Neural Networks, with their remarkable ability to derive meaning from complicated or imprecise data, can be used to extract patterns and detect trends that are too complex to be noticed by either humans or other computer techniques. A trained Neural Network can be thought of as an "expert" in the category of information it has been given to analyze. This expert can then be used to provide

projections given new situations of interest and answer "what if" questions. Other advantages include:

- **Adaptive learning**: An ability to learn how to do tasks based on the data given for training or initial experience.
- **Self-Organization**: An ANN can create its own organization or representation of the information it receives during learning time.
- **Real Time Operation**: ANN computations may be carried out in parallel, and special hardware devices are being designed and manufactured which take advantage of this capability.
- **Fault Tolerance via Redundant Information Coding**: Partial destruction of a network leads to the corresponding degradation of performance. However, some network capabilities may be retained even with major network damage.

In the present work, a Backpropagation Neural Network is used. It was first described by Bryson and Ho (1969) but gained recognition only after 1974 (Ethem, 2010).

A Back Propagation Neural Network (BPNN) is an artificial Neural Network where connections between the units do not form a directed cycle. They are the first simplest type of artificial Neural Network devised where, the information moves in only one direction, forward, from the input nodes, through the hidden nodes (if any) and to the output nodes. There are no cycles or loops in the network. The Back Propagation Algorithm is a common way of teaching artificial Neural Networks how to perform a given task. It requires training which has knowledge or can calculate the desired output for any given input. Backpropagation algorithm learns the weights for a multilayer network, given a network with a fixed set of units and interconnections. It employs gradient descendent rule to attempt to minimize the squared error between the network

output values and the target values for these outputs. The back propagation learning algorithm can be divided into two phases: propagation and weight update.

Phase 1: Propagation

Each Propagation involves the following steps:

1. Forward propagation of a training pattern's input through the Neural Network in order to generate the propagation's output activations.
2. Backward propagation of the propagation's output activations through the Neural Network using the training pattern's target in order to generate the deltas of all output and hidden neurons.

Phase 2: Weight update

For each weight-synapse

1. Multiply its output delta and input activation to get the gradient of the weight.
2. Bring the weight in the opposite direction of the gradient by subtracting a ratio of it from the weight.

This ratio influences the speed and quality of learning; it is called the learning rate. The sign of the gradient of a weight indicates where the error is increasing, this is why the weight must be updated in the opposite direction. Repeat the phase 1 and 2 until the performance of the network is good enough.

3.1.2. Support Vector Machine (SVM)

A Support Vector Machine (SVM) is a concept in computer science for a set of related supervised learning methods that analyze data and recognize patterns that are mainly used for classification and regression analysis. The standard SVM takes a set of input data and predicts, for each given input, which of two possible

classes the input is a member of, which makes the SVM a non-probabilistic binary linear classifier. Given a set of training examples, each marked as belonging to one of two categories, an SVM training algorithm builds a model that assigns new examples into one category or the other. An SVM model is a representation of the examples as points in space, mapped so that the examples of the separate categories are divided by a clear gap that is as wide as possible. New examples are then mapped into that same space and predicted to belong to a category based on which side of the gap they fall on.

Although SVM is originally designed as binary classifiers, approaches that address a multi-class problem as a single "all-together" optimization problem exist (Weston and Watkins, 1999), but are computationally much more expensive than solving several binary problems.

A multi-class classification task usually involves separating data into training and testing sets. Each instance in the training set contains one 'target value" (i.e. class labels) and several "attributes" (i.e. features). The goal of SVM is to produce a model (based on the training data) which predicts the target values of the test data given only the test data attributes. Mathematically SVM can be described as follows (Boser et al., 1992; Cortes and Vapnik, 1995)

Given a training set of instance-label pairs (x_i, y_i), where $i = 1, \ldots, l$ where $x_i \in R^n$ and $y \in \{1, -1\}^l$, the Support Vector Machines (SVM) require the solution of the following optimization problem :

$$\min_{w,\, b,\, \xi} \frac{1}{2} W^T W + C \sum_{i=1} \xi_i$$

$$\text{Subject to } y_i(w^T \varphi(X_i) + b) \geq 1 - \xi_i, \xi_i \geq 0. \qquad (3.1)$$

The training vectors x_i is mapped into a higher dimensional space by the functionϕ. SVM finds a linear separating hyperplane with the maximal margin in this higher dimensional space. $C > 0$ is the penalty parameter of the error term. Furthermore, $K(x_i, x_j) = (x_i)^T \phi(x_j)$ is called the kernel function. There are four basic kernels:

1. Linear Kernel : $K(x, x_j) = x_i^T x_j$
2. Polynomial Kernel : $K(x_i, x_j) = (\gamma x_i^T x_j + r)^d$, $\gamma > 0$.
3. Radial Basis Function (RBF) : $K(x_i, x_j) = \exp(-\gamma \|x_i - x_j\|^2)$, $\gamma > 0$.
4. Sigmoid Kernel : $K(x_i , x_j) = \tanh(x_i^T x_j + r)$.

Here,γ, r and d are kernel parameters.

- **Multiclass Classification using SVM**

Multiclass SVM aims to assign labels to instances by using support vector machines, where the labels are drawn from a finite set of several elements. The dominant approach for doing so is to reduce the single multiclass problem into multiple binary classification problems (Duan and Keerthi, 2005). SVMs classifiers address a multi-class problem as either

(i) One of the labels to the rest (one-versus-all) or
(ii) Between every pair of classes (one-versus-one).

Classification of new instances for one-versus-all case is done by a winner-takes-all strategy, in which the classifier with the highest output function assigns the class (it is important that the output functions be calibrated to produce comparable scores).

For the one-versus-one approach, classification is done by a max-wins voting strategy, in which every classifier assigns the instance to one of the two classes, then the vote for the assigned class is increased by one vote, and finally

the class with most votes determines the instance classification. The present research work uses a "one-against-one" approach (Knerr et al., 1990) for multiclass classification. Some early works of applying this strategy to SVM include, for example, Kressel (1998). Hsu and Lin (2002a) performed a comparison and concluded that "one-against-one" is a competitive approach for multiclass problem.

In multi-class SVM classification, if k is the number of classes, then k (k – 1) = 2 classifiers are constructed and each one trains data from two classes. For training data from the i^{th} and the j^{th} classes, the two-class classification problem is solved as below.

$$\min_{\omega^{ij}, b^{ij}, \xi^{ij}} \quad \frac{1}{2}(\omega^{ij})^T \omega^{ij} + C \sum (\xi^{ij})_t$$

$$\begin{aligned} \text{Subject to} \quad & (\omega^{ij})^T \varphi(x_t) + b^{ij} \geq 1 - \xi_t^{ij}, \text{ if } x_t \text{ in the ith class,} \\ & (\omega^{ij})^T \varphi(x_t) + b^{ij} \geq 1 - \xi_t^{ij}, \text{ if } x_t \text{ in the ith class,} \\ & \xi_t^{ij} \geq 0. \end{aligned} \tag{3.2}$$

During classification, a voting strategy is used. Here, each binary classification is considered to be a voting where votes can be cast for all data points x. A point is assigned to a class that has maximum number of votes. When two classes have identical votes, the class that appears first in the array of storing class names is chosen.

3.1.3. K-Nearest Neighbor Classification

Among the various methods of supervised statistical pattern recognition, the Nearest Neighbor rule achieves consistently high performance, without a priori assumptions about the distributions from which the training examples are drawn. It involves a training set of both positive and negative cases. A new sample is

classified by calculating the distance to the nearest training case; the sign of that point then determines the classification of the sample. The k-NN classifier extends this idea by taking the k nearest points and assigning the sign of the majority. It is common to select k small and odd to break ties (typically 1, 3 or 5). Larger k values help reduce the effects of noisy points within the training data set, and the choice of k is often performed through cross-validation. It is a non-parametric classification model, where the training dataset is used to classify each member of a "target" dataset. The structure of the data is that there is a classification (categorical) variable of interest ("buyer," or "non-buyer," for example), and a number of additional predictor variables (age, income, location...). The algorithm is given below.

1. For each row (case) in the target dataset (the set to be classified), locate the k closest members (the K Nearest Neighbors) of the training dataset. A Euclidean Distance measure is used to calculate how close each member of the training set is to the target row that is being examined.

2. Examine the K Nearest Neighbors to find the class that is very near to the category and assign this category to the row being examined.

3. Repeat this procedure for the remaining rows (cases) in the target set.

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. In experiment, a value of 3 was set to 'k' (k=3).

- **Distance Measure Used**

In the domain of image classification from large databases using image, each 'n' dimensional feature vector may be considered as a point in the 'n' dimensional vector space. Thus, a feature vector F = {f1, f2, …, fn} is mapped to a

point P(f1, f2, ..., fn) in the n-dimensions. This mapping helps to perceive the images (represented by their feature vectors) as high-dimensional points. The advantage of this representation is that different distance metrics can be used to

(i) Finding similarity between two images and

(ii) Ordering a set of images based on their distances from a given image.

These measures can be used to perform a nearest neighbor search on a large database of images and retrieve a result set containing images having similar features. It is evident that the images and their ordering depend both on the feature extraction method as well as on the distance metric used. In this work, Euclidean distance metric is used.

Euclidean distance or Euclidean metric is the distance between two points $u(x_1, y_1)$ and $v(x_1, y_1)$ and is calculated using Equation 3.3.

$$EU(u,v) = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} \tag{3.3}$$

Instead of two dimensions, if the points have n- dimensions, such as a = (x1, x2, xn) and b = (y1, y2, ... yn) then the above equation is generalized by defining the Euclidean distance between a and b as

$$EU(a,b) = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2 + ... + (x_n - y_n)^2} = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2} \tag{3.4}$$

3.2. IMAGE FEATURES

The following six measures of the TIHC images are used to generate the feature vector in the present study.

1. Area - Area of an image in square pixels. It is calculated by multiplying number of rows and number of columns of the image.

2. Mean – It is calculated as the sum of the gray values of all the pixels in the image divided by the number of pixels.
3. Standard deviation – Refers to the standard deviation of the gray values used to generate the mean gray value.
4. Minimum intensity - Minimum intensity value of the image.
5. Maximum intensity- Maximum intensity value of the image
6. Median - The median value of the pixels in the image

The feature vector has seven columns and 'n' rows, where n is the number of images in the dataset. The data structure used to store the feature vector is a 2-dimensional matrix array as given in Figure 3.1.

Feature 1	**Feature 2**	**Feature 3**	**Feature 4**	**Feature 5**	**Feature 6**	**Image Label**
Real Value	Real Value	Real Value	Real Value	Real Value	Real Value	Integer Value

Figure 3.1: Feature Vector Data Structure

The last column of the feature vector contains the integer code that acts as Target (Label) of each image. The integer code is an unique number assigned to each character of TIHC. An example is shown in Figure 3.2. The letter  is assigned the target label as 0 is assigned a label 1 and so on.

அ	ஆ	இ	ஈ	உ	ஊ	எ	ஏ	ஐ
0	1	2	3	4	5	6	7	8
ஒ	ஓ	ஃ	க	ங	ச	ஞ	ட	ண
9	10	11	12	13	14	15	16	17
த	ந	ப	ம	ய	ர	ல	வ	ழ
18	19	20	21	22	23	24	25	26
ள	ற	ன	ஸ	ஷ	ஜ	ஹ	டி	டீ
27	28	29	30	31	32	33	34	35
கு	ங	சு	ஞ	டு	ண	து	நு	பு
36	37	38	39	40	41	42	43	44
மு	யு	ரு	வு	வு	ழு	ளு	று	னு
45	46	47	48	49	50	51	52	53
கூ	நூ	சூ	ஞா	டூ	ணா	தா	றா	பூ
54	55	56	57	58	59	60	61	62
மூ	யூ	ரூ	வா	வூ	ழூ	ளூ	றா	னா
63	64	65	66	67	68	69	70	71
◌	ா	ி	ீ	ு	ூ	ெ	ே	ை
72	73	74	75	76	77	78	79	80
ொ	ோ	◌்						
81	82	83						

Figure 3.2: Class Label Code for each Tamil Character

3.3. TECHNIQUES USED FOR CLASSIFIER COMBIINATION

Having decided on the number of classifiers and type of classifiers to combine, the next step is to decide the specific methods that has to be used during fusion classification. This section explains these methods.

3.3.1. Partitioning (evaluation) method - 'Holdout' Method

Given a data set Z of size N x n, containing n-dimensional feature vectors describing N images, it is desirable to use as much as possible of the data to build the classifier (training) and also as much as possible unseen data to test its performance (testing). However, using the same data for training and testing, results in "over-training" of the classifier. In such a situation, the classifier

perfectly learns the available data, but fails with unseen data. Thus, it becomes important to have a separate data set to train and test a classifier and make the best use of Z. Several methods exist, like, Re-substitution (R-Method), Hold-Out method (H-Method), Bootstrap method and Cross-validation method.

The proposed fusion classifier uses the hold-out method for splitting the dataset into training and testing samples. The hold-out method procedure is given in Figure 3.3.

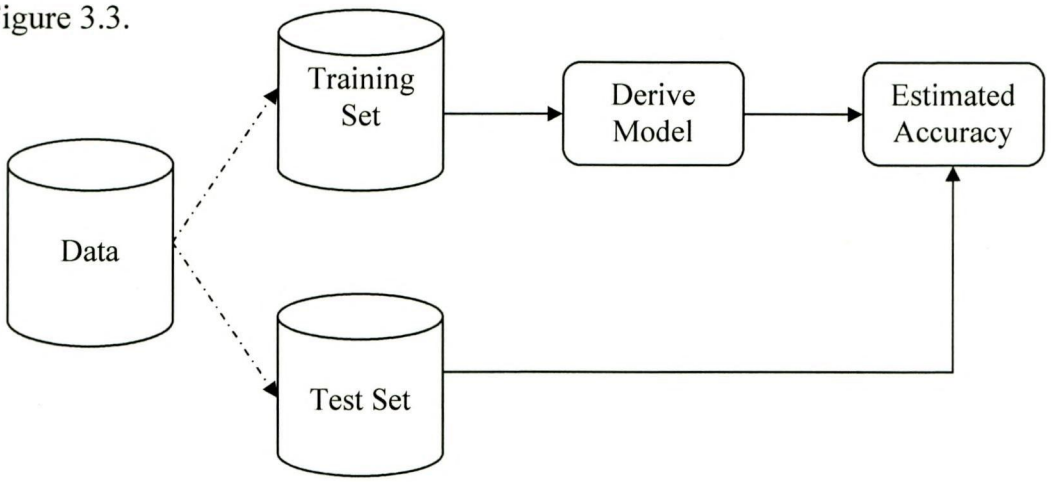

Figure 3.3: Hold-out Method

The holdout method randomly partitions the dataset into two independent sets, training and testing. Generally, two-thirds of the data are allocated to be the training set and remaining one-third is allocated as test set (Figure 3.4). The method is pessimistic because only a portion of the initial data is used to derive the model.

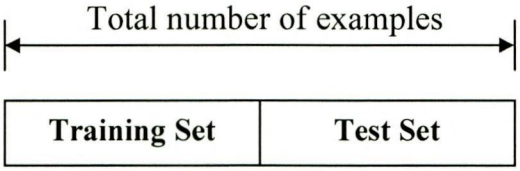

Figure 3.4: Dataset Split into training and testing set

3.3.2. Aggregation Method

While using multiple classifiers, a method that combines the results of the various classifiers is needed. Several techniques exists, namely, majority voting, maximum, sum, min, average, product, Bayes, decision template and behavior knowledge space. This research work uses the majority voting scheme to combine the outputs of classifiers. Majority vote scheme is one of the oldest strategies for decision making. Its roots are traced back to the era of ancient Greek city states and the Roman Senate. This technique is chosen because of its simplicity and speed. The method is explained below.

Let the decision of the i^{th} classifier be defined as $d_{t,j} \in \{0, 1\}$, $t = 1, \ldots, T$ and $j = 1, \ldots, C$, where T is the number of classifiers and C is the number of classes. If the i^{th} classifier chooses class ω_j, then $d_{t,j} = 1$ and 0, otherwise. In majority voting scheme, a class ω_j is chosen, if

$$\sum_{t=1}^{T} d_{t,J} = \max_{j=1}^{c} \sum_{t=1}^{T} d_{t,j} \tag{3.5}$$

The majority voting is an optimal combination rule under the minor assumptions of

- An odd number of classifiers for a two class problem
- The probability of each classifier choosing the correct class is p for any instance x; and
- The classifier outputs are independent.

Then, with majority voting, the fusion classifier makes the correct decision if at least $\lfloor T/2 \rfloor + 1$ classifiers choose the correct label, where the floor functions $\lfloor . \rfloor$ returns the largest integer less than or equal to its argument. The accuracy of the fusion classifier can be represented by the binomial distribution as

the total probability of choosing $k \geq \lfloor T/2 \rfloor + 1$ successful ones out T of classifiers, where each classifier has the success rate of p. Hence, P_{ens}, the probability of fusion classification success is

$$P_{ens} = \sum_{k=(T/1)+1}^{T} \binom{T}{k} p^k (1-p)^{T-k} \qquad (3.6)$$

3.3.3. Types of training

There are various methods used while training a multiple classifier system. They are,

a) Training of the individual classifiers and applying aggregation that does not require further training (e.g., aggregation techniques like average, minimum, product, maximum, etc.)

b) Training of the individual classifiers followed by training the aggregation

c) Simultaneous training of the whole scheme.

The present scheme uses the first method where after training the individual classifier, further classification is not required. This method is selected because the fusion classification depends on the result of the individual classifier.

3.4. PROPOSED CLASSIFIER COMBINATION METHOD

The process of image classification allows users to find desired information faster by searching only the relevant categories and not the whole information space. Image classification normally involves the processing of two main tasks.

- Feature extraction task – extract image features and forms feature vectors
- Classification task – uses the extracted features to discriminate the classes

The majority voting method using visual features is used to classify TIHC. The classifiers based on Neural Networks (NN), K Nearest Neighbor (KNN) and Support Vector Machine(SVM) that can efficiently classify images.

The proposed multiple classification method is shown in Figure 3.5. The present study proposes three two-classification fusion models and one three-classification fusion models as given below.

1. Neural Network + KNN + SVM - three class fusion model
2. Neural Network + KNN
3. KNN + SVM
4. Neural Network + SVM

Thus, the present study proposes four fusion models for the classification problem of Tamil Isolated Handwritten Character (TIHC) images. All the proposed models work in a three-step procedure.

- Train the classifiers with the training feature vector
- Use the selected classifiers to classify the test features vector to an output label
- Perform aggregation to combine the results and make the final decision.

Six features of the images were extracted to create the feature vector that is used as input during classification. The selected features are area, mean, standard

deviation, median, maximum intensity and minimum intensity. During training the last column of the feature vector contained the target label of each image.

The training and testing dataset was partitioned using the hold off method, which divided the dataset into 60% and 40%. The aggregation method used to combine the results of the various classifiers is a majority voting scheme method. Each classifier was trained separately and no training was provided on the fused model.

Classifier combination is the best combination of a set of classifiers depends on the application and on the classifiers to be combined according to the accuracy of a multiple classifier system and the individual constituent classifiers. One approach is to generate a large number of classifiers and then to select the best combinations to use in particular regions of the input space. Classifier combination methods have proved to be an effective tool to increase the performance of pattern recognition applications.

To improve the accuracy of the character classification system, the present research work proposes the use of multiple classifier or fusion of classifiers. Fusion-based classification is a technique that has been proven to be efficient than single classification algorithms. Classifier Combination has several advantages over single classifiers. It improves the accuracy of classification and reduces the failure to recognize rates.

A performance analysis to analyze the effectiveness of the proposed fusion classifiers in terms of accuracy and speed of classification and compare them against their single classifier counterparts. The Classifier Combination gives the best result for the KNN+SVM classifier.

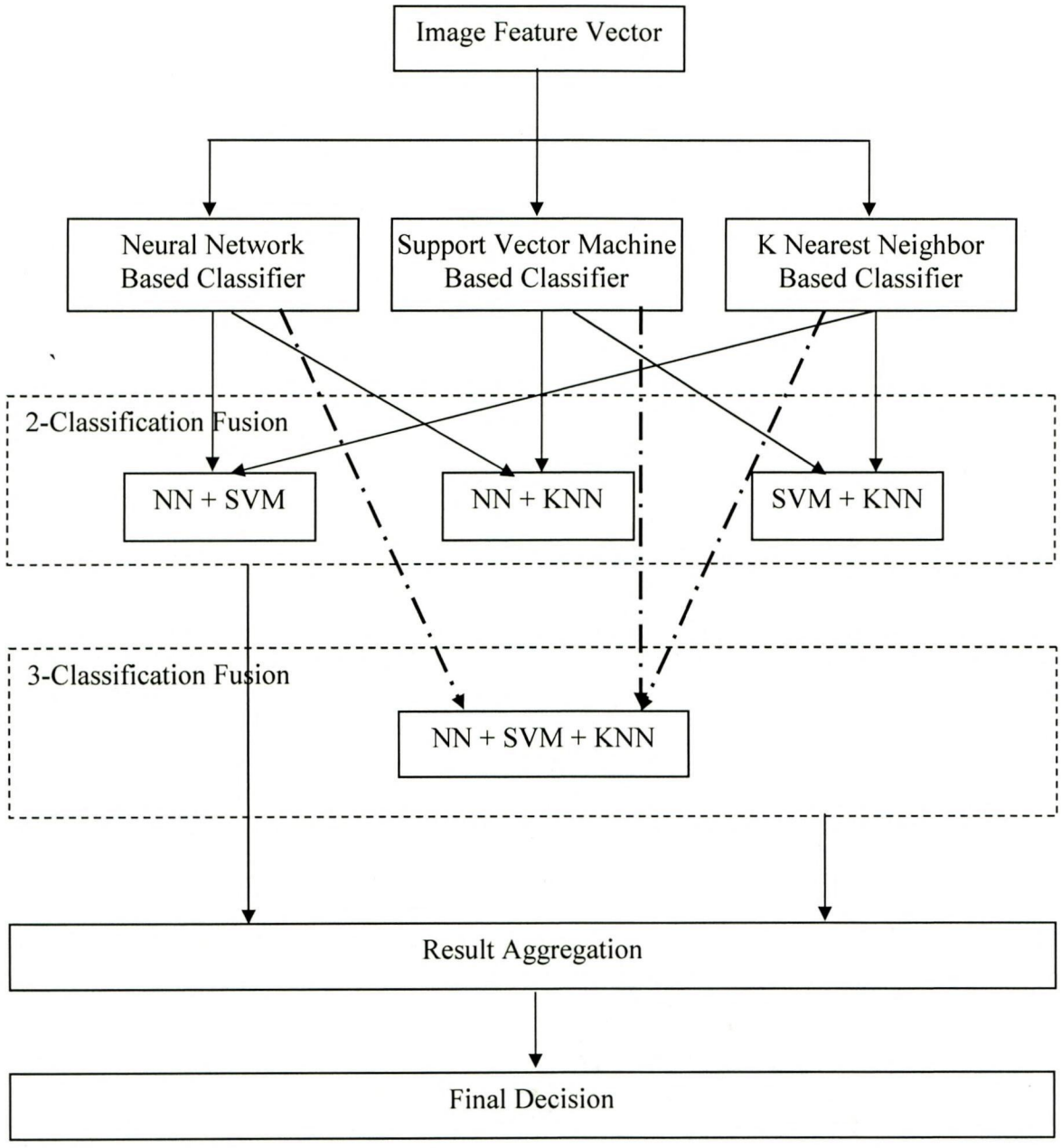

Figure 3.5: Proposed Methodology

3.5. CONCLUSION

This chapter presented the various methods and techniques used by the proposed classifier combination. Four systems are proposed that combines Neural Networks, Support Vector Machine and K Nearest Neighbor classifiers. Several experiments were conducted with the feature vectors obtained from TIHC images and the results are tabulated and discussed in the next chapter, Results and Discussion.