# Results and Discussion

# 4. RESULTS AND DISCUSSION

Image classification is an area of research which has proved to be challenging for the past several decades and it has gained more attention in Tamil character recognition due to the new challenges posed by the Tamil Scripts. In this research work, fusion-based classifiers are constructed, which takes different input features from the images. These features are used to train and test the classifiers. The various features extracted are statistical features, namely, area, maximum and minimum intensity, mean, standard deviation and median. This research work proposes four fusion classifiers which use these features for the classification task. The performances of all these classification models were tested vigorously and the results obtained are presented in this chapter.

## 4.1. DATASET USED

The TIHC dataset contains approximately 350 isolated samples each of Tamil "characters" written by native Tamil writers including school children, university graduates, and adults from the cities of Bangalore, Karnataka, India and Salem, Tamil Nadu, India. Character recognition is the important area in image processing and pattern recognition fields. Image features namely, area, mean, standard deviation, median, minimum intensity and maximum intensity. The TIHC dataset consists of 350 images, out of which 210 (60%) were taken as training data and 140 (40%) images were taken as testing data.

## 4.2. PERFORMANCE METRICS

The performance of classification algorithms greatly depends on the characteristics of the data to be classified. There is no single classifier that works best on all given problems. The purpose of this study is to develop the fusion based classification method for classifying THIC images with maximum accuracy. The performance of different classification techniques including Back Propagation

Neural Network, K-Nearest Neighbor and Support Vector Machine along with their 2-classifier and 3-classifier fused counterparts are evaluated using several performance parameters as listed below.

(i) Error Rate
(ii) Accuracy
(iii) Speed of classification

The error rate is calculated as

$$\text{Error Rate} = \frac{\text{No. of incorrectly predicted data}}{\text{Total Number of training data}} \text{x}100 \qquad (4.1)$$

The accuracy of the classifiers is calculated as

$$\text{Accuracy} = 1 - \text{Error Rate} \qquad (4.2)$$

An effective classifier should reduce the error rate while increasing the accuracy.

Training and testing time refers to the time taken by the algorithm to train and test the proposed classifier. The sum of training and testing is used to analyze the speed of the proposed classifiers.

## 4.3. EXPERIMENTAL RESULTS

All the experiments were conducted on a Pentium IV machine with 2 GB RAM. The classifier combination was developed using MATLAB 2009a. The TIHC dataset consists of 350 images, out of which 210 (60%) were taken as training data and 140 (40%) images were taken as testing data. As a preprocessing step, all the image features were calculated prior to classification and was converted to a feature vector which was given as input to the classifiers. N-fold cross-validation method is used, where N is varied from 1 to 5 in steps of 1. When N > 1, the average value is calculation and is taken as the final performance result.

Further, a standard three-layered back-propagation network with the tangent-sigmoid transfer function is considered. The weights and biases of the Neural Networks are initialized randomly, and the number of neurons in the hidden node is determined heuristically as inputs + outputs. A small value of the learning rate (0.15) and a large value of the momentum rate (0.8) are chosen to avoid local minima. The number of training epochs was 500. To implement the principles of SVM, we used the LIB-SVM. The two most important steps in implementation of SVM is scaling and kernel selection; for scaling, the values of all features were linearly scaled to the range [1, +1] to prevent the cases that features great numeric ranges dominating those in smaller numeric ranges. Among many available kernel functions linear kernel was used. The experimental results of ensemble classification with different base classifiers are presented in the following sections.

### 4.3.1. Error Rate

Table 4.1 shows the error rates obtained by the proposed 2-classifier combination and 3-classifier combination while varying the number of runs (replications) to obtain an average value. The results of the proposed classification systems are compared with their single classifier base algorithms.

### TABLE 4.1

### COMPARISON OF CLASSIFIERS ERROR RATES

| Classifier | Number of runs | | | | |
|---|---|---|---|---|---|
| | 1 | 2 | 3 | 4 | 5 |
| NN | 0.1744 | 0.1628 | 0.1396 | 0.1302 | 0.1163 |
| KNN | 0.1221 | 0.1112 | 0.1047 | 0.1040 | 0.0930 |
| SVM | 0.1628 | 0.1437 | 0.1312 | 0.1186 | 0.1024 |
| KNN+SVM | 0.0140 | 0.0123 | 0.0116 | 0.0086 | 0.0029 |
| KNN+NN | 0.0395 | 0.0320 | 0.0194 | 0.0174 | 0.0116 |
| NN + SVM | 0.1395 | 0.1337 | 0.1233 | 0.1142 | 0.1003 |
| NN+KNN+SVM | 0.0843 | 0.0736 | 0.0718 | 0.0698 | 0.0465 |

The above data can be analyzed to estimate the following two performance behaviors.

1. Which combination of the classifier gives the best result (that is, low error rate) ?
2. What is the effect of number of runs with error rate?

To answer these two questions, the above data was converted into two graphs (Figures 4.1 and 4.2). From these Figures, it is evident that the performance of classifier that combines KNN with SVM and KNN with NN is better when compared to other single classifiers with regard to error rate. Surprisingly, the performance of single classifier that uses KNN is better to the two-classifier combination that combines NN + SVM. While comparing KNN combined with SVM and NN, the KNN + SVM combination produced better error rate than KNN + NN.

Further, it is also clear that performance of three-classifier fusion that combines, KNN, SVM and NN has improved the error rates when compared with all the single classifiers. Its performance when compared with two-classifier fusion systems is lower with KNN + SVM and KNN + NN but higher than SVM + NN. It could be inferred from the projected results that the SVM and KNN algorithms play important role in improving the error rate of the proposed two-classification fusion system.

While considering the number of runs used, it is evident that the number of runs and error rate are indirectly proportional to each other. This desired trend is observed by the lower error rate achieved by the classifiers when the number of runs was fixed to 5 which has increased when the number of runs was fixed to 1.

Thus, from the various results it is clear that the KNN + SVM combination of classification produces improved results when compared with other classifier combination and hence is considered is the clear winner with respect to error rate.

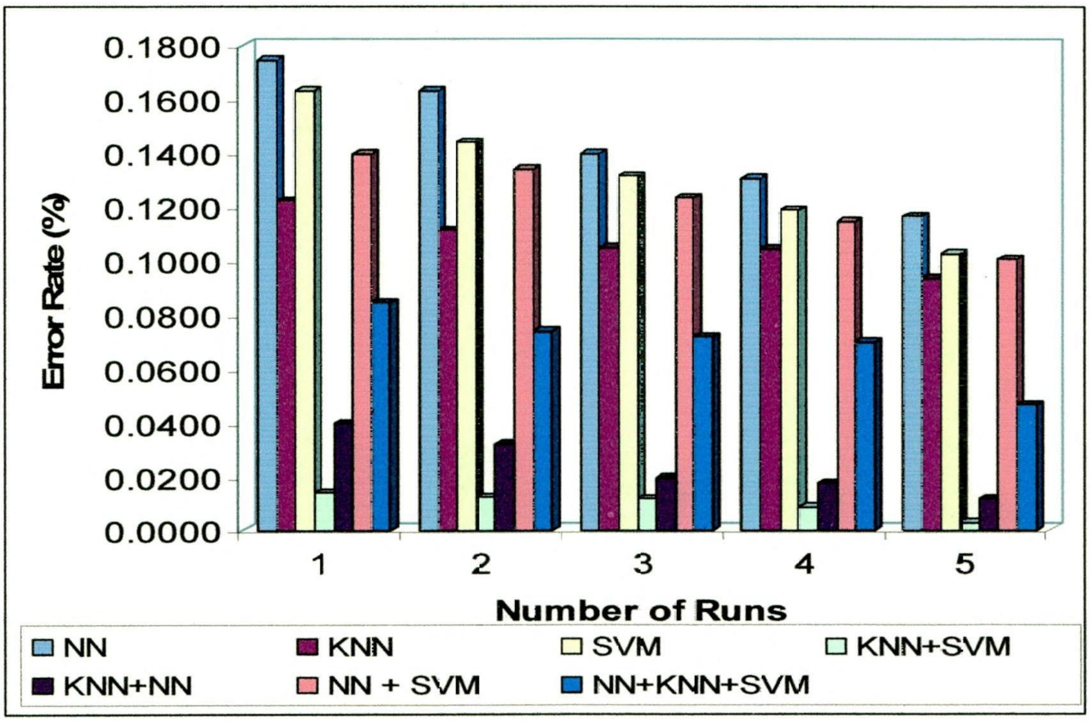



**Figure 4.1: Error Rate of Classifiers**

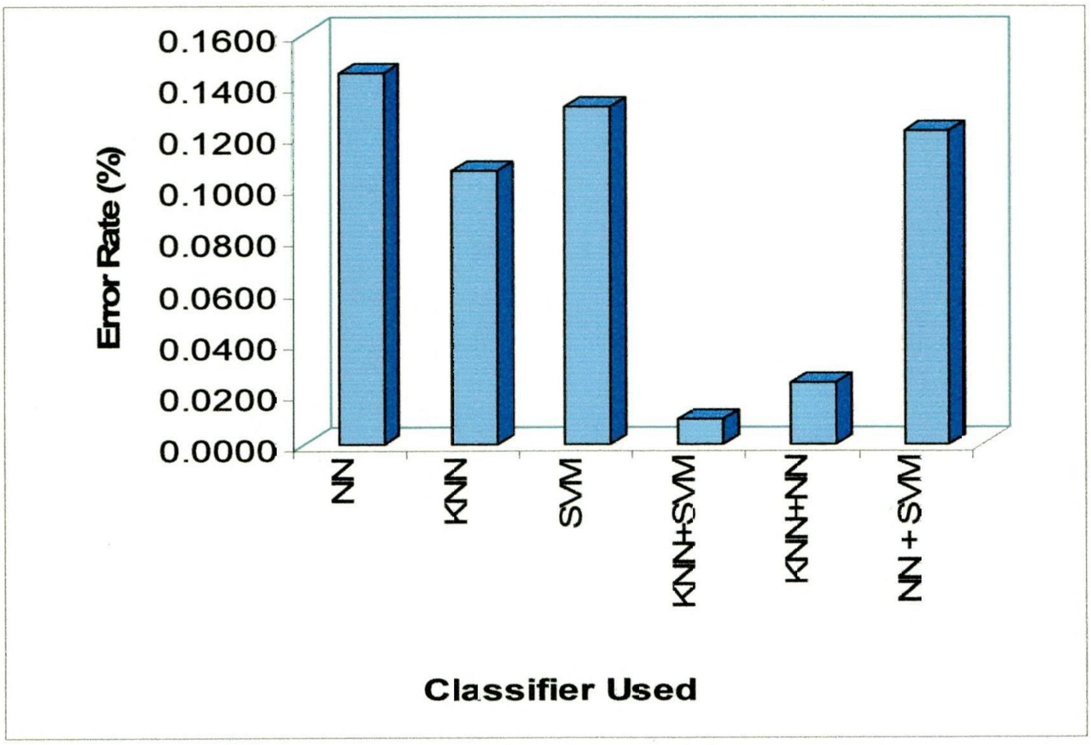



**Figure 4.2: Average Error Rate of Classifiers**

### 4.3.2. Accuracy

The next performance metric used to evaluate the proposed classification models is accuracy. Table 4.2 shows the results obtained by all the proposed fusion system while varying the number of runs (replications) to obtain an average value. Again, to analyze the efficiency of the proposed algorithms, the results of the proposed classification systems are compared with their single classifier base algorithms. Figures 4.3 and 4.4 shows the trend obtained while varying the number of runs and overall classification performance with respect to accuracy.

**TABLE 4.2**

**COMPARISON OF CLASSIFIERS ACCURACY**

| **Classifier** | **Number of Runs** | | | | |
|---|---|---|---|---|---|
| | **1** | **2** | **3** | **4** | **5** |
| NN | 82.56 | 83.72 | 86.04 | 86.98 | 88.37 |
| KNN | 87.79 | 88.88 | 89.54 | 89.61 | 90.70 |
| SVM | 83.72 | 85.63 | 86.88 | 88.14 | 89.76 |
| KNN+SVM | 98.61 | 98.77 | 98.84 | 99.14 | 99.71 |
| KNN+NN | 96.05 | 96.80 | 98.06 | 98.26 | 98.84 |
| NN + SVM | 86.05 | 86.63 | 87.67 | 88.58 | 89.97 |
| NN+KNN+SVM | 91.57 | 92.64 | 92.82 | 93.02 | 95.35 |

The results again prove that the KNN + SVM combination produces best classification accuracy, followed by KNN + NN and three-classification fusion system, NN + KNN + SVM. The accuracy of the algorithm increases with the number of runs. The single classification base systems, NN and SVM produce lower accuracy.

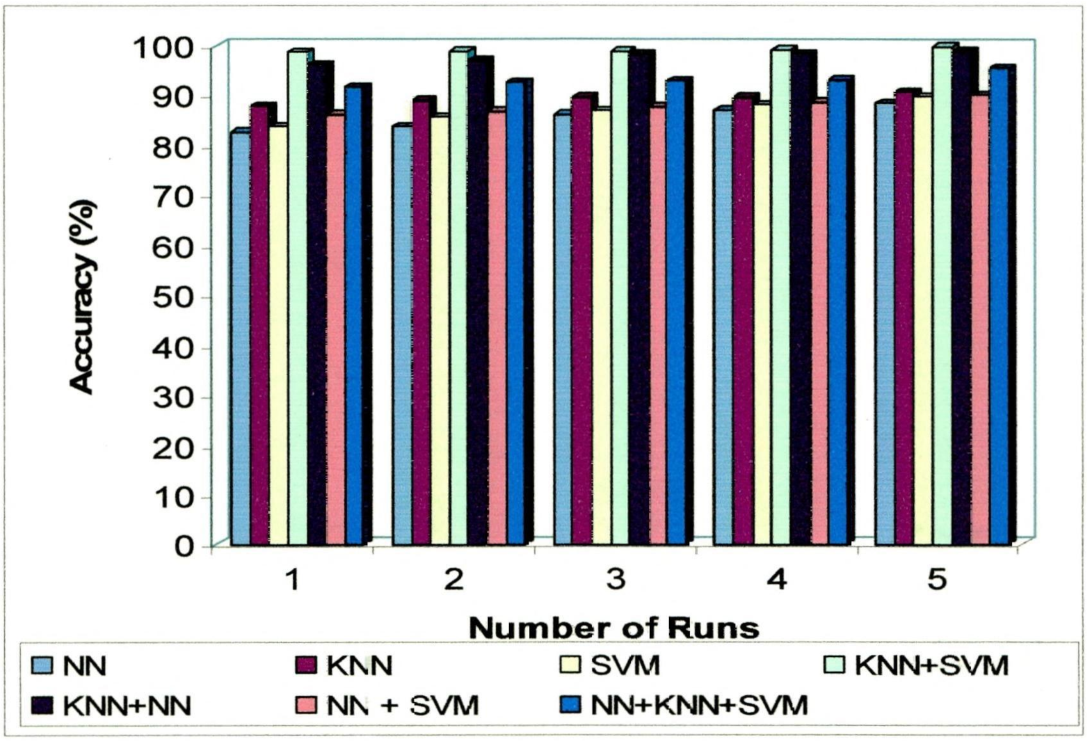



**Figure 4.3: Accuracy of Classifiers vs. Number of Runs**

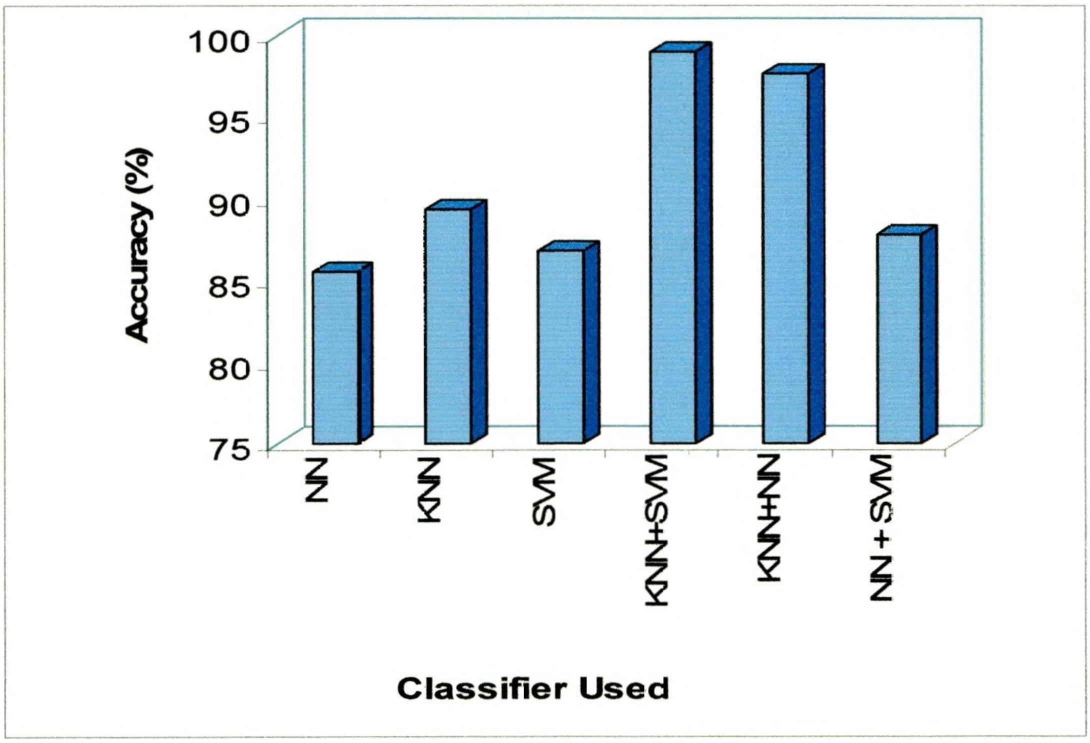



**Figure 4.4: Average Accuracy**

The KNN algorithm is again better when compared with the fusion system that combines NN + SVM classifier combination. From the results, it can be understood that the combination of KNN + SVM is better for classification in terms of accuracy.

In order to ascertain the performance efficiency obtained by the winning combination KNN + SVM over other classifiers, the gain obtained with respect to average accuracy was calculated and the results are projected in Table 4.3.

**TABLE 4.3**

**AVERAGE EFFICIENCY GAIN OF KNN + SVM CLASSIFICATION FUSION SYSTEM**

| Classifier | Average Gain (%) |
|---|---|
| NN | 13.61 |
| KNN | 9.81 |
| SVM | 12.31 |
| KNN+NN | 1.43 |
| NN + SVM | 11.34 |
| NN+KNN+SVM | 5.99 |

From the values projected in the above table, the maximum gain was achieved by NN classifier (13.61%) followed by SVM classifier (12.31%) and NN + SVM classifier (11.34), showing that these algorithms produced poor classification results.

### 4.3.3. Training and Testing Time

The classification time of a model is calculated as the sum of training and testing time. The results obtained with respect to classification time are shown in Table 4.4.

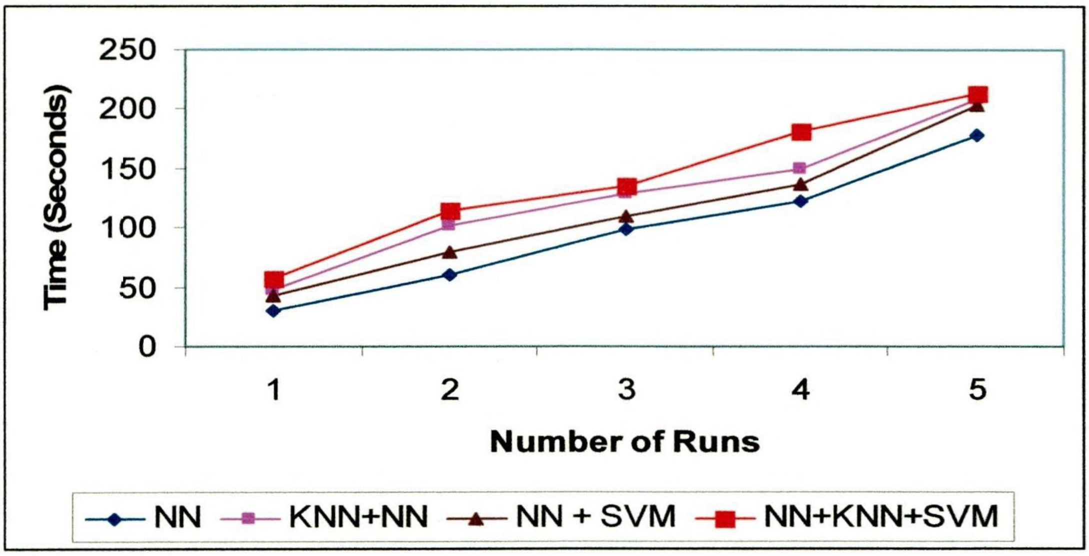
**TABLE 4.4**

**COMPARISON OF CLASSIFICATION TIME**

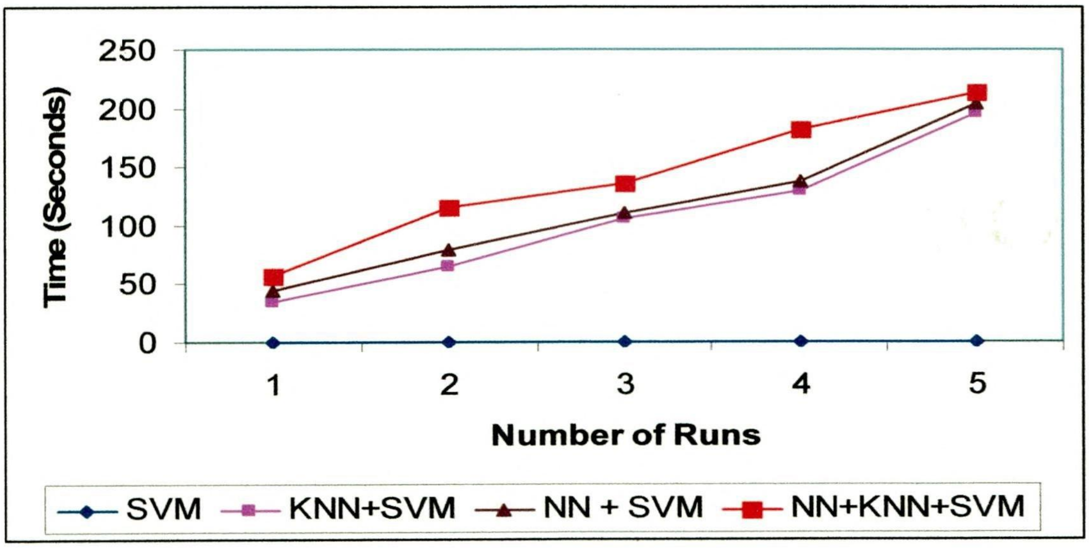
| Classifier | Number of Runs | | | | |
|---|---|---|---|---|---|
| | 1 | 2 | 3 | 4 | 5 |
| NN | 30.86 | 59.98 | 98.40 | 122.30 | 177.60 |
| KNN | 0.07 | 0.08 | 0.11 | 0.12 | 0.14 |
| SVM | 0.17 | 0.39 | 0.45 | 0.55 | 0.69 |
| KNN+SVM | 35.58 | 65.06 | 106.58 | 129.50 | 195.46 |
| KNN+NN | 47.80 | 102.24 | 128.69 | 150.39 | 209.15 |
| NN + SVM | 43.70 | 79.20 | 110.62 | 137.30 | 203.60 |
| NN+KNN+SVM | 57.40 | 114.80 | 135.90 | 181.56 | 213.80 |

From the results, as expected, the execution time of the single classifier base system is lower when compared to the classifier combination. However, careful scrutiny among two-classifier fusion revealed that the KNN+SVM classification model is the fastest followed by NN + SVM model. The three classifier combination was the slowest among all the three.

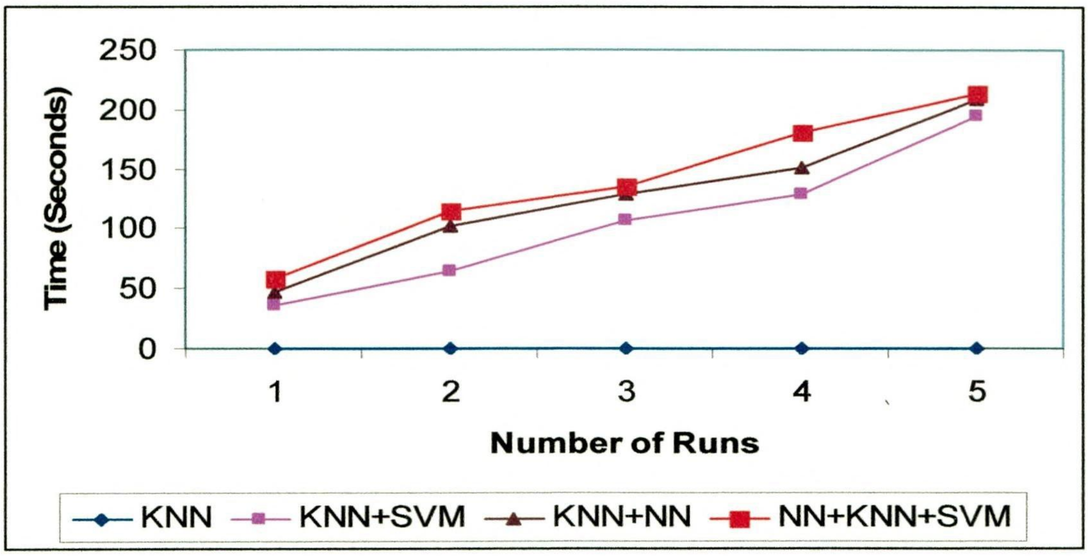
Figures 4.5a, b and c show the comparison of single classifiers with their combination counterparts.




**(a) NN with NN-Based Fusion Classifiers**




**(b) SVM with SVM-Based Fusion Classifiers**




**(c) KNN with KNN-Based Fusion Classifiers**

**Figure 4.5: Comparison of Single Base classifiers with Fusion Classifiers**

While considering Neural Networks, the combination NN + SVM was faster when compared with NN + KNN and NN + SVM + KNN. With SVM, the KNN + SVM algorithm was quicker to perform classification when compared with other fusion algorithms. The KNN + SVM combination was the fast among the KNN-based classifier combination.

Finally, it could be seen that the KNN + SVM classifier combination is the fastest to perform classification, while the three-classifier combination that combines KNN + SVM + NN is the slowest.

Further, it was also noticed that the speed of the algorithm reduces with the number of runs, which is evident from the rising trend obtained in the figures.

Thus, after comparing the result of the fusion classification models with their single classifiers, it is clear that the KNN + SVM classifier shows significant improvement both in terms of error and accuracy. Even though the classifier is slow when compared with single classifier, the accuracy gain obtained is more important.

## 4.4. CONCLUSION

The results clearly indicate that the two-classification fusion algorithm that combines KNN and SVM is better than all the other three proposed models. This combination shows high performance in terms of accuracy, error rate and speed and can reliably be used in Tamil Optical Character Recognition Systems where classification of Tamil character images is mandatory. The research work is summarized and concluded in the next section, Summary and Conclusion.