

An Efficient Method for Handwritten Kannada Digit Recognition based on PCA and SVM Classifier

Ramesh G*

Department of Computer Science & Engineering, University Visvesvaraya College of Engineering, Bengaluru, India.
rameshmg6308@gmail.com

Prasanna G B

Department of Computer Science & Engineering, University Visvesvaraya College of Engineering, Bengaluru, India.
Prasannabhagwat98@gmail.com

Santosh V Bhat

Department of Computer Science & Engineering, University Visvesvaraya College of Engineering, Bengaluru, India.
Santoshbhat1998@gmail.com

Chandrashekar Naik

Department of Computer Science & Engineering, University Visvesvaraya College of Engineering, Bengaluru, India.
Canaik24@gmail.com

Champa H.N

Department of Computer Science & Engineering, University Visvesvaraya College of Engineering, Bengaluru, India.
champahn@yahoo.co.in

Received: 16/Oct/2020

Revised: 01/May/2021

Accepted: 22/May/2021

Abstract

Handwritten digit recognition is one of the classical issues in the field of image grouping, a subfield of computer vision. The event of the handwritten digit is generous. With a wide opportunity, the issue of handwritten digit recognition by using computer vision and machine learning techniques has been a well-considered upon field. The field has gone through an exceptional turn of events, since the development of machine learning techniques. Utilizing the strategy for Support Vector Machine (SVM) and Principal Component Analysis (PCA), a robust and swift method to solve the problem of handwritten digit recognition, for the Kannada language is introduced. In this work, the Kannada-MNIST dataset is used for digit recognition to evaluate the performance of SVM and PCA. Efforts were made previously to recognize handwritten digits of different languages with this approach. However, due to the lack of a standard MNIST dataset for Kannada numerals, Kannada Handwritten digit recognition was left behind. With the introduction of the MNIST dataset for Kannada digits, we budge towards solving the problem statement and show how applying PCA for dimensionality reduction before using the SVM classifier increases the accuracy on the RBF kernel. 60,000 images are used for training and 10,000 images for testing the model and an accuracy of 99.02% on validation data and 95.44% on test data is achieved. Performance measures like Precision, Recall, and F1-score have been evaluated on the method used.

Keywords: Computer Vision; Dimensionality Reduction; Handwritten Digit Recognition; Kannada-MNIST Dataset; PCA; SVM.

1- Introduction

Machine learning and deep learning assumes a significant function in computer technology and artificial intelligence. With the utilization of machine learning, human exertion can be diminished in recognizing, learning, predicting and lot more regions. It is a fast-growing field of computer science that is making its way into all other domains. A significant space in this field is efficient and generic handwritten digit recognition. The handwritten digit recognition has many potential real applications such as marks digitization, banking utilities, reading postal code and tax form. The isolated handwriting recognition process

can be broken down into three stages: pre- processing, feature extraction and classification. The important role in Feature extraction in getting high accuracy rates. However, along with this proper pre-processing of data also contributes to high accuracy. Many research activities are made in this regard for English numerals and impressive outputs are obtained. However, there is room for more improvement when it comes to Kannada numerals.

Computer vision is the field of computer engineering that centers around duplicating portions of the unpredictability of the human vision framework and empowering computer to recognize and deal with objects in pictures similarly that people do. On a specific level Computer Vision is about example pattern recognition. So one approach to prepare a computer how to understand

* Corresponding Author

visual information is to include the image, bunches of images on the off chance that conceivable that have been named, and afterwards subject those to different programming strategies, or calculations, that permit the computer to chase down examples in all the components that identify with those labels. Machine learning and computer vision are two fields that have gotten firmly identified with each other. Machine learning has improved computer vision about recognition and following. It offers successful strategies for acquisition, image processing, and object focus are utilized in computer vision. In turn, computer vision has broadened the scope of machine learning. PC vision has expanded the extent of AI. Machine learning is utilized in computer vision in the translation phase of digital image recognition.

A pictorial representation for the steps involved in image classification and recognition can be seen in fig. 1 Image Acquisition is the first step and is basically capturing an image and generally involves pre-processing of image, such as scaling, de-skewing etc. Following the acquisition step is image enhancement where the noise in the image are removed and also necessary enhancements such as increasing/decreasing contrast etc. is done in order to improve the image quality. Image Restoration forms the very next step and mainly targets improving appearance of an image by considering the image blur and hence reducing it using mathematical or probabilistic models. Compression as the name suggests involves reducing the size of the image using few well known techniques without losing much of the image quality. Segmentation is one of the important phases which involves separating the data into distinct groups. The member of the segment is different to other and they are similar from of the well-defined segment to other Representation and Description involves representing the data in various forms in an N-Dimensional space. Description or Feature Selection helps in extracting useful information. Recognition is the final process where labels are attached to the images based on the feature matching and classification

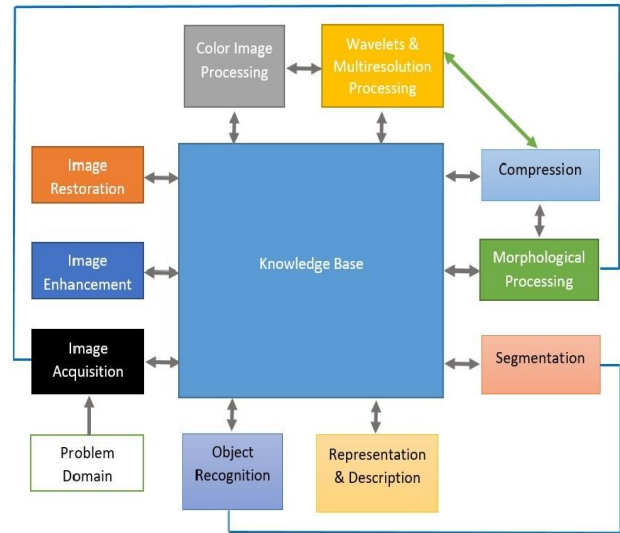


Fig. 1. Steps involved in image classification and recognition [23].

Support Vector Machines (SVM) is widely used for classification of numbers in handwritten digit recognition due to high accuracy. But when Principal Component Analysis (PCA) is used as a pre-processing step along with SVM much higher accuracy rate is seen. PCA reduces the number of features, and then they use some Principal Components (the eigenvectors of the covariance matrix) as the new features. This in turn removes the non-predictive features and gives much better results. This work presents recognizing the manually written Kannada digits (0 to 9) from the K-MNIST dataset, looking at SVM classifier and the cascade of SVM-PCA on RBF kernel. Various performance factors like the accuracy, precision, re- call, F1 score have been compared for the kernel and the two classifiers.

1-1- Motivation

Kannada numerals have a very long and rich history. The earliest inscription having all 9 Kannada numerals have been engraved in the Gudnapur Inscription which dates back to the time of Kadamba Ravivarma (485 A.D. to 519 A.D.). The symbols used to represent digits from 0 to 9 in the language are different from the well-known and modern Hindu-Arabic numerals. Even today, people of Karnataka use Kannada digits for day-to-day affairs. Kannada numerals also got itself a full- fledged Kannada-MNIST dataset in 2019. There have been numerous works around Kannada digits in ML before this. However, Kannada-MNIST data provides sufficient amount of data for training and testing. The state-of-the-art classifiers like SVM along with PCA have been used for recognition of handwritten digits in various languages. However, the method has not been implemented yet on a standard native

language dataset. This motivated us to use the SVM classifier along with PCA on the Kannada-MNIST dataset.

1-2- Contribution

We have used the Kannada-MNIST dataset to first train our model only using SVM. In the next phase we applied PCA before applying SVM. Based on the validation result we fine-tuned the parameters such as n-component value for PCA. After this we again trained our model using SVM. And this model was used to predict labels in the test data.

1-3- Organization

The work is unfurled over the pages in 7 segments. section I gives the introduction to the proposed work. An understanding about the past work is done on the difficult explanation is spread out in Section 2. section 3 depicts obviously the issue explanation we have drawn closer to settle. The proposed framework is described in Section 4 and Section 5 gives an itemized portrayal of the proposed framework. section 6 contains the test consequences of the proposed framework while Section 7 is an end to the work.

2- Literature Survey

The work on Kannada handwritten digit recognition is very limited and less research has been taken place in this field. There is scope for improvement in the techniques used till date for recognition of Kannada numerals. The recognition of isolated Kannada characters was first explored by Kunte [1] where wavelet features were extracted from the character contour and used as features. A character recognition accuracy of 56% was achieved using a Multi-layer feedforward neural network with one hidden layer. In Rajput et al., [2] binary images of numerals are created by scanning and a size of 40 x 40-pixel image is created after normalization. The line between the object pixels and the background (the crack) is computed and these are termed as Crack Codes. These codes are then represented in complex plane and features computed from 10 dimensional Fourier descriptors are used. The experiment is carried out using five-fold cross validation method with SVM as classifier. The work states that an accuracy of 99.76% and 95.22% has been obtained for printed and handwritten numerals, respectively. Segmenting and recognizing arbitrarily connected and superimposed handwritten numeral recognition in one-stroke finger gestures has been a problem and a solution to this has been proposed by Chiang et al., [3]. The method has two phases, key numeral spotting (KNS) phase and recognition by concatenation (RBC) phase. For recognizing key numerals in gesture, dynamic time warping (DTW) algorithm which is an endpoint detection

method is used. The proposed solution achieves 94% precision.

Ramappa et al., [4] presented by the continuous exploration in Optical Character Recognition Systems with center around various techniques for division, segmentation, feature extraction and for classification. They are considering eight different features computed from zonal extraction, radon transform, fan beam projections, image fusion, discrete fourier transform, run length count, directional chain code and curvelet transform along with 10 different classifiers like Euclidean distance, Chebyshev distance, Manhattan distance, K-NN, K-medoids, Linear classifier, Cosine distance, Artificial Immune system, K-means and Classifier fusion are considered. They conclude by observing that a maximum recognition rate of 98.5% is achieved by zoning features with Artificial Immune System and that it outperforms all the other combinations.

A Zone based features is employed for recognition of handwritten and printed digits by Dhandra et al., [5]. 64 zones are formed from a digit image each zone is compared for pixel density. This procedure is sequentially repeated for entire zone. For classification and recognition 64 features are selected from the previous step. The value of zone row/column with empty foreground pixels in the feature vector is zero. By using KNN and SVM classifiers the work concludes saying that an accuracy of 97.32% and 98.30% respectively was achieved for mixed handwritten and printed Kannada digits. Karthik et al., [6] present a Histogram of Oriented Gradients (HOG) based method for the recognition of handwritten kannada numerals. HOG descriptors are considered one of the best descriptors for character recognition problem since they are invariant to geo-metric transformation. Multi-class Support Vector Machines (SVM) has been used for the classification. The proposed algorithm has achieved an average accuracy of 95% when experimented on 4,000 images of isolated handwritten Kannada numerals. Zone and Distance metric based feature extraction approach has been carried out by Rajashekararadhya [7] where character centroids are computed for each character image. The image is further divided into an equal zones and an average distance from the character centroid to each pixel present in the zone is computed. The classification and recognition is carried out using a feed forward back propagation neural network a recognition accuracy of 98% and 96% are obtained for Kannada and Telugu numerals respectively. The problem of recognizing printed and handwritten numerals seen in various documents has been addressed by Rajput et al. [8] where a Support Vector Machine based classification is implemented. Scanned numerals are converted to binary image and normalized to a size of 40 x 40. The boundaries are traced and chain codes are calculated. These codes are then represented in complex plane and features computed from 10 dimensional Fourier descriptors are used. These

are input to a multi-class SVM for recognition of class. An accuracy of 97.76% has been achieved for a data set size of 5000 mixed numerals image.

Hallur et al., [9] proposes a Holistic based approach for the recognition of offline handwritten numeral recognition. Due to varied shapes of kannada digits several features are considered carefully. Initial digits recognition varies after choosing such a set of features. Gradient features are extracted from gray scale image (value ranges from 0 to 255). Quadratic classifiers are used for classification purpose and an overall recognition accuracy of 95.98% is achieved on a dataset size of 1470 contributed by 147 people. U. Pal et al. [10] propose a quadratic classifier-based scheme for recognition of offline handwritten numerals of six popular Indian script. Devnagari, Bangla, Telugu, Oriya, Kannada and Tamil scripts are considered for the experiment. The bounding box of a numeral is segmented into blocks and the directional features are computed in each of the blocks as part of feature computation. Gaussian filter is then used to down sample these blocks and the features obtained are fed to a modified quadratic classifier for recognition.

Bovolo et al., [11] proposes a classifier that incorporated fuzzy logic for generalizing the properties of SVMs for the identification and modeling of many classes in mixed pixels. The results of this is a fuzzy-input fuzzy-output support vector machine classifier. This classifier processes fuzzy information given to as input to the classification algorithm in the learning phase of the classifier for modeling the subpixel information. It also provides a fuzzy modeling of the classification results, allowing a relation many-to-one between classes and pixels. Maloo et al., [12] propose a method for the recognition of Gujarati handwritten numerals which is basically an SVM based recognition scheme. Morphological operations are considered during the pre-processing stage of this method. Each isolated numeral is segmented into blocks in order to compute features. These blocks then create base for four sets of features. These sets of features are then used to obtain affine invariant moments as features which are fed as input to SVM classifier. The work mentions that the method has obtained a recognition rate of 91% approximately. Sajedi et al., [13] proposes a standardization of research works on OCR in Persian language. A database named PHOND, Persian Handwritten Optical Numbers & Digits is used for classification purpose. The proposed method uses K-Nearest Neighbour (KNN), and Support Vector Machines (SVM) with RBF, Linear and Polynomial kernels employed in SVM for measuring the effectiveness of the extracted features. The proposed method is said to achieve a higher recognition rate up to 99%.

A multiclass classifier-based approach is mentioned in [14] where PCA of HOG is utilized for exact and quick recognition of handwritten digits. HOG is known as a

powerful element descriptor, while PCA brings about quick multiclass recognition. By joining PCA with HOG, the PCA-of-HOG based classifier is reported to have achieved a recognition rate of 98.39% when applied on 10,000 test data from standard MNIST database. A very recent work on kannada MNIST can be inferred from [15] where the authors have used a skip architecture of CNN to skip some layers of the neural network and the output of previous layer is fed as the input to the next or some other layer. Experimental results show that they have achieved 97.53% accuracy on Kannada MNIST and 85.02% accuracy on Dig-MNIST Dataset. The Generative Adversarial Networks (GAN) is one of the technique that doesn't need earlier information on the possible variabilities that exist across guides to make novel artificial models [16]. On account of a training dataset with a low number of named models, which are portrayed in a high dimensional space, the classifier may sum up ineffectively. Hence, we target advancing information bases of images or signals for improving the classifier execution by planning a GAN for making artificial images. The proposed and evaluated generative adversarial networks as an information expansion procedure for the grouping of manually written digits of various contents (Latin, Bangla, Devanagari, and Oriya). The outcomes propose that such a methodology offers a significant increase in the accuracy.

The Deep Convolutional Self-organizing out Maps (DC-SOM) network contains a course of convolutional SOM layers prepared [17] successively to speak to various levels of features. The 2D SOM network is normally utilized for either information perception or feature extraction. This work utilizes high dimensional map size to make another deep network. A lot of experiments utilizing MNIST manually written digit information base and every one of its variations are directed to assess the robust representation of the proposed DCSOM network. Test results uncover that the presentation of DCSOM outperforms state-of-the-art techniques for noisy digits and accomplish a comparable performance with other deep learning architecture for other image varieties.

Cocotte et al., [19] presented by a novel dynamic learning strategy for the classification of manually written digits. The proposed technique depends on a k-closest neighbor graph got with an image deformation model, which considers nearby deformations. During the dynamic learning system, the user is first approached to labels the vertices with the highest number of neighbors. Label propagation function is performed to consequently label the models. The system is repeated until all the pictures are marked, they are evaluating the performance of the strategy on four database bases relating to various contents (Latin, Bangla, Devnagari, and Oriya) and they show that it is possible to label just 332 pictures in the MNIST

training database to acquire an accuracy of 98.54% on this same database base (60000 images).

Table I Summary of Literature Survey

Author Year	Title	Attribute
Kunte et al., 2000	170wavelet features based on-line recognition of handwritten	A character recognition accuracy of 56% was achieved using a Multi-layer feedforward neural network with one hidden layer.
Rajput et al., 2010	Printed and handwritten kannada numeral recognition using crack codes and fourier descriptors plate	Five-fold cross validation method with SVM as classifier is used to get an accuracy of 99.76% and 95.22% for printed and handwritten numerals, respectively.
Chiang et al., 2017	Recognizing arbitrarily connected and superimposed handwritten numerals in intangible writing interfaces	Dynamic time warping (DTW) algorithm which is an endpoint detection method is used. The proposed solution achieves 94% precision.
Ramappa et al., 2013	A comparative study of different feature extraction and classification methods for recognition of hand-written kannada numerals	Maximum recognition rate of 98.5% is achieved by zoning features with Artificial Immune System and it outperforms all the other combinations
Dhandra et al., 2011	Zone based features for handwritten and printed mixed kannada digits recognition	KNN and SVM classifiers are used to get an accuracy of 97.32% and 98.30% for mixed handwritten and printed Kannada digits
Karthik et al., 2015	Handwritten kannada	Histogram of Oriented Gradients(HOG)

	numerals recognition using histogram of oriented gradient descriptors and support vector machines	descriptors are used for character recognition and Multi-class Support Vector Machines (SVM) has been used for the classification. Accuracy of 95% achieved on 4,000 images of isolated handwritten Kannada numerals
Rajshaka rarahya et al., 2018	Neural network based handwritten numeral recognition of kannada and telugu scripts	The classification and recognition is carried out using a feed forward back propagation neural network a recognition accuracy of 98% and 96% are obtained for Kannada and Telugu numerals respectively
Rajput et al., 2010	Printed and handwritten mixed kannada numerals recognition using svm	Scanned numerals are converted to binary image and normalized to a size of 40 x 40. The boundaries are traced and chain codes are calculated. Features computed from 10 dimensional Fourier descriptors are used. These are input to a multi-class SVM for recognition of class. An accuracy of 97.76% has been achieved for a data set size of 5000 mixed numerals image
Hallur et al., 2014	Offline kannada handwritten numeral recognition: Holistic approach	Gradient features are extracted from gray scale image (value ranges from 0 to 255). Quadratic classifiers are used for classification purpose and an overall recognition accuracy of 95.98% is achieved on a dataset size of 1470 contributed by 147 people
U.Pal et al., 2007	Handwritten numeral recognition of six popular indian scripts	The bounding box of a numeral is segmented into blocks and the directional features are computed in each of the blocks as part of feature computation. Gaussian filter is then used to down sample these blocks and

		the features obtained are fed to a modified quadratic classifier for recognition
Bovolo et al., 2010	A novel technique for subpixel image classification based on support vector machine	Fuzzy logic classifier used to process fuzzy information given to as input to the classification algorithm in the learning phase of the classifier for modeling the subpixel information and provides a fuzzy modeling of the classification results.
Maloo et al., 2011	Support vector machine based gujarati numeral recognition	Morphological operations are considered during the pre-processing stage of this method. Later SVM classifier is used to get a recognition rate of 91% approximately
Sajedi et al., 2016	Handwriting recognition of digits, signs, and numerical strings in persian	PHOND, Persian Handwritten Optical Numbers & Digits is used for classification purpose. The proposed method uses K-Nearest Neighbour (KNN), and Support Vector Machines (SVM) with RBF, Linear and Polynomial to achieve a recognition rate up to 99%.
Cocotte et al., 2016	Active graph based semi-supervised learning using image matching: application to handwritten digit recognition	K-closest neighbor graph got with an image deformation model ND Label propagation function is performed to consequently label the models to acquire an accuracy of 98.54% on MNIST training database

3- Problem Statement and Objectives

This work mainly focuses on solving the recognition of handwritten Kannada digits, by using the state-of-the-art classifier, which gives very high accuracy along with reliability. This effort is focused on using SVM with PCA on the Kannada-MNIST dataset to recognize the digits accurately. The goal of this work is to solve Recognition of Handwritten Kannada Digits, which is a very

challenging topic for the researchers in recent years. This lets the computer to understand Kannada digits that is written manually by the user using PCA and SVM. This work likewise analyses the limitations in different methods that are being used as the solution for the issue.

4- System Architecture

The architecture followed in the proposed method is neatly laid out in Fig. 2 the dataset comprises of images of handwritten numerals in Kannada with 60,000 image in preparing set and 10,000 image in the test set with a size of 28x28 for each image. Pre-processing of the data is carried out in the subsequent step where data is de-skewed first to correct the alignment of the data. After this step, we scale the data for normalization. We also reduce the noise in the data by filling out the missing data or ignore the less important data. Following the pre-processing stage, feature extraction is carried out where PCA is used to map the high dimensionality space of input features to lower dimensions by throwing away some columns. It selects some columns of features, which has low standard deviation and deletes it from the feature matrix. The data points with the highest correlation will be retained after applying this method and has good impact on further training process.

One of the major parts of the system architecture include classification using SVM, which classifies the digits based on the features of the array of grayscale images. The features considered here are the Mean and standard deviation of each digit. Multiclass SVM classifiers are trained using these features which effectively separates different classes of digits by finding the hyperplanes with maximum margins. The performance of the model can be scrutinized by visualizing the output using pictorial elements such as charts, graphs, and maps. Data visualization tools provide assistance in understanding trends, outliers, deviation and patterns in data. By this, we can compare the expected and predicted output of the models. We can also diminish the performance of different algorithms by representing data in a pictorial form. Output analysis is the essential step to evaluate the performance of the model. A good model should give satisfactory results for the input data. We can use Confusion Matrix, F1 Score, classification accuracy, mean squared errors etc. to evaluate the performance of the model.

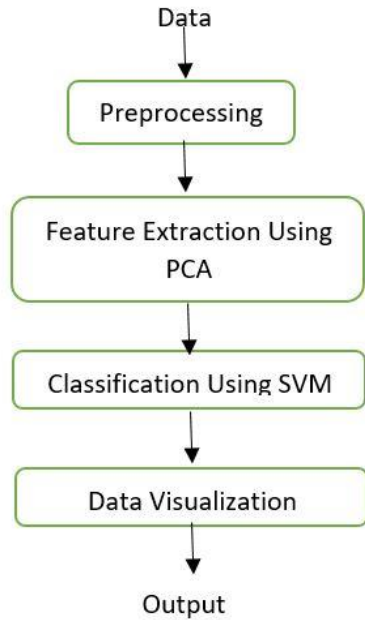


Fig. 2. System Architecture.

5- Proposed PCA-SVM Classifier

5-1- Kannada Numerals

Kannada is the official and the most spoken language in Karnataka- a state of India. It is one of the oldest Dravidian languages of India just like Tamil. The earliest inscription having all 9 Kannada numerals have been engraved in the Gud napur Inscription which dates back to the time of Kadamba Ravivarma (485 A.D. to 519 A.D.). The symbols used to represent 09 as shown in Fig. 3 are distinct from the modern Hindu-Arabic numerals.

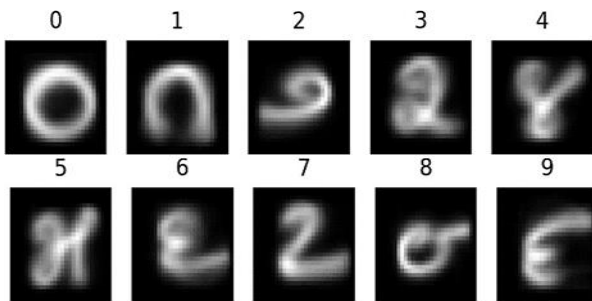


Fig. 3. Handwritten digits in the Kannada MNIST dataset.

5-2-Principal Component Analysis

PCA is a strategy, which utilizes orthogonal transformation change to convert connected information into uncorrelated information known as principal components. It produces a simple representation of a data set by eliminating the columns containing less significant features and thus reduces the dimensionality of the data as shown in Fig. 4. An arrangement of linear combination of the factors that have greatest change and are commonly uncorrelated [21]. Apart from finding, the features with major significance it also used for data visualization.

$$Z_1 = \Phi_{11}X_1 + \Phi_{21}X_2 + \dots + \Phi_{p1}X_p \tag{1}$$

In Equation 1, a set of features X_1, X_2, \dots, X_p is converted into a single principal component Z_1 . Here Z_1 is the largest variance of normalized linear combination of the features X_1, X_2, \dots, X_p . And $\Phi_{11}, \Phi_{21}, \dots, \Phi_{p1}$ are the loadings of the first principal component. Eliminating the correlated data minimizes the loss of information.

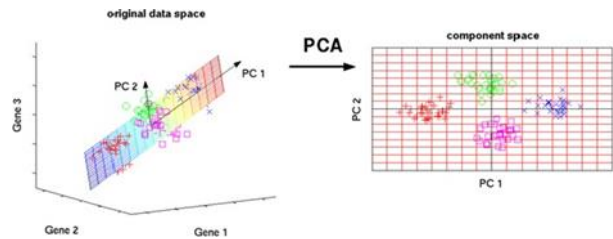


Fig. 4. Dimensionality reduction from 3D to 2D by finding the common hyperplane.

5-3-Support Vector Machine

The SVM is used as the classification algorithm and is a supervised learning model. It finds a hyperplane in an N-dimensional space (N- Number of features) and such hyper- plane contains all the unique features of the data in Fig. 5. The objective is to find the best-fit hyperplane of all the possible hyperplanes. The plane is called best fit when it has the maximum distance between data points of different classes in Fig. 6. Support vectors are the points, which are close to the hyperplane and impact on the locality and inclination of the hyperplane. We can maximize the distance of the data points, which lie in different classes using these support vectors in Fig.7.

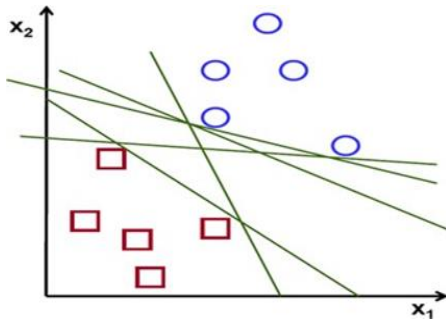


Fig. 5. Possible Hyperplane.

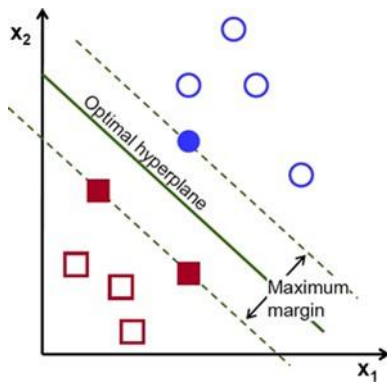


Fig. 6. Best Fit Hyperplane.

Hyperplane that separates the features of different classes is defined by the equation:

$$Y_i (W * X_i + b) \geq 1 \text{ for } 1 \leq i \leq n \tag{2}$$

where X_i are the data points shown as single dimensional matrix in a d-dimensional space, 'n' is the count of data points, Y_i -1,1 are classes of respective occurrences. 'w' and 'b' are the parameters of hyperplane. Hyperplane has to lie at a large distance as much as possible from data points of both the classes. This distance can be increased using $\frac{2}{\|w\|}$

Real world information contains noise and outliers that can be eliminated using the equation:

$$Y_i (W * X_i + b) \geq 1 - \epsilon_i, \epsilon_i \geq 0, 1 \leq i \leq n \tag{3}$$

Equation 3 is the improvised equation of 2. It has the slack variable 'ε' to separate the noise and outliers in the data. We can represent the model using different kernels like sigmoid, polynomial and RBF kernels to make it linearly separable.

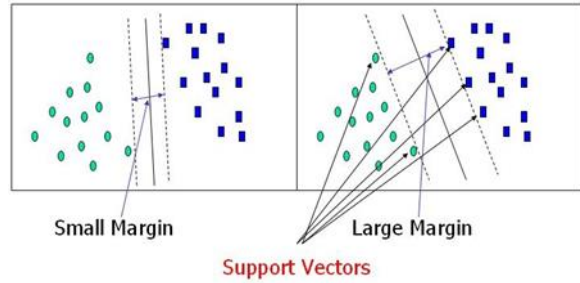


Fig. 7. Illustration of support vectors.

5-4- RBF Kernel

In the proposed method, we have used RBF kernel function as the pattern of distribution is found to be radial and it leads to impressive classification accuracy for the digits. RBF kernel function is defined as:

$$K(X_i, X_j) = \exp(-\gamma \|X_i - X_j\|) \tag{4}$$

where X_i, X_j is the Euclidean distance between X_i and X_j . γ is the parameter of kernel function. This parameter has effect on the standard or caliber of classifier, so adjusting the value of this parameter is a significant part.

5-5-Flow Diagram

Our system uses the following algorithm to solve the problem statement. The methods followed by the algorithm has been neatly laid out in Fig. 8 below.

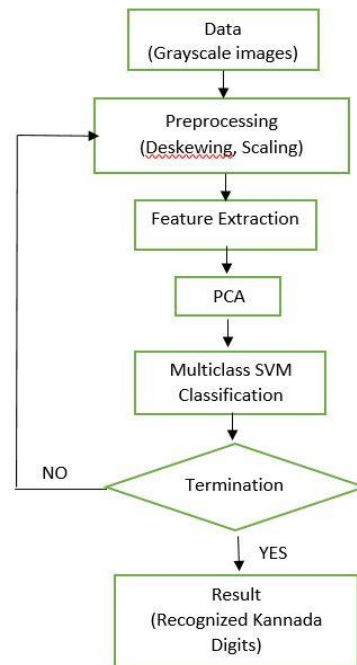


Fig. 8. Flow chart of SVM-PCA merge algorithm.

5-6- Implementation

In order to tackle the challenge in handwritten digit recognition we underwent following steps to increase the accuracy

- **De-skewing:** Different people write at different angles on paper or any surface. This leads to inclination of numbers written. Thus, the numbers appear to be skewed. Human eye can find similarities even though the images are variations of one another. But computer distinguishes such variations as different images. Hence de-skewing becomes necessary. De-skewing is the method of making a handwritten crooked image which has been scanned straight by changing the inclination of the image. To be specific, de-skewing is defined as an affine transformation. It is assumed that when the image was made initially (that is the skewed image), the image was some affine skew transformation on the de-skewed image.

$$\text{Image}^j = A(\text{Image}) + b \quad (5)$$

Where Image^j is the skew corrected version of original image Image. b gives the histogram of the gradient orientation of the original image. To find out to what extent the image has to be offset, the center of mass of the image has to be found. After this the covariance matrix of the image pixel intensities can be figured out (for this approximate the skew of the angle can be used). The following formula shows this,

$$\begin{pmatrix} 1 & 0 \\ a & 1 \end{pmatrix}$$

Where

$$a = \frac{\text{Cov}(X,Y)}{\text{Var}(X)} \quad (6)$$

- **Data Normalization:** If the values for different features vary largely from one another then the understanding capacity of the model fails significantly. It will take long time to complete the learning process and affect the model accuracy adversely. So, to achieve fast convergence we need to normalize the data and squash them into [0,1] in the preprocessing step.
- **N Component Analysis:** It is one of the necessary parameter to apply PCA on the feature space. It is the value of variance we need to keep before reducing the dimension of feature space in order to get the peak value of accuracy. The accuracy of the model has been analyzed along the Y-axis with the variance along the X-axis using the line graph shown in Fig. 9. It is found that the accuracy obtained is maximum at 0.7. Then the value of N component to our PCA algorithm has been fetched. The model kept only those features which have the variance value $\zeta=0.7$. It removed those

data points of less variance (less information) and made it easy for relational mapping of features by reducing the dimensions. PCA reduced the dimension of the feature matrix from 784 to 57 by eliminating the features having less variance than 0.7 which wouldn't contribute to the performance of the model.

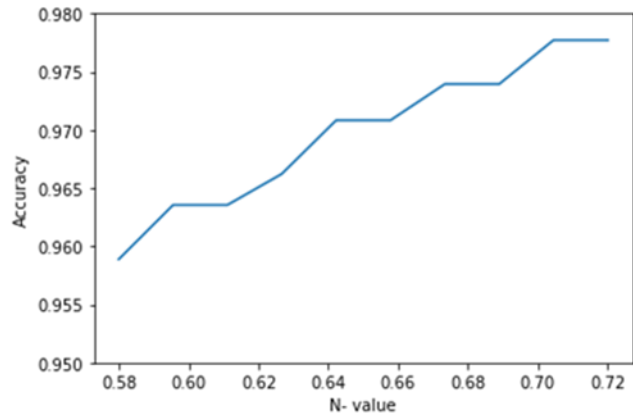


Fig. 9. Accuracy Vs Variance (N-value).

1) SVC parameters: Support Vector Classifier attempts to find the best fit hyperplane to distinguish the different classes by increasing the distance between sample points and the hyperplane. In order to demo the best fitting hyperplane fine tuning of the parameters is very important. The parameters which are tuned to get the robust model is illustrated below.

1) **Regularization Parameter (C):** It is utilized to compromise correct classification of preparing models against maximization of the decision functions margin. Small margin will be accepted for greater values of C if there is better classification of all training points correctly by the decision function. Larger margin will be encouraged by a lower C. Thus we get a simpler decision function, at the expense of training accuracy.

2) **Kernel:** To classify the nonlinear data we should apply kernel functions. These functions convert linearly independent data into linearly dependent ones. On each data instance, the kernel functions are applied which maps the initial non-linear observations into a higher-dimensional space in which they become independent. Our model reacts positively to the RBF kernel function and the value is set to rbf

3) **Gamma:** It is used to define the reach of a single training example's influence. Here far means low values and close means high values. It is clear that the gamma parameters are the inverse of the radius of influence of samples selected as support vectors by the model. Here, we set the gamma value as auto

which uses $1/n$ -features. The gamma parameter largely influences the behavior of the model. A large value of gamma will lead to the radius of the area of influence of the support vectors to include only the support vector itself. And overfitting cannot be prevented with any amount of regularization with C. On the other side, the model becomes very much constrained with a smaller value of gamma. Thus model cannot capture the shape or complexity of the data. Any selected support vector's region of influence would contain the entire training set. The model created will act in same way as a linear model which has a set of hyperplanes that differentiates high density centers of any pair of two classes.

4) Degree: It is the degree of the polynomial function. The default value of this is 3. It will be ignored by other kernels.

6- Experimental Results

The KMNIST dataset for training as well as testing the SVM+PCA model. The data is preprocessed using methods like normalization and de-skewing to increase the accuracy of the model. Then the n-component value for PCA is determined. The hyper parameters of SVM are also tuned to get the best accuracy before using the model on test data.

6-1-Dataset Analysis

The proposed methodology has been applied to a dataset [18] consisting of 70000 samples of unique handwritten Kannada numbers. For experimentation, 60000 samples are used for model training and the remaining 10000 samples are used for model testing. The size of image considered is $28 * 28$ pixel. In the preprocessing step these images undergone de-skewing and min-max normalization methods to remove the noisy and irregular data. The results of the evaluation are carried out in two-fold, first without applying the PCA and then with PCA. Of the various kernel functions available in SVM such as linear, RBF and polynomial, due to the nonlinearity of the data, RBF kernel appeared to be the best fit and with higher accuracies than the other functions.

6-2-SVM + PCA Model

The model is first trained on 42,000 digits of the dataset. Then, it is validated using 18,000 digits. After this, PCA is applied on the data to reduce the dimension of data. For this purpose, we need to choose the number of components for reducing dimension. First, a random value of 0.6 is chosen as the n-component value. This reduces the

dimension from 784 to 57. Then, SVM classifier is used on the data. The accuracy obtained was 98.8%. To improve accuracy, n-component analysis is done to get the optimal value of n-components. The accuracy change with change of N-component value can be seen in Fig. 9. The accuracy curve flattens after reaching n-components value of 0.7. This value is chosen for applying in PCA, as this is the optimal value. For n-component value of 0.7, we get accuracy of 99.02%. Based on validation results, fine tuning of the model is done and the model is then tested again on 10,000 digits. as mentioned in Table II. The difference in accuracies can be clearly seen between only SVM and SVM+PCA. Reducing the dimensionality of the data helps in increasing the accuracy in both validation data as well as test data.

Table II Analyzing Accuracy for SVM and SVM + PCA

ACCURACY	SVM	SVM + PCA
Validation data(18000 digits)	96.33%	99.02 %
Test data (1000 digits)		95.44%

6-3-Classification Report

The model is first trained only using SVM classifier without applying PCA and the accuracy obtained is 96.35%. After the application of PCA on the data, accuracy of the model increased to 99.02%. The measure of performances accuracy, precision, recall and f1 score which has been calculated for the model for the given data set. The models performance for the data set can be studied using these measures and proper tuning of parameters can be done. The accuracy obtained for individual digits is shown in Table III.

Accuracy: Accuracy is closeness of the measurements to a specific value. In classification problems, it is the proportion of number of correct predictions to the total number of input tests. Based on Confusion matrix given in Fig. 14. and the above definition we can calculate the accuracy of individual digits. This is shown in Table III and Fig. 10.

$$Accuracy = \frac{TP+TN}{FP+FN+TP+TN} \tag{7}$$

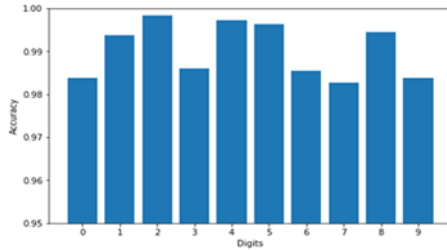


Fig. 10. Accuracy for Individual Digits.

Digit	Accuracy
0	98.38%
1	99.38%
2	99.83%
3	98.61%
4	99.72%
5	99.64%
6	98.55%
7	98.27%
8	99.44%
9	98.38%

Precision: It is the closeness of measurements to each other. In classification problems, it is the fraction of appropriate occurrences among the extracted occurrences. The precision value for individual digits calculated for the SVM + PCA model and visualized in Fig. 11.

$$\text{Precision} = \frac{TP}{FP+TP} \quad (8)$$

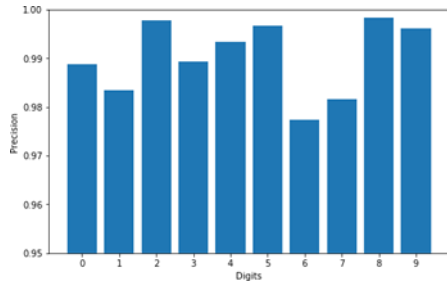


Fig. 11. Precision for Individual Digits.

Recall: It is the fraction of the total amount of appropriate occurrences that were actually extracted. It is also known as sensitivity. The recall value for individual digits calculated for the SVM + PCA model and visualized in Fig. 12.

$$\text{Recall} = \frac{TP}{FN+TP} \quad (9)$$

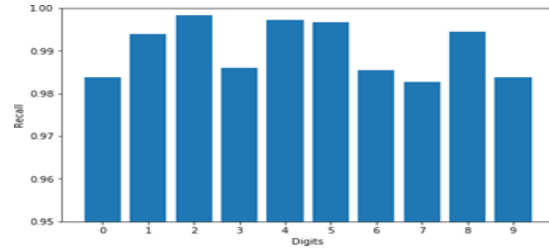


Fig. 12. Recall for Individual Digits.

F1 Score: It shows the balance between recall and precision. It is also defined as the harmonic mean of the recall and precision. The F1 Score value for individual digits calculated for the SVM + PCA model and visualized in Fig. 13.

$$\text{F1 Score} = 2 * \left(\frac{\text{Recall} * \text{Precision}}{\text{Recall} + \text{Precision}} \right) \quad (10)$$

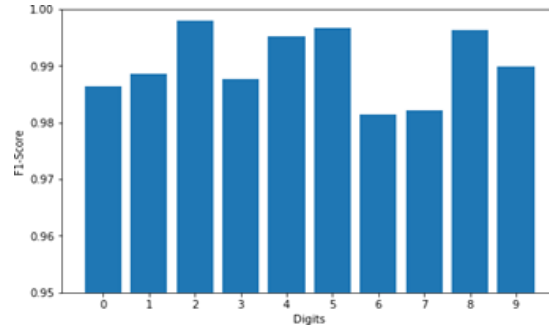


Fig. 13. F1 Score for Individual Digits.

6-4 Confusion Matrix

Confusion matrix is used in classification problems having two or more classes in the output for measurement of performance. It is also known as Error matrix. It is a particular table design which shows the representation of the presentation of an algorithm, normally a managed learning algorithm. The instances in a predicted class are represented by the rows of the matrix while instances in an actual class are represented by the columns. The confusion matrix shown in Fig. 14 clearly shows that a maximum misclassification error is observed between the digits 0 and 1. The misclassification of handwritten 0 is shown in Fig.15 The second most misclassification is seen between the digits 6 and 7. The reasons for misclassification may be due to similarity of shape found between some of the digits, but it also depends on different writing styles followed by individuals, which make the samples of a particular class closer to other class. Training accuracies for SVM along with PCA have peaked at 99.02% and the test accuracy of 95.44% as mentioned in Table IV

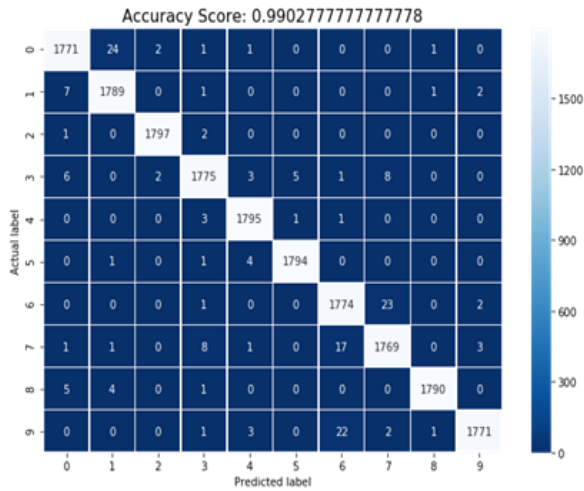


Fig. 14. Confusion Matrix for SVM+PCA model.

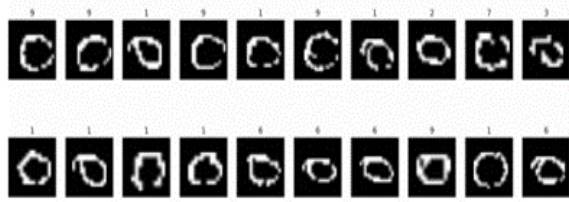


Fig. 15. Misclassification error between '0' and '1'.

Table IV Evaluation Results.

Author	Method	Accuracy
UÇAR et al.,[22] 2019	Capsule Network	81.63%
Hallur et al., [20], 2020	DCNN	96.00%
Proposed Method	PCA+ SVM	Validation - 99.02% Test data - 95.44%

7- Conclusion

In this work, we present the combined model of PCA and SVM for classification of Kannada handwritten numerals. We also present how PCA can boost the performance of SVM in terms of better classification accuracies and at improved training speed. The proposed method outperforms the trivial method of using only the Support Vector Machines without Component Analysis. Training accuracies for SVM along with PCA have peaked at 99.02% and the test accuracy of 95.44%. With these great outcomes, the real-life requirements that motivated us to take a shot at the problem statement are reachable with greater reliability. The introduced strategy's performance

measures such as Precision, Recall and F1-scores have also been summarized and studied for carefully tuning the hyper parameter. A portion of the improvements required lie in handling misclassification of digits whose physical appearance are similar. These improvements can make the framework much more strong, enabled its favorable circumstances with to deal with raw images, saving pre-processing time, to save positional data of substances, which helps in the recognition of kannada handwriting digit.

References

- [1] R. R. Kunte and R. Samuel, "170wavelet features based on-line recognition of handwritten," *Journal of the Visualization Society of Japan*, vol. 20, no. 1, pp. 417–420, 2000.
- [2] G. Rajput, H. Rajeswari, and C. Sidramappa, "Printed and handwritten kannada numeral recognition using crack codes and fourier descriptors plate," *International Journal of Computer Application (IJCA) on Recent Trends in Image Processing and Pattern Recognition (RTIPPR)*, pp. 53-58, 2010.
- [3] C. Chiang, R.-H. Wang, and B.-R. Chen, "Recognizing arbitrarily connected and superimposed handwritten numerals in intangible writing interfaces," *Pattern Recognition*, {Elsevier} vol. 61, pp. 15–28, 2017
- [4] M. H. Ramappa and S. Krishnamurthy, "A comparative study of different feature extraction and classification methods for recognition of hand-written kannada numerals," *International Journal of Database Theory & Application*, vol. 6, no. 4, pp. 71–90, 2013.
- [5] B.V.Dhandra, G. Mukarambi, and M. Hangarge, "Zone based features for handwritten and printed mixed kannada digits recognition," *IJCA Proceedings on International Conference on VLSI, Communications and Instrumentation (ICVCI)*, no. 7, pp. 5–8, 2011
- [6] S. Karthik and K. Murthy, "Handwritten kannada numerals recognition using histogram of oriented gradient descriptors and support vector machines," *Advances in Intelligent Systems and Computing*, vol.2, pp. 51–57, 2015.
- [7] S. V. Rajashekaradhya and P. Vanaja Ranjan, "Neural network based handwritten numeral recognition of kannada and telugu scripts," in *TENCON 2008 - 2008 IEEE Region 10 Conference*, pp. 1–5, 2008.
- [8] G. Rajput, Horakeri, Rajeswari, and C. Sidramappa, "Printed and handwritten mixed kannada numerals recognition using svm," *International Journal on Computer Science and Engineering*, vol. 2, pp. 1622- 1626, 2010.
- [9] V. Hallur and R. Hegadi, "Offline kannada handwritten numeral recognition: Holistic approach," *Proceeding of Second International Conference on Emerging Research in Computing, Information, Communication and Applications*, vol. 3, pp. 632-637, 2014.
- [10] U. Pal, N. Sharma, T. Wakabayashi, and F. Kimura, "Handwritten numeral recognition of six popular indian scripts," in *Ninth International Conference on Document Analysis and Recognition (ICDAR 2007)*, vol. 2, pp. 749–753, 2007.
- [11] F. Bovolo, L. Bruzzone, and L. Carlin, "A novel technique for subpixel image classification based on support vector machine," *IEEE Transactions on Image Processing*, vol. 19, no. 11, pp. 2983–2999, 2010.
- [12] M. Maloo and K. Kale, "Support vector machine based gujarati numeral recognition," *International Journal of Computer Science Engineering (IJCSE)*, ISSN 0975-3397, vol. 3, pp. 2595–2600, 07 2011.
- [13] H. Sajedi, "Handwriting recognition of digits, signs, and numerical strings in persian," *Computers Electrical Engineering*, vol. 49, pp. 52– 65, 01 2016.
- [14] W. Lu, "Handwritten digits' recognition using pca of histogram of oriented gradient," in *2017 IEEE Pacific Rim Conference on Communications, Computers and Signal Processing (PACRIM)*, pp. 1–5, 2017.
- [15] E. S. GATI, B. D. NIMO, and E. K. ASIAMAH, "Kannadammnist classification using skip cnn," in *2019 16th International Computer Conference on Wavelet Active Media Technology and Information Processing*, pp. 245–248, 2019.
- [16] G. Jha and H. Cecotti, "Data augmentation for handwritten digit recognition using generative adversarial networks," *Multimedia Tools and Applications*, pp. 1–14, 2020.
- [17] S. Aly and S. Almotairi, "Deep convolutional self-organizing map network for robust handwritten digit recognition," *IEEE Access*, vol. 8, pp. 107035–107045, 2020.
- [18] V. U. Prabhu, "Kannada-mnist: A new handwritten digit's dataset for the kannada language," *arXiv preprint arXiv:1908.01242*, 2019.
- [19] H. Cocotte, "Active graph based semi-supervised learning using image matching: application to handwritten digit recognition," *Pattern Recognition Letters*, vol. 73, pp. 76--82, 2016.
- [20] Hallur, Vishweshwrayya C., and R. S. Hegadi. "Handwritten Kannada numerals recognition using deep learning convolution neural network (DCNN) classifier." *CSI Transactions on ICT*, vol. 8, pp. 295-309, 2020.
- [21] Aly, Saleh, and Ahmed Mohamed. "Unknown-length handwritten numeral string recognition using cascade of pcasvmnet classifiers." *IEEE Access* vol. 7, pp. 52024-52034. 2019.
- [22] UÇAR, Emine, and Murat UÇAR. "Applying Capsule Network on Kannada-MNIST Handwritten Digit Dataset." *Natural and Engineering Sciences* (2019).
- [23] Gonzalez, Rafael C., and Richard E. Woods. "Digital image processing." (2002).

Ramesh G is currently a Research Scholar in the Department of Computer Science and Engineering, University Visvesvaraya College of Engineering (UVCE), Bangalore University, Bangalore. He has completed his B. E and M.Tech from Vishveswaraya Technological University (VTU), Karnataka. All the degrees are in Computer Science and Engineering (CS&E) discipline. He has published papers in International Reputed Journals and International Conferences. He has attended various FDP programs. His current research lies in the areas of Image Processing, Machine learning, deep learning. He is a student member of the IEEE.

Prasanna G B is currently working as a Software Engineer in a reputed Automotive Company. He has completed his B.E in Information Science and Engineering from Department of Computer Science and Engineering, University Visvesvaraya College of Engineering (UVCE), Bangalore University, Bangalore. His current interest lies in the field of Data Processing, Image Processing and Machine Learning.

Santosh V Bhat is currently working as a Software Engineer in a reputed multinational software company. He has completed his B.E in Information Science and Engineering from Department of Computer Science and Engineering,

University Visvesvaraya College of Engineering (UVCE), Bangalore University, Bangalore. His current interests lie in field of Computer Vision and Machine Learning.

Chandrashekar Naik is currently working as a Software Engineer in a reputed Multinational Company. He has completed his B.E in Information Science and Engineering from Department of Computer Science and Engineering, University Visvesvaraya College of Engineering (UVCE), Bangalore University, Bangalore. His current interests lie in field of Data Processing, Data Analytics and Data Visualization.

Champa H N has completed Bachelor of Engineering, Masters of Technology and Doctoral Degree in Computer Science and Engineering. She has 30 years of teaching experience. Currently she is Professor in the Dept. of CSE, University Visvesvaraya College of Engineering, Bangalore. She has over 20 research papers to her credit. She is currently guiding 04 Ph.D Students. Her research interests include Image processing, Artificial Intelligence, Machine learning and Database systems.