J. Electrical Systems 20-10s (2024):3298-3308

[1]Subin M. Varghese

[2]Dr. K. Aravinthan

Real Time Feature Based Finger Sign Language Recognition Using Deep Learning

Abstract: **-** Sign language recognition is one of the fastest growing research areas today. Most of the research on HCI gesture recognition is based on artificial neural networks (ANN) or hidden Markov models (HMM). There are many effective algorithms for segmentation, classification, pattern matching, and recognition. The main goal of this paper is to compare the classifiers, which will definitely help researchers to find the best solution. The most important thing in gesture recognition system is the selection of input function and classifier. To improve the recognition rate and make the recognition system robust to viewpoint changes, the concept of shape descriptors from the available feature set is introduced. K-Nearest Neighbor (KNN), Proximity Support Vector Machine (PSVM), and Naive Bayes are used as classifiers to recognize static words. The performance analysis of the proposed methods is presented along with experimental results. Comparative analysis of these methods with other popular techniques demonstrates good real-time efficiency and robustness. The experimental results demonstrate the effectiveness of the proposed work, with the recognition efficiency of the KNN classifier being 78%, the PSVM classifier being 91%, the Naive Bayes classifier being 93%, and the proposed deep learning classifier being 97%.

Keywords: Artificial Neural Networks (ANN), Hidden Markov Model (HMM), K-Nearest Neighbor (KNN), Naïve Bayes, Proximal Support Vector (PSVM), Deep Learning

1 INTRODUCTION

A sign language is a replacement of speech for hearing and mute people. Because of this reason it has provoked many researchers to work in this field. Sign languages offer a much more structured and constrained research environment than common gestures. Moreover, gesture recognition is a tool for the virtual reality environment with his/her hands. There are different sign languages all over the world. Researchers have contributed to different sign languages like American Sign Language (ASL), British Sign Language (BSL), Taiwanese Sign Language (TSL), etc. The application was extended to several international sign languages including Chinese and Arabic [12, 16]. There have been no such distinct contributions for South Indian Languages by any of the researchers in this area. There may be different regional versions available in a particular language. However, the sign language is common and applicable to any variant of language. This paper manages a framework which perceives the Communication through signing to assist individuals with such inabilities. The dumb are ignored by society because normal people do not try to communicate or learn sign language. This makes them feel isolated and uneducated. This document aims to break the gap between the common people by introducing a sign language that will allow the user to understand the meaning without the help of any interpreter. Sensor-based methods, such as data gloves, can provide accurate measurements of hands and movement. Unfortunately, these methods require extensive calibration; they also restrict the natural movement of hands and are often very expensive. Video-based methods are less invasive, but present new problems: locating the hand and segmenting it is not a trivial task. Recently, depth cameras have become popular at entry-level prices. Depth information makes the task of segmenting the hand from the background much easier. Depth information can be used to improve the segmentation process, as used in [4b], [5b], [6b], [7b]. In this paper, static gestures are recognized by giving words as input. K nearest neighbor (KNN), proximal support vector machine (PSVM), naive Bayes classifier and deep learning algorithms are used to recognize static gestures. This paper is organized as follows: Section 2 explains the related work. Section 3 explains the proposed approach. Section 4 discusses the system design of the proposed work and describes in detail the features and classifiers used to recognize static hand gestures. Section 5 explains the experiments and results.

[1] Ph.D Research Scholar, Department of Computer Science, Adaikalamatha College, Affiliated to Bharathidasan University, Vallam, Thanjavur, Tamilnadu, subinmvarghesephd2020@gmail.com

[2]Assistant Professor, Department of Computer Science, Adaikalamatha College, affiliated to Bharathidasan University, Vallam, Thanjavur, Tamilnadu, aravinthk83@gmail.com

2 RELATED WORK

The methodologies used in Sign Language recognition can be categorized into several types based on feature extraction methods, input type and the hardware dependency. Traditionally, there have been three main types of sign language recognition: hand shape classification, isolated sign language recognition, and continuous sign classification [3a]. The overall goal of this work is to help the hearing-impaired communicate with the hearing-impaired and use sign language instead of traditional language. Another application of gesture language is human-computer interaction, which uses hand gestures as input data to a computer through webcam. In HCI, a visual interface is created to provide a natural way of communication between man and machine [4a].

Earlier researchers [6] the main focus is on the capture and classification of sign language gestures. Researchers have developed a variety of signal recognition methods. Edge detection algorithms and boundary tracking were used in [9].Hand gestures are recognized automatically using the shape descriptors. The image of the hand gesture is grabbed and converted into feature vector [19]. The hand gesture input is taken with the help of a data glove and artificial neural networks are used to recognize the gesture [8]. Sara Bilal et al. [4] a system for automatic translation of static and dynamic gestures of Indian Sign Language was developed one prominent approach describes vision-based recognition techniques [19], where visual information is obtained in the form of feature vectors. Gestures are represented as a hierarchy of multi-scale color images [21]. In some systems more than one feature extraction methods and neural networks are implemented to recognize the gestures made by hand [6]. Additionally, other papers have explored HCRF [8a] and other variants for gesture recognition. Morenci et al. used LD-CRF [9a] for gesture recognition on continuous image streams and achieved excellent results. Elmezan et al. [10a] also studied CRF, HCRF, and LD-CRF in recognizing alphabetic characters and numbers drawn in the air using manual trajectories, and each model achieved 91.52%, 95.28%, and 98.05% respectively.

3 PROPOSED APPROACH

The objective discussed in this paper is a vision based identification static signs of Sign Language. The system deals with images of bare hands which provide an easy interaction with the system. Gestures are of two types. i) Static gesture and ii) Dynamic gesture. Fig.1 shows the block diagram of the Sign Language Recognition system. The proposed work consists of three stages. First stage is preprocessing, were in the sample images are processed by using the following steps i) resizing ii) gray conversion iii) filtering iv) black and white conversion. Second stage is the feature extraction, which extracts the required feature vectors from the output obtained from the first stage. Features like solidity, eccentricity, perimeter, convex area, Major axis length, Minor axis length, orientation are used to obtain the shape. Third phase is the classification where three different classifiers are used to find better accuracy. The classifiers used are K-Nearest Neighbor (KNN), Proximal Support Vector Machine (PSVM) Naïve Bayesian and Deep Learning algorithm.

4 METHODOLOGY USED

A vision based analysis is used in this work. Vision based analysis, is based on the way human beings perceive information about their surroundings, it is probably the most difficult to implement in a satisfactory way. Several different approaches have been tested so far.

- One is to build a three-dimensional model of the human hand. The model is matched to images of the hand by one and parameters corresponding to palm orientation and joint angles are estimated. These parameters are then used to perform gesture classification.
- Second one to capture the image using a camera then extract some features and those features are used as input in a classification algorithm for classification. In this work we have used second method for modeling the system. Images are captured using a camera and the features are extracted and for the extracted feature a classifier is applied to classify the signs. Fig.1 shows the block chart of indication language acknowledgment framework.

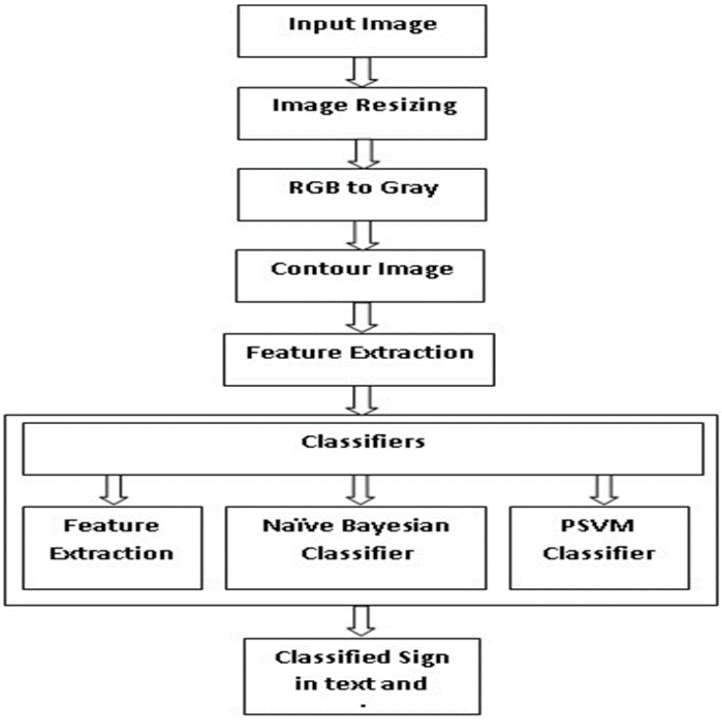

Fig.1 Block Diagram of Sign Language Recognition System

The system consists of the following stages

- Input image
- Preprocessing
- Feature extraction
- Classification.

4.1 Image Acquisition

The main phase of any vision framework is the picture securing stage. Static hand gestures and facial gestures were captured using USB connected camera. Each image represents a unique sign word. The resolution of the grabbed image is large so it is resized to a resolution of 200 into 200, which is given as input to the next stage of the model i.e., preprocessing. The sample images used are shown in Fig. 2.

Fig. 2 Collection of Static Images

4.2 Preprocessing

The preprocessing method uses a small neighborhood of pixels in the input image to obtain new brightness values in the output image. It consists of two steps.

- Segmentation
- Gaussian filtering

Segmentation is done to convert gray scale images into binary image. The obtained image has some noise. So it is better to filter that noise using Gaussian filtering approach. This method can help us obtain smooth, closed and complete gesture contours. The result obtained at this stage is a black and white image, using functions such as RGB to gray scale conversion, filtering and thresholding. Fig. 3 shows the result of the preprocessed stage.

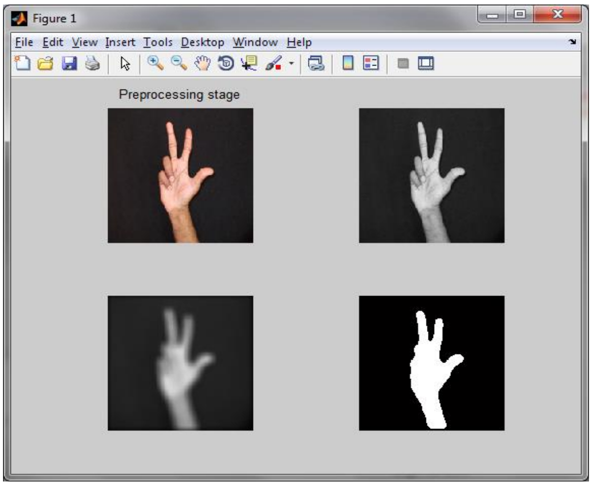

Fig. 3 Result of Preprocessing Stage

4.3 Feature extraction

Region-based analysis [9, 20] exploits both boundary and interior pixels of an object. The following are the shape descriptors used as features (a) Solidity (b) Eccentricity (c) Perimeter (d) Convex area (e) Major axis length (f) Minor axis length (g) Orientation. These features are described in the following sub sections.

4.3.1 Solidity

Scalar, defined as the ratio of the area to the convex area of the same object. The calculation formula is

$$Solidity = \frac{Area}{Convex\ Area}$$

For a solid object or cell, this value is 1.

4.3.2 Eccentricity

A scalar quantity which is defined to be ratio of the major to the minor axis. The value is between 0 and 1. It is given by the equation

$$Eccentricity = \frac{\text{Minor length axis}}{\text{Major length axis}}$$

4.3.3 Perimeter

A non-vector value, defined to be ratio of the major to the minor axis. The quantity is between 0 and 1 which is derived by the calculation.

4.3.4 Convex area

A scalar that indicates how many pixels make up a "convex image" The image is the same size as the region's bounding box. This feature is limited to input label matrices that are two dimensions.

4.3.5 Major axis length

Scalar specifying the length (in pixels) of the major axis of the ellipse that has the same normalized second central moments as the region. This property is supported only for 2-D input label matrices.

4.3.6 Minor axis length

A scalar that indicates the minor axis length (in pixels) of an ellipse whose normalised second central moments match those of the region. This feature is limited to input label matrices that are two dimensions.

4.3.7 Orientation

Scalar specifying the angle between the x-axis and the major axis of the ellipse with the same second moment as the area (in the range -90 to 90 degrees). This property is only supported for 2D input label arrays..

4.4 Classification

Classifier always tries to improve the classification rate by pushing classifiers into an optimized structure [10]. Three different classifiers are used in this work in order to compare and find the best classifier with high accuracy to recognize the static Sign.

K-Nearest Neighbor (KNN): The k-nearest neighbour algorithm (k-NN) is a technique for classifying objects in pattern recognition based on the closest training samples in the feature space. K-NN is a kind of lazy learning, or instance-based learning, in which all computation is postponed until classification and the function is only locally approximated. Using an instance-based classifier, classification (generalization) [5] can be as easy as finding the closest neighbor in instance space and assigning the same class label to the unknown instance as the neighbor that was identified (known). This approach is often referred to as a neighbor classifier. A classifier always tries to improve the classification rate by pushing classifiers into an optimized structure. Classification mainly concentrates on finding the best matching features vector for the new vector among the set of reference features. K-NN [5, 2] is one of the most commonly used methods in sign language recognition systems. It uses feature vectors generated during the training phase to get the *k*-NN in a dimensional space. The majority vote of the neighbours classifies the characteristics vector. From a group of items for which the proper classification is known, neighbours are selectedThe difference between the query and the target shape feature vectors is computed using Euclidean distance measurements, which yield the approximate number of nearest neighbours.

SVM Classifiers*:* The support vector machine (SVM) has performed successfully in many real-world problems. The SVM is attractive in its ability which condenses the information contained in the training set and to find a decision surface determined by certain points in the training set. For multiclass problems [14], the computation can be very challenging even for moderately sized datasets if the number of classes *k* is large. Proximal support vector machines (PSVM) were recently introduced as a variant of SVM [1] for binary classifications. Recently a simpler classifier, the proximal support vector machine (PSVM) [18, 3], were implemented. In proximal support vector classification two parallel planes are generated such that each plane is closest to one of two datasets to be classified and the two planes are as far apart as possible**.** But training a PSVM classifier by only solving a set of linear equations is substantially faster. Extensions of the PSVM multiclass problems [10, 14, 15] have been considered based on the one-versus-rest scheme. The main idea here is to solve *k* binary classification problems by separating one class from the rest, then construct the decision rule according to the maximal output. This approach has been considered in this work for classifying images belonging to different set of classes. There are two classes of SVM the standard SVM and the PSVM. Within SVM, a two-dimensional input space is mapped into a three-dimensional feature by the kernel function Φ (x). as shown in fig. 4(a) and (b). A kernel function may be any of the symmetric functions that satisfy the Mercer's conditions [22]. There are several SVM kernel functions as given below.

Polynomial Kernel: (xt.xi+1)p (2)

Gaussian Kernel: exp[-‖xt - xi‖2/ 2σ2] (3)

Sigmoidal Kernel:

Tanh(β0(xt.xi) + β1) (4)

where x is the input patterns, xi is the support vectors, p is the degree of polynomial. σ2 is variance, 1≤ i ≤ Ns, where Ns is the number of support vectors.

β0 and β1 are constant values.

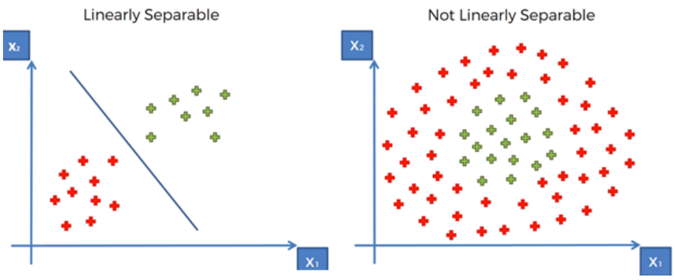

Fig.4 Overview of SVM (a) non-linear problem, (b) linear problem

Proximal SVM Classifier*:* The proximal SVM also uses a hyper plane $w.\,x + b = 0$ as the separating surface between positive and negative training examples. However, the following problem must be solved in order to determine the parameters w and b.

$$\min \frac{1}{2}\left(\|w\|^2 + b^2\right) + C\sum_i \xi_i^2$$

$$s.t. \forall i, y_i\left(w.x_i + b\right) + \xi_i \geq 1,$$

The main difference between standard SVM and proximal SVM is the constraints. While proximal SVM uses an equality requirement, standard SVM uses an inequality constraint. We can see that standard SVM only considers points on the wrong side of $w.\,xi + b = 1$ and $w.\,xi + b = -1$ as training errors. The experiment considered three possible choices of kernel functions; the Gaussian, Quadratic and Linear. For the Gaussian kernel, a coarse-to-fine grid search was conducted in the hyper parameter space in order to find a optimum . For each trained machine, the testing dataset twice was evaluated: at first using the *1-vs-1* voting scheme, then with the DDAG decision. We have annotated the performance of the classifiers, measured in terms of Cohen's kappa ( ), the total number of unique support vectors needed in the voting scheme and the average number of vector evaluations in the DDAG decision path. As linear machines can also be written in a compact form, for linear machines we consider the number of machine evaluations instead of vector evaluations.

Naive Bayesian Classifier

A Naïve Bayesian classifier assigns a new observation to the most probable class, assuming that the features are conditionally independent given the class value [14]. It can outperform more sophisticated classification methods by categorizing incoming objects to their appropriate class. Whether continuous or categorical, an arbitrary number of independent variables can be handled by Naive Bayesian classifiers.

The Naive Bayesian classifier is used to justify the objects using new methods to get a maximum. In each image, a measure of properties is taken to determine the sign in different position. They estimate the probability that a sign belongs to each of the target classes that is predetermined. In the training phase, the training set is used to decide how the parameters must be weighted and combined in order to separate the various classes of signs.

It classifies data in two steps:

Training step: This method uses training samples to estimate the parameters of the probability distribution, assuming that the features are conditionally independent given the class.

Prediction step: The method calculates the posterior probability of any unseen test sample belonging to each class. The test sample is then categorised by the procedure based on the largest posterior probability.

Bayes theorem used, takes the equation as given in (1) and (2)

$P(H|X) = P(X|H)P(H)/P(X)$ …….. (1)

It can also be expressed as

$$P(C_i|\mathrm{X}) = \frac{\mathrm{P}(\mathrm{X}|C_i)\mathrm{P}(C_i)}{\mathrm{P}(\mathrm{X})} \quad \ldots\ldots\ldots (2)$$

Where C is a constant for all classes only need be maximized.

The class-conditional independence assumption greatly simplifies the training step since estimation can be done using one-dimensional class-conditional density for each feature individually. This assumption of class independence allows the Naive Bayesian classifier to better estimate the parameters required for accurate classification while using less training data than many other classifiers. This makes it particularly effective for datasets containing many predictors or features.

Proposed Deep Learning

In the process of deep learning, it is necessary to calculate the characteristics of the learning flow change interval, obtain the flow fluctuation range, and define it as the learning sample of flow change characteristics [20].

In the model space, the flow change interval is defined as *QN*, which represents the problems encountered in the process of deep learning and represents the range of deep learning, under which the flow change has a relatively stable confidence. Assuming that the confidence of the initial change coefficient of the flow change interval *QN* is $(1-s)\%$, *a* and *o* represent the minimum critical value and the maximum critical value, respectively. The range value corresponding to $(1-s)\%$ belongs to the range of confidence value interval. According to the influence of confidence on the fluctuation interval, it is defined as the interval fluctuation judgment index, and then the interval depth coefficient (Q_{QNVQ}) and interval judgment square root weight (Q_{QNITR}) are obtained, where Q_{QNVQ} represents the accuracy of the depth learning algorithm and Q_{QNITR} represents the uniformity of the deep learning algorithm. The corresponding expressions are

$$Q_{\mathrm{QNVQ}} = \frac{1}{i}\sum_{n=1}^{i} v_n, \quad (1)$$

$$Q_{\mathrm{QNITR}} = \frac{1}{T}\sqrt{\frac{1}{i}\sum_{n=1}^{i}(o_n - a_n)^2}, \quad (2)$$

where *i* and *T* represent the measurement coefficient and peak range of flow interval, respectively

5 EXPERIMENTS AND RESULTS

The system for recognizing a set of sign words using three different classifiers has been developed by using MATLAB R2010a which is processed in a Windows 8 Operating system. The acronym MATLAB, or Matrix Laboratory, refers to a cutting-edge mathematical software programme that is widely utilised in both industry and academics. The proposed work was trained and tested with 41 categories each containing 10 subjects. **Leave-one-out-cross validation** method is used. The Sign Language Dataset contains 410 samples for each of 41 signs, recorded from 10 different persons. Each sample has a RGB image and a depth image. The sign April, July and August are not used, because these signs have motion and the proposed model only works with static signs. The dataset has variety of background and viewing angles. Due to the variety in the orientation when the signal is performed, signs become strongly similar. Figure 3 shows the most similar signs March, May, June, and January.

Fig.5 Similar Gestures

The examples are taken from the same user. It is easy to identify the similarity between these signs, all are represented by a opened fist, and differ only by the thumb position, leading to higher confusion levels. Therefore, these signs are the most difficult to differentiate in the classification task. The accuracy of the system is calculated by taking different number of features into consideration and the comparison chart is shown in Fig. 6

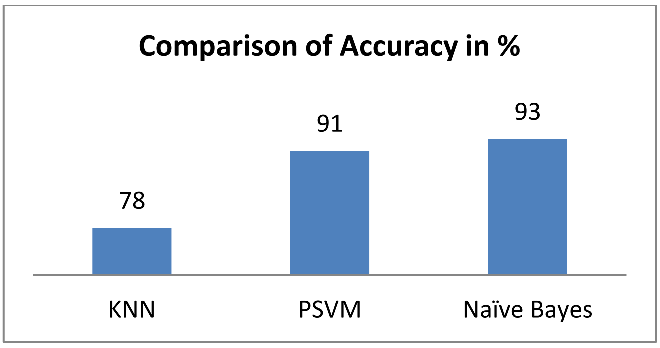

Fig. 6 Comparison Chart of Classifiers

The data must be split into three distinct sets in order to obtain genuine error estimates and model selection at the same time.

Training set: a set of examples used for learning, to fit the parameters of the classifier.

Validation set: a set of examples used to tune the parameters of a classifier.

Test set: a collection of cases intended solely for evaluating a fully-trained classifier's performance. Of all the machine learning algorithms, the K-Nearest Neighbour algorithm is the most straightforward. An object is allocated to the class most frequent among its k nearest neighbours (k is a positive integer, usually small) based on a majority vote of its neighbours. The object is simply put into the class of its closest neighbour if k = 1.

The training process for KNN consists only of storing the feature vectors and class labels of the training samples [27]. One major problem of using this technique is the class with the more frequent training samples would dominate the prediction of the new vector, since they more likely to come up as the neighbor of the new vector due to their large number. *k*-selection, another important issue which is to be taken into account is how to choose a suitable *k* for this algorithm. Generally, according to shakhnarovish et.al [23], larger values of k reduce the effect of noise on the classification, but make boundaries between distinct classes. Selecting the right k is crucial to improving the classification's effectiveness..

A variety of applications for human-computer interaction are made possible by the significant and difficult problem of recognising human motions and facial expressions in image sequences. The best system to process the frames generated while processing the continuous gestures would be one that has been evolved to a level of strength sufficient to process the static gestures. Since the collected signs were of different shapes, scales and brightness, all the signs could not be perfectly recognized by a single classifier.

The performance evaluation and the comparison of the performance measures of the classifiers are shown in Table.1and Fig. 7 respectively. Though, a maximum of the gestures are recognized to a higher accuracy with three different classifiers namely K-Nearest Neighbor (KNN), Proximal Support Vector Machine (PSVM) and Naïve Bayesian., Naive Bayesian classifier is found to the best classifier with an accuracy rate of 93% where as 91% for PSVM classifier and for K-Nearest Neighbor it is 78%.

Table 1 Performance Evaluation of NB, PSVM and KNN Classifiers

S.No	Performance Measures	NB	PSVM	KNN	PDL
1	Classified rate	0.96	0.94	0.92	0.97
2	Sensitivity	0.71	0.67	0.57	0.75
3	Specificity	0.85	1.00	0.68	0.89
4	Error rate	0.34	0.50	0.40	0.17
5	Inconclusive rate	0	0	0	0
6	Positive Predictive value	0.55	0.42	1.00	0.55
7	Negative Predictive value	0.92	1.00	0.30	0.98
8	Negative likelihood	0.33	0.53	0.63	0.33
9	Positive likelihood	2.60	1.86	1.77	3.10
10	Prevalence	0.20	0.40	0.20	0.20

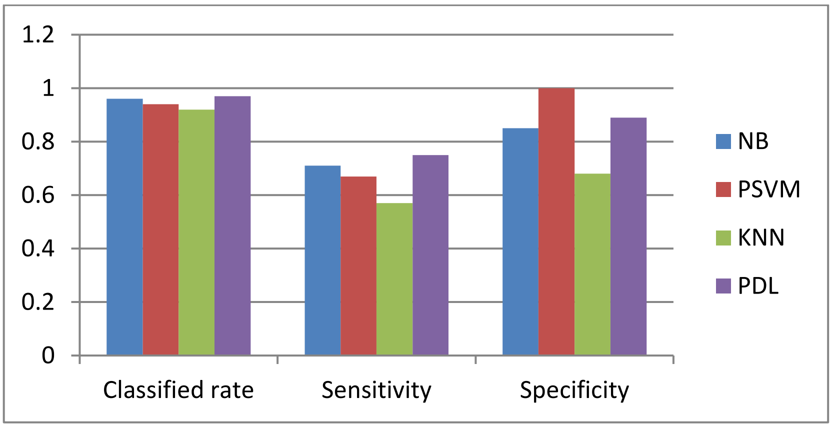

Fig. 8 Comparison of Performance Measures of the Classifiers

6 CONCLUSION AND FUTURE WORK

Sign language recognition is a wide area of research. An analysis of different classifiers is done in which the Naive bayes approach is proved to be the better for sign language recognition system. The aim of this work is to develop a sign language recognition system for deaf-dumb people. In this project, an image processing technique has been presented and designed for recognizing the signs of language for deaf-dumb persons. In this work more data has been collected and processed. Instead of taking only static hand gestures additionally hand with facial gestures are also taken. So, a large set of data are processed with extracted features called moment descriptors which are classified by using three different classifiers namely K-Nearest Neighbor (KNN), Proximal Support Vector Machine (PSVM) and Naïve Bayesian. The results of the classification technique is evaluated and found that Naïve Bayesian works well with 93% accuracy where as 91% for PSVM classifier and for K-Nearest Neighbor it is 78%. The work presented in this paper recognizes static signs only. In future, the work can be extended to recognize the dynamic signs of Sign Language. Now, the system deals with images with, uniform background, but it could be made background independent. This research result developed here is a principled technique that will enable the use, not only in sign language or hand gesture recognition but also in other related areas of computer vision.

REFERENCES

[1] Jain, Deepak, Akshi Kumar, and Geetanjali Garg. "Sarcasm detection in mash-up language using soft-attention based bi-directional LSTM and feature-rich CNN." *Applied Soft Computing* 91 (2020): 106198.

[2] Hussain, Zawar, Quan Z. Sheng, and Wei Emma Zhang. "A review and categorization of techniques on device-free human activity recognition." *Journal of Network and Computer Applications* 167 (2020): 102738.

[3] Bhuiyan, Md Rafiuzzaman, Sharun Akter Khushbu, and Md Sanzidul Islam. "A deep learning based assistive system to classify COVID-19 face mask for human safety with YOLOv3." *2020 11th international conference on computing, communication and networking technologies (ICCCNT)*. IEEE, 2020.

[4] Chunawale, Abhishek, and M. V. Bedekar. "Human emotion recognition using physiological signals: A survey." *2nd International Conference on Communication & Information Processing (ICCIP)*. 2020.

[5] Chunawale, Abhishek, and M. V. Bedekar. "Human emotion recognition using physiological signals: A survey." *2nd International Conference on Communication & Information Processing (ICCIP)*. 2020.

[6] ud din Tahir, Sheikh Badar, Ahmad Jalal, and Mouazma Batool. "Wearable sensors for activity analysis using SMO-based random forest over smart home and sports datasets." *2020 3rd International Conference on Advancements in Computational Sciences (ICACS)*. IEEE, 2020.

[7] Tamura, Hitoshi, Kazuki Itou, and Yasushi Kambayashi. "A User Independent Method for Identifying Hand Gestures withsEMG." (2022).

[8] Acheampong, Francisca Adoma, Henry Nunoo-Mensah, and Wenyu Chen. "Transformer models for text-based emotion detection: a review of BERT-based approaches." *Artificial Intelligence Review* (2021): 1-41.

[9] Kottursamy, Kottilingam. "A review on finding efficient approach to detect customer emotion analysis using deep learning analysis." *Journal of Trends in Computer Science and Smart Technology* 3.2 (2021): 95-113.

[10] Mozafari, Marzieh, Reza Farahbakhsh, and Noël Crespi. "Hate speech detection and racial bias mitigation in social media based on BERT model." *PloS one* 15.8 (2020): e0237861.

[11] Sharma, Vipul, and Roohie Naaz Mir. "A comprehensive and systematic look up into deep learning based object detection techniques: A review." *Computer Science Review* 38 (2020): 100301.

[12] Golestani, Negar, and Mahta Moghaddam. "Human activity recognition using magnetic induction-based motion signals and deep recurrent neural networks." *Nature communications* 11.1 (2020): 1551.

[13] Nasir, Jamal Abdul, Osama Subhani Khan, and Iraklis Varlamis. "Fake news detection: A hybrid CNN-RNN based deep learning approach." *International Journal of Information Management Data Insights* 1.1 (2021): 100007.

[14] Garcia-Garcia, Belmar, Thierry Bouwmans, and Alberto Jorge Rosales Silva. "Background subtraction in real applications: Challenges, current models and future directions." *Computer Science Review* 35 (2020): 100204.

[15] Long, Shangbang, Xin He, and Cong Yao. "Scene text detection and recognition: The deep learning era." *International Journal of Computer Vision* 129 (2021): 161-184.

[16] Sirohi, Deepika, Neeraj Kumar, and Prashant Singh Rana. "Convolutional neural networks for 5G-enabled intelligent transportation system: A systematic review." *Computer Communications* 153 (2020): 459-498.

[17] Raj, Suman, and Sarfaraz Masood. "Analysis and detection of autism spectrum disorder using machine learning techniques." *Procedia Computer Science* 167 (2020): 994-1004.

[18] Bogdashina, Olga. *Communication issues in autism and Asperger syndrome: Do we speak the same language?*. Jessica Kingsley Publishers, 2022.

[19] Wang, C. Jason, Chun Y. Ng, and Robert H. Brook. "Response to COVID-19 in Taiwan: big data analytics, new technology, and proactive testing." *Jama* 323.14 (2020): 1341-1342.

[20] Chen, Yucheng, Yingli Tian, and Mingyi He. "Monocular human pose estimation: A survey of deep learning-based methods." *Computer Vision and Image Understanding* 192 (2020): 102897.

[21] Henry, Brandon Michael, and Giuseppe Lippi. "Chronic kidney disease is associated with severe coronavirus disease 2019 (COVID-19) infection." *International urology and nephrology* 52 (2020): 1193-1194.

[22] Gehrig, Daniel, et al. "Video to events: Recycling video datasets for event cameras." *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020.

[23] Aldweesh, Arwa, Abdelouahid Derhab, and Ahmed Z. Emam. "Deep learning approaches for anomaly-based intrusion detection systems: A survey, taxonomy, and open issues." *Knowledge-Based Systems* 189 (2020): 105124.

[24] Corcoran, Cheryl Mary, and Guillermo A. Cecchi. "Using language processing and speech analysis for the identification of psychosis and other disorders." *Biological Psychiatry: Cognitive Neuroscience and Neuroimaging* 5.8 (2020): 770-779.

[25] Wang, Pin, En Fan, and Peng Wang. "Comparative analysis of image classification algorithms based on traditional machine learning and deep learning." *Pattern Recognition Letters* 141 (2021): 61-67.

[26] Pandya, Sharnil, and Hemant Ghayvat. "Ambient acoustic event assistive framework for identification, detection, and recognition of unknown acoustic events of a residence." *Advanced Engineering Informatics* 47 (2021): 101238.

[27] Saarijärvi, Markus, and Ewa-Lena Bratt. "When face-to-face interviews are not possible: tips and tricks for video, telephone, online chat, and email interviews in qualitative research." (2021): 392-396.

[28] Zhong, Xu, Elaheh ShafieiBavani, and Antonio Jimeno Yepes. "Image-based table recognition: data, model, and evaluation." *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXI 16*. Springer International Publishing, 2020.
[29] Karlekar, Aditya, and Ayan Seal. "SoyNet: Soybean leaf diseases classification." *Computers and Electronics in Agriculture* 172 (2020): 105342.