Single-Handed Indian Sign language - Alphabets and Numbers Gesture Recognition System

Journal Name :	Jordanian Journal of Computers and Information Technology
Manuscript ID :	71-1639888607
Manuscript Type :	Original Article
Submission Date :	19-Dec-2021
Authors :	Harshit Bhanushali |Pravin Futane |Suvarna Pawar |Manohar Kodmelwar |
Abstract :	Sign language helps build communication in addition to bridging missing gaps, especially between a person with hearing and speaking disabilities and a perfectly abled person. This research proposes a method where, sign in the form of a gesture is given as an input to the system and then, to be able to use the project in a real scenario, a dataset of 18000 images has been used, 500 images per alphabet and numeral. The first step performed is skin segmentation, which is based on extraction of skin colour pixels to detect the shape of the sign. Further, surf technique is used to extract descriptors and then the descriptors are then clustered into similar groups and a histogram using bag-of-features technique is generated. After extracting features from the images, CNN and KNN are used to classify the sign. The result of the experiments have resulted in an accuracy of 92.52%.
Keywords :	Indian Sign Language, Feature Extraction, CNN, Mini-Batch K-means, Bag-of-Features.

SINGLE HANDED INDIAN SIGN LANGUAGE – ALPHABETS AND NUMBERS GESTURE RECOGNITION SYSTEM

Pravin Futane[1], Harshit Bhanushali[2], Suvarna Pawar[1], Manohar Kodmelwar[1]

ABSTRACT

Sign language helps build communication in addition to bridging missing gaps, especially between a person with hearing and speaking disabilities and a perfectly abled person. This research proposes a method where, sign in the form of a gesture is given as an input to the system and then, to be able to use the project in a real scenario, a dataset of 18000 images has been used, 500 images per alphabet and numeral. The first step performed is skin segmentation, which is based on extraction of skin colour pixels to detect the shape of the sign. Further, surf technique is used to extract descriptors and then the descriptors are then clustered into similar groups and a histogram using bag-of-features technique is generated. After extracting features from the images, CNN and KNN are used to classify the sign. The result of the experiments have resulted in an accuracy of 92.52%.

KEYWORDS

Indian Sign Language, Feature Extraction, CNN, Mini-Batch K-means, Bag-of-Features`

1. INTRODUCTION

A sign language is composed of is various gestures formed by different orientations of hands, either single-handed or double-handed, as well as dynamic gestures. People with hearing and speaking disabilities face barriers in communication especially in public spaces. Sign language is how the hearing impaired contribute to a conversation and are able to express their emotions seamlessly, thus being able to lead a normal life. Sign language is a complete language in its own which has its syntax and grammar. The sign language used in the Indian subcontinent is known as the Indian Sign Language. The Indian Sign Language is a naturally evolved language like other oral languages. It involves naturally evolved visual-manual signs. It can be then effectively in many types of educational settings. The Indian Sign Language is a good base to demonstrate the use of human machine interfaces (HCI). In the proposed approach, the focus is on developing an Indian Sign Language recognition system in the domain of alphabets and numerals accurately, in real time. Hence, the speed, simplicity and accuracy is of utmost importance in the concerned approach. The algorithm designed is divided into 5 modules, namely: Image Pre-processing, Dividing the Dataset, Creating Histograms, Classification Models, and Reverse Recognition (speech-to-text). The initially steps include skin segmentation based on skin colour statistics, followed by feature extraction using SURF, and finally using various classification algorithms like Convolution Neural Network (CNN), KNN, Naïve-Bayes, Logistic-regression and SVM. Any gesture recognition consists of 2 different approaches; 1) Glove Based Approach, 2) Vision based Approach. Glove based approach involves wearing censored gloves and has limitations as it requires for the signer to bear the hardware during the entire process. The latter approach is more popular in pattern recognition and it has been followed in the proposed approach.

2. RESEARCH WORK

The following section compiles all the research carried out on existing systems and methodologies used in existing systems to arrive at faster, better, and efficient approach.

1. Pravin Futane, Suvarna Pawar, Manohar Kodmelwar are the Faculty at Department of Information technology, Vishwakarma Institute of Information Technology, Pune, India. Email: pravin.futane@viit.ac.in, suvarna.pawar@viit.ac.in, manohar.kodmelwar@viit.ac.in.
2. Harshit Bhanushali is a Research Intern at the Department of Computer Engineering, Vishwakarma Institute of Information Technology, Pune, India. Email: harshit.21810372@viit.ac.in

2.1 Indian Sign Language Recognition System

The proposed approach involves segmenting the hand based on the skin colour, then convert that segmented image into binary, apply feature extraction on the binary image obtained, for extraction of the features the techniques used are Euclidean distance (for distance transformation), Discrete Fourier Transform, Probability distribution property that is central moments, Hu moments and for classification and recognition Artificial Neural Network (ANN) and SVM are used. A total of 848 images comprising of 17 hand signs have been used from a publicly available dataset. Out of 848, 300 images have been used for testing purposes and the rest for training. The dimensions of the images are 320*240 dpi. Artificial Neural Network (ANN) produced an average accuracy of 94.37% and SVM classifier produced an accuracy of 92.12%. Both the classifiers showed higher accuracy with 13 features as opposed to 6 and 8 feature sets experimented with. ANN gives better accuracy even with a smaller number of feature set [1].

2.2 Indian Sign Language Recognition using neural networks and KNN classifiers

A numeric sign custom database containing 5000 images, 500 images per sign has been used for the purpose. The dataset has been created using 100 different people, 69 male and 31 females. 5 images for each sign have been contributed by each individual. Feature extraction is done using hierarchical centroid and Direct pixel value. Neural Network Pattern Recognition Tool (NPR Tool) and KNN classifiers are then used for recognition of the signs. The KNN classifier has an accuracy rate of 100%, but only for such a user who has already contributed to the dataset. KNN produces a recognition rate of 78.63% with the Direct pixel value feature extractor, and 70.96% with Hierarchical Centroid. NPR Tool produces a recognition rate of 97.10% with the Direct pixel value feature extractor, and 94.30% with Hierarchical Centroid. The ratio of training dataset to testing dataset is 70% : 30%. According to the experiments, the direct pixel value feature extraction performs better than that of the hierarchical centroid method. The system gets confused between the signs ‘9’ and ‘5’ as both have very similar shapes. Along with that, signs ‘8’ and ‘9’, and signs ‘4’, ‘5’ are also not accurately predicted by the system. The system can only be useful for static ISL numeric signs and hence cannot be considered as a complete system [2].

2.3 Artificial Neural Network Based Method for Indian Sign Language Recognition

In the proposed system, for segmentation RGB colour space is transformed into YCbCr colour space. To identify the skin coloured pixels, thresholding technique is used based on the skin colour based distribution of the YCbCr colour space. Thus, the segmentation phase ends. For feature extraction, initially Euclidean distance transformation along with City Block, Chessboard have been used as 3 different distance measures for distance transformation. Row and column projection vectors have been applied on distance transformed image to create 2 1-Dimensional modules which represent the shape of the hand for the input image in consideration. Fourier descriptor is applied on row and column projected image. For feature vectors, central moments along with variance, skewness and Kurtosis have been used. Kurtosis helps in measuring peak of a probability distribution. The ANN classifier is trained to classify 36 hand signs. The training dataset contains 360 images with 10 images of each of the 36 signs. 2/3$^{rd's}$ of the dataset has been used for training purposes whereas the rest 1/3rd has been used for testing purposes. An accuracy of 91.11% has been achieved [3].

2.4 AWAAZ: A Communication System for Deaf and Dumb

For segmentation, they have used HSV histogram. HSV is preferred over RGB as HSV is more robust towards external lighting changes. For the extraction of the features, Harris algorithm is used. For feature matching and recognition, the dataset already has the feature extracted of standard image stored as N*2 matrix file. The matrix value of this image is then matched with every image and the minimum distance between the matched features is calculated to achieve the expected recognition accuracy. Noise removal and filtering is done using median filter followed by erosion and then dilation (Opening Morphology). The proposed approach fulfils the Indian sign language hand gesture recognition but has the limitation of not being able to use double-handed gestures, leading to which satisfactory results have not been achieved. Poor lighting conditions are also an obstacle. The system only takes into account the numeric single-handed signs [4].

2.5 Indian Sign Language Gesture Classification as Single or Double Handed Gestures

The gestures have been classified into sub-categories which simplifies the gesture recognition process due to presence of lesser number of gestures in each sub-category. Feature extraction was done using Histogram of Gradients (HOG) features and geometric descriptors such as solidity, equivalent diameter to Minor Axis Ratio, Eccentricity, Major to minor Axis Ratio and Area to area of Bounding box. KNN and SVM classifiers have been used on a dataset consisting of images of 26 English alphabets present in the ISL under variable background. KNN when applied on Geometric feature set 1 resulted in an accuracy of 65.38%, with Geometric feature set 2, the accuracy achieved was 69.23%and with HOG features it was 88.46%. SVM when combined with Geometric feature set 1 reached a recognition rate of 75.08% (correctly classified 38 gestures out of 52), with Geometric feature set 2, 1 reached a recognition rate of 80.77% was achieved (correctly classified 42 gestures out of 52) and with HOG features, an accuracy of 94.23% was achieved (Correctly classified 49 gestures out of 52). A custom-made dataset consisting of 260 images consisting of 26 different gestures was collected for the experiments. A total of seven different individuals were chosen to help in creation of the dataset under different intensities of lights and background. 80% of the images from the dataset were used for training purpose and the rest 20% for testing purposes. In conclusion, Support Vector Machine classifiers showcased a better accuracy than K-Nearest Neighbours classifier in terms of both geometric and HOG features [5].

2.6 Indian Sign Language Recognition

The proposed method has been successful in recognizing alphabets and numbers both. Recognition is done using the Principal Component Analysis (PCA). The propose approach follows a simple initial approach of RGB Thresholding, followed by creating red, green and blue masks, followed by creating object masks and then in the end concatenating all the individual masked components for skin segmentation. The issue of segmentation of overlapping of hands in some signs has been tackled with great accuracy in the proposed algorithm. For extracting features, a fingertip finding algorithm has been implemented using Thinning algorithm (for Distance transformation) followed by finding the perimeter pixels of the image, followed by finding corner points which has been achieved using Harris Corner detection, and then finally eliminating the corner points that do not satisfy the fingertip criteria. The individual doffed gloves of blue and red colours for ease of processing. For clustering, PCA has been used. In case of live images, individual frames of live video have been given as an input and every twentieth frame has been recognised. The accuracy achieved using PCA is 94%. 15 images for individual gestures have been captured. Total images captured for twenty-five alphabetical and nine numeral gestures are 510 (34*15). The images have been captured under a uniform intensity emitting light. The proposed approach proves that combining fingertip algorithm and Principal component analysis is very efficient [6]

2.7 Real Time Sign Language using PCA

The proposed approach Otsu's method for skin segmentation. Morphology is used for noise removal and filtering purposes. For feature extraction, the principal component has been used. 260 images have been captured, 10 images for each of the alphabetical hand gestures. The dimensions of the images are 380*420 dpi. Each image in training phase is represented as a column vector. The next step includes in deriving an eigen vector of the covariance matrix. The proposed approach used the MATLAB environment for performing the task using Principal Component Analysis (PCA) and the accuracy of the recognition rate has been achieved at a considerable stage. The recognised gestures are converted into voice format, text and also features are displayed on the Graphical user Interface [7].

2.8 Indian Sign Language Recognition Using Optimized Neural Networks

The proposed approach presented 3 methods to solve the problem of gesture recognition of ISL, effectively by combining Neural Network (NN) with Genetic Algorithm, Evolutionary Algorithm and Particle Swarm Optimisation (PSO). The 3 approaches were intertwined to get the best possible accuracy. The results depicted that Neural Network alone achieved a recognition rate of 93.64%, but if combined with Particle Swarm optimization, it yielded an accuracy rate of 98.29%. Neural Network if combined with Multilayer Perceptron Feed-Forward Network, gave a better accuracy of 96.7% than using standalone Neural Network [8].

3. RESEARCH OUTCOME

The following section highlights the important discoveries from the research work carried out in a tabular form as shown in Table 1.

Table 1. Literature Survey Outcome.

Ref No.	Algorithm Used	Dataset Used	Performance Measure
[1]	Adaptive Probabilistic model (Skin detection) Morphology (Noise removal and filtering) Euclidean Distance, Fourier Descriptors, Central and Hu moments (Feature Extraction) ANN, SVM (Classification)	848 images used (17 hand signs) Dimensions - 320*240 dpi	13 features taken under consideration. Accuracy of ANN classifier is 94.37% and that of SVM classifier is 92.12%.
[2]	Sobel Edge Detector (Edge detection) Direct pixel value, Hierarchical Centroid (Feature Extraction) KNN, Neural Network Pattern Recognition Tool (NPR Tool) (Classification)	5000 images used (10 numeric hand signs) 500 images per sign Dimensions – 200*300 dpi	The KNN classifier has an accuracy rate of 100%. (For a user who has contributed to the creation of the dataset). KNN produces a recognition rate of 78.63%, while NPR tool gives 97.10% with the Direct pixel value feature extractor. The accuracy for KNN is 70.96% with Hierarchical Centroid, and 94.30% for NPR Tool.
[3]	YCbCr colour space, Thresholding (Hand Segmentation) Euclidean Distance, City Block, Chessboard, Row and Column Projectors vector, Fourier Descriptors, Central Moments, Variance, Skewness (Feature Extraction) ANN (Classification)	540 images (26 alphabets and 10 numeric hand signs) 10 images of each sign for training and 5 of each for testing.	The ANN classifier achieves an average recognition rate of 91.11%. 2/3rd of the dataset was used for training purpose and remaining 1/3rd for testing.
[4]	HSV Histogram (Skin Segmentation) Median filter, Morphology (Noise Removal and filtering) Harris Algorithm (Feature Extraction) Feature Matching	The size of the dataset has not been mentioned.	The proposed approach fulfils the Indian sign language hand gesture recognition but has the limitation of not being able to use double-handed gestures, leading to which satisfactory results have not been achieved. Poor lighting conditions are also an obstacle.
[5]	RGB Thresholding, (Hand Segmentation) Morphology (Noise Removal and filtering) Geometric Descriptors [Solidity, Major to Minor Axis Ratio, Eccentricity], Histogram of Gradients (Feature Extraction) KNN, SVM (Classification)	260 images (26 alphabets hand signs) 10 images for each sign under different lighting conditions with the assistance of 7 different people.	KNN when applied on, Geometric feature set 1 = 65.38% accuracy, Geometric feature set 2 = 69.23% accuracy, HOG features = 88.46%. SVM when applied on, Geometric feature set 1 = 75.08% accuracy, Geometric feature set 2 = 80.77% accuracy, HOG features = 94.23%.
	RGB Thresholding (Skin	510 images (25	The accuracy rate achieved using

[6]	Segmentation) Thinning using Distance transformation, Finding perimeter and corner points (Harris Corner Detection), Elimination of corner points (Feature Extraction) PCA (Classification)	alphabets and 9 numeric hand signs) 15 images per sign have been created.	PCA is 94%. Overlapping hand gestures and dynamic hand gestures create ambiguities in the recognition.
[7]	Otsu's algorithm, (Hand Segmentation) Morphology (Noise Removal and filtering) Euclidean Distance, principal component (feature extraction) PCA (Classification)	260 images (26 alphabets hand signs) 10 images for each sign. Dimensions – 380*420 dpi.	The proposed approach used the MATLAB environment for performing the task using Principal Component Analysis (PCA) and the accuracy of the recognition rate has been achieved at a considerable stage.
[8]	Neural Network, Genetic Algorithm, Particle Swarm Optimization	220 images (22 hand gestures) 10 images per sign have been created. 70% of the dataset has been used for training and the rest 30% for testing.	Neural Network alone achieved a recognition rate of 93.64%, but if combined with Particle Swarm optimization, accuracy rate = 98.29%. Neural Network if combined with Multilayer Perceptron Feed-Forward Network, gave a better accuracy of 96.7%.

4. PROPOSED APPROACH

The following section describes in great detail the proposed methodology and the reason behind it.

4.1 Flow Diagram

The flow of the proposed approach is given in Figure 1.

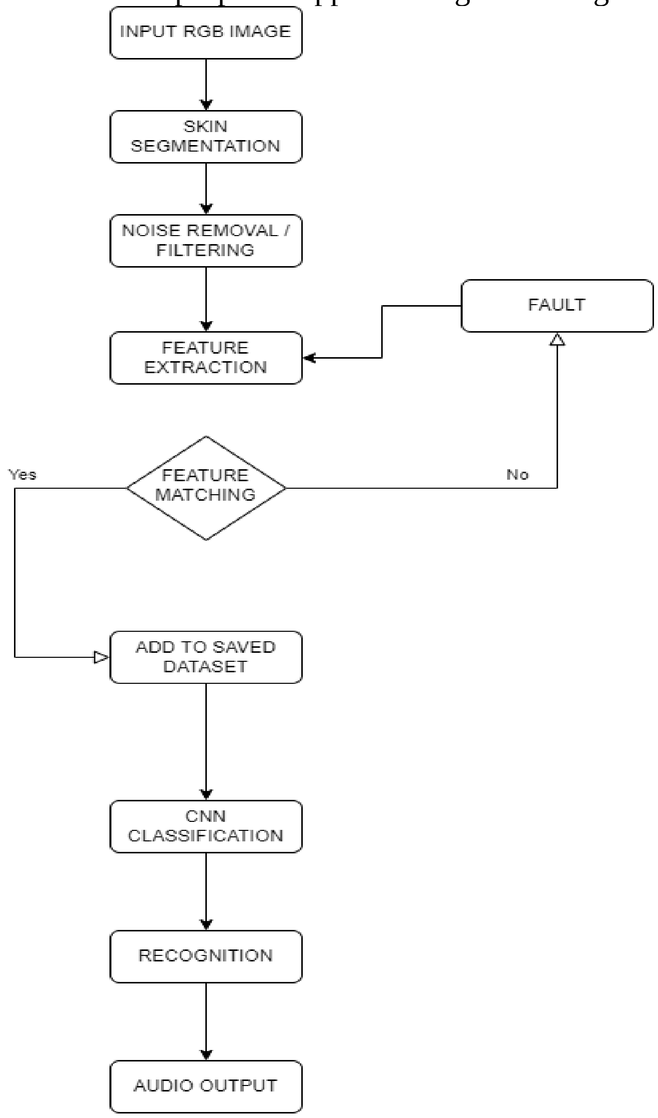

Figure 1. Architecture Diagram

4.2 Image Acquisition

A completely unprocessed is given as the input to the system initially. In this system, dataset is created using a single HD laptop camera, wherein 500 images for a single class have been captured. A total of 18000 images (A - Z and 0 – 9) have been captured. Image have been resized to the dimensions of 128*128 dpi.

4.3 Image Pre-Processing

Firstly, the RGB images were converted to grey scale using OpenCV's RGB2GRAY function. From grayscale, the image has been then converted to HSV. The next step is to define a boundary (trial and error) and produce a mask through it. (Boundary contains upper and lower level of the pixel tuple). The next step performed removes noise from the masked image. Initially, medianBlur (processes the edges while removing noise) and addWeighted (sharpens an image using smoothing filter) are used to achieve the mentioned. To better achieve edge enhancement and noise removal, morphology has been performed. Opening morphology (erosion followed by dilation) has been implemented with the use of morphologyEx function. The last step includes extracting the hand by masking and the edges, which is achieved using canny edge detection. Below diagrams in Figure 2 depict the nature of the entire image processing algorithm used.

Figure 2. Transition from Mask to Canny Edge

4.4 Feature Extraction

It is important to perform feature extraction from the images that have been cleaned in order to classify them using various classification algorithms. Following sub-sections explain the algorithms chosen for feature extraction.

4.4.1 Extraction using SURF (Speeded-Up Robust Features)

It is important to perform feature extraction from the images that have been cleaned in order to classify them using various classification algorithms. The shape that is segmented in the pre-processed image is a very critical feature during any feature extraction process. In the proposed approach, The Speeded Up Robust Feature (SURF) algorithm is employed to achieve the goal of feature extraction. The total classes are 36 (A - Z and 0 - 9). The system then performs feature extraction, using them in the bag-of-words model, which generates redundant feature points because background is involved in the extraction of feature points as shown in Figure 3.

Figure 3. SURF features

4.4.2 K-Means Clustering

K-means is used for partitioning similar data points into K-distinct, non-overlapping clusters. To implement K-means, it is important to define the desired number of clusters K. The algorithm then tries to assign each data point at-least one of the groups from the specified clusters. In the proposed approach, Bag of visual words (BOVW) is used for image classification. The general idea of bag of visual words (BOVW) is to represent an image as a set of features.

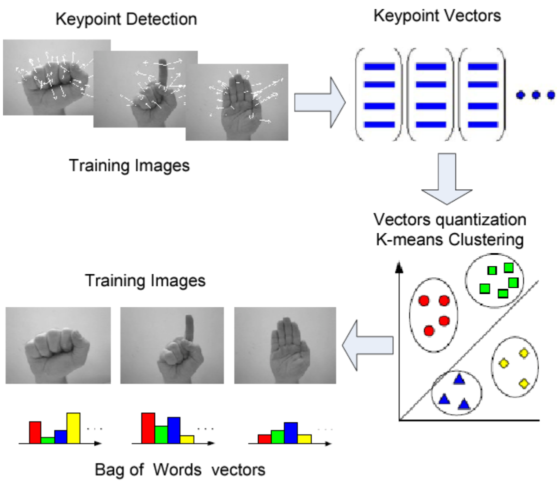

Figure 3. Generating a Bag-of-words Model [Courtesy: IEEE TRANSACTIONS ON INSTRUMENTATION AND MEASUREMENT, VOL. 60, NO. 11]

The clustering for the descriptors in the proposed approach is done using Mini-Batch K-Means**.** The K-Means algorithm is excellent because of its time performance, but with the increasing size of the dataset under consideration, it's computation time or the worst-case time complexity increases exponentially and also because of the constraint of needing the whole dataset in the main memory. To tackle this, Mini Batch K-means uses constricted random batches of fixed data points. These batches are small enough to be stored in the main memory of the system. For every new iteration, a sample from the data space is chosen and employed to update the defined clusters. This operation is continued until convergence is reached. With time, as the number of loops executed increases, the effect of new data points reduces exponentially, and then convergence is confirmed when the system is not able to find any new groups occurring in several loops.

4.5 Classification

In the proposed research, 36 classes have been taken into consideration. The dataset for this system is manually created in different hand orientations and the train-test ratio of 80:20 is used. A total of 500 images for each class, totaling to a dataset of 18000 images has been created as stated earlier. Out of 18000 images, 14400 images have been used for the training purpose while the rest have been used for testing phase. With the help of these feature extracted training samples, the classifiers have developed some model of prediction.

4.5.1 K-Nearest Neighbours (KNN)

K-NN algorithm predicts the resemblance between fresh data points and the present classes and puts the fresh data points into the classes that the data is resembling to the most from the presently available classes. It is a non-parametric algorithm, which means it does not make any assumption on underlying data. The Table 2 showcases confusion matrix for KNN classifier based on the parameters of True Positive, True Negative, False Positive, False Negative, accuracy, precision and F1-Score derived using sensitivity.

Table 2. Confusion Matrix depicting results obtained from KNN.

Sign	TP	TN	FP	FN	Precision	Accuracy	F1-Score
0	879	2104	232	385	79.11%	82.86%	74.01%
1	932	2366	177	95	84.46%	92.44%	87.61%
2	1189	2367	13	31	98.91%	98.77%	98.18%
3	777	2599	156	68	83.27%	93.77%	87.40%
4	613	2710	168	109	78.48%	92.30%	81.57%
5	602	2488	289	221	67.56%	85.83%	70.24%
6	911	2391	231	67	79.77%	91.72%	85.94%
7	876	2089	137	489	86.42%	82.36%	73.39%

8	720	2421	356	103	66.91%	87.25%	75.82%
9	649	2684	291	12	69.04%	91.66%	81.07%
A	984	2411	51	154	94.79%	94.30%	90.56%
B	1221	2377	2	0	99.83%	99.94%	99.91%
C	788	2185	274	353	74.19%	82.58%	71.53%
D	992	2443	108	57	90.18%	95.41%	92.32%
E	888	2335	258	119	77.48%	89.52%	82.48%
F	955	2383	174	88	84.58%	92.72%	87.93%
G	723	2711	59	107	92.45%	95.38%	89.70%
H	912	2661	17	10	98.17%	99.25%	98.17%
I	822	2300	93	385	89.83%	86.72%	77.47%
K	519	2469	191	421	73.09%	83%	62.09%
J	845	2217	357	181	70.29%	85.05%	75.85%
L	666	2139	259	536	72%	77.91%	62.62%
M	653	2581	61	305	91.45%	89.83%	78.11%
N	913	2413	119	155	88.46%	92.33%	86.94%
O	784	2477	54	285	93.55%	90.58%	82.22%
P	761	2298	304	237	71.45%	84.97%	73.77%
Q	872	2334	362	32	70.66%	89.05%	81.57%
R	877	2618	61	44	93.46%	97.08%	94.35%
S	934	2259	251	156	78.81%	88.69%	82.10%
T	658	2611	318	13	67.41%	90.80%	79.90%
U	917	2416	48	219	95.02%	92.58%	87.29%
V	532	2197	247	624	68.29%	75.80%	54.98%
W	605	2102	214	679	73.87%	75.19%	57.53%
X	918	2329	217	136	80.88%	90.19%	83.87%
Y	801	2497	139	163	85.21%	91.61%	84.13%
Z	653	2449	246	202	72.63%	87.38%	74.45%

The average accuracy rate of KNN is 89.36%. It has a slightly lower accuracy rate than CNN for the proposed methodology.

4.5.2 Convolutional Neural Network (CNN)

Convolutional Neural Networks require comparatively very less pre-processing than other classifiers like SVM, KNN, Naïve-Baiyes, and Logistic Regression. Following is the Table 2 that showcases confusion matrix for CNN classifier based on the parameters of True Positive, True Negative, False Positive, False Negative, accuracy, precision and F1-Score derived using sensitivity.

Table 2. Confusion Matrix depicting results obtained from CNN.

Sign	**TP**	**TN**	**FP**	**FN**	**Precision**	**Accuracy**	**F1-Score**
0	1002	2531	18	49	98.23%	98.13%	96.75%
1	917	2631	31	21	96.72%	98.55%	97.24%
2	715	2865	13	7	98.21%	99.44%	98.61%
3	701	2873	18	8	97.49%	99.27%	98.17%
4	694	2881	11	14	98.43%	99.30%	98.22%
5	670	2919	4	7	99.40%	99.69%	99.17%
6	715	2865	2	18	99.72%	99.44%	98.61%
7	879	2104	232	385	79.11%	82.86%	74.01%
8	769	2799	19	13	97.58%	99.11%	97.95%
9	1023	2214	204	159	83.37%	89.91%	84.92%
A	893	2107	411	189	68.84%	83.33%	75.06%
B	913	2413	119	155	88.46%	92.33%	86.94%
C	901	2651	21	27	97.72%	98.66%	98.66%
D	1139	2111	118	232	90.61%	90.27%	86.67%

E	1019	2063	167	351	85.91%	85.61%	79.73%
F	928	2303	187	182	83.22%	89.75%	83.41%
G	984	2411	51	154	94.79%	94.30%	90.56%
H	1189	2367	13	31	98.91%	98.77%	98.18%
I	877	2618	61	44	93.46%	97.08%	94.35%
K	798	2308	174	320	82.09%	86.27%	76.36%
J	992	2443	108	57	90.18%	95.41%	92.32%
L	717	2031	467	385	60.55%	76.33%	62.72%
M	899	2519	127	55	87.62%	94.94%	90.80%
N	653	2449	246	202	72.63%	87.38%	74.45%
O	933	2531	81	55	92.01%	96.22%	93.20%
P	1137	2314	132	17	89.59%	95.86%	93.85%
Q	801	2497	139	163	85.21%	91.61%	84.13%
R	940	2502	87	71	91.52%	95.61%	92.24%
S	714	2684	116	86	86.02%	94.38%	87.60%
T	723	2711	59	107	92.45%	95.38%	89.70%
U	1088	2444	41	27	96.36%	98.11%	96.96%
V	601	2192	304	497	66.04%	77.75%	60.01%
W	628	2117	351	504	64.14%	76.25%	59.49%
X	1111	2278	123	88	90.03%	94.13%	91.23%
Y	932	2366	177	95	84.46%	92.44%	87.61%
Z	909	2212	288	191	75.93%	86.69%	79.14%

The accuracy for any classification model can be expressed in the ratio of Sum of TP and TN of the testing dataset to the Sum of all True and False Negatives and Positives [(TP + TN) / (TP + TN + FP + FN)]. Precision can be expressed in the ratio of TP to Sum of TP and FP [TP / (TP + FP)]. Sensitivity or recall is expressed as (TP) / (TP + FN). Specificity is characterized as (TN) / (TN+FP). The final measurement is F1_Score is expressed as 2 * (Precision × Sensitivity) / (Precision + Sensitivity). The average accuracy rate of CNN is 92.52%.

5. RESULTS AND DISCUSSIONS

The dataset was divided into 2 parts i.e testing and other training. The training dataset comprised 80% of the aggregate data, while the testing dataset comprised remaining 20% of the aggregate data. The results of the experiments thus conducted have an accuracy rate of **89.36% for KNN classifier and 92.52% for CNN classifier.** The accuracy could have gone up-to a 96-97% if there weren't any similar signs under consideration. For example, in the below figures, it is evident that the signs of '2' and single-handed alphabet 'V' are exactly same. Similarly, the signs of '3' and single-handed alphabet 'W' have the exact orientation same.

Figure 4. Similar signs of 2 and V.

Figure 5. Similar signs of 3 and W.

Due to the above constraints, the accuracy rate dropped down a bit. Nonetheless, it can be inferred from the above experiment that CNN performs better than KNN under the circumstances and algorithms used for Image Pre-Processing and feature extraction.

6. CONCLUSIONS

The gesture recognition system is capable of recognising only static single-handed numerical and alphabetical hand gestures of the Indian Sign Language with an accuracy of 92.52%. The average accuracy of recognition rate for only static single-handed numerical gestures for KNN is 89.90%, whereas for static single-handed alphabetical gestures is 89.15%. Hence the accuracy rate for CNN classifier is 89.36%. On the other hand, the average accuracy of recognition rate for only static single-handed numerical gestures for CNN is 96.57%, whereas for static single-handed alphabetical gestures is 90.96%. Hence the accuracy rate for CNN classifier is 92.52%.

7. FUTURE SCOPE

The proposed system is useful for static single-handed numerical and alphabetical hand gestures of the Indian Sign Language. It cannot be considered as a fully-fledged working system as for that, double-handed gestures, dynamic gestures, words and sentences would have to be considered. The next phase of the research can include taking into considerations various other extraction algorithms, classifiers like Naïve-bayes, PCA and SVM, or a combination of these to achieve a better accuracy and term the resultant system as a full-fledged working Indian Sign Language Recognition System.

8. REFERENCES

[1] Yogeshwar I. Rokade, Prashant M. Jadav, “Indian Sign Language Recognition System”, International Journal of Engineering and Technology, July 2017.

[2] Ashok Kumar Sahoo, “Indian Sign Language Recognition using neural networks and KNN classifiers”, Journal of Engineering and Applied Sciences, August 2014.

[3] Adithya V, Vinod P. R, Usha Gopalakrishnan – “Artificial Neural Network Based Method for Indian Sign Language Recognition”, IEEE Conference on Information and Communication Technologies [ICT], IEEE, 2013.

[4] Anchal Sood, Anju Mishra, “AAWAAZ: A Communication System for Deaf and Dumb”, 5th International Conference on Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO), IEEE, 2016.

[5] Akanksha Singh, Saloni Arora, Pushkar Shukla, Ankush Mittal, “Indian Sign Language Gesture Classification as Single or Double Handed Gestures”, Third International Conference on Image Infonnation Processing, 2015.

[6] D. Deora, N. Bajaj, “Indian Sign Language System”, International Conference on Emerging Technology Trends in Electronics Communication and Networking, 2012.

[7] Sawant, Shreyashi Narayan and M. S. Kumbhar, “Real Time Sign Language Recognition Using PCA”, International Conference on Advanced Communication Control and Computing Technologies (ICACCCT), IEEE, 2014.

[8] J Sireshendu Hore, Sankhadeep Chatterjee, V. Santhi, Nilanjan Dey, Amira S. Ashour, “Indian Sign Language Recognition Using Optimized Neural Networks”, Springer International Publishing Switzerland 2017.