|  |  |  |
| --- | --- | --- |
| D:\Admission\Admissions_1011\wwwroot\hdrlogo.gif | Department of Computer Engineering  D. J. Sanghvi College of Engineering  University of Mumbai  A. Y. 2018-2019 | E:\new logo.JPG |

**Hand Gesture Detection and Classification System**

**MINI PROJECT**

By

**Maulik S. Shah 60004160094**

**Parth N. Shah 60004160099**

**Raj B. Shah 60004160101**

Guide:

**Prof. Kiran Bhowmick**

(Assistant Professor)

**CERTIFICATE**

This is to certify that the mini project entitled **“Hand Gesture Detection and Classification System”** is a bonafide work of **“Maulik S Shah (60004160094), Parth N Shah (60004160099) and Raj Bhavesh Shah (60004160101)”** submitted to the University of Mumbai in partial fulfillment of the requirement for the award of the degree of B.E. in Computer Engineering.

**(Prof Kiran Bhowmick)**

**Guide**

**(Dr Prof Narendra M Shekokar) (Dr Hari Vasudevan)**

**Head of Department Principal**

**Mini Project Report Approval.**

This mini project report entitled ***Hand Gesture Detection and Classification System*** ***by Maulik S Shah***, ***Parth N Shah*** and ***Raj B Shah*** is approved for the partial fulfillment of the degree of ***B.E. in Computer Engineering.***

Examiners

1.---------------------------------------------

2.---------------------------------------------

Date:

Place:

Declaration

We declare that this written submission represents our ideas in our own words and where others' ideas or words have been included, we have adequately cited and referenced the original sources. We also declare that we have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in our submission. We understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

-----------------------------------------

(Maulik S Shah: 60004160094)

-----------------------------------------

(Parth N Shah: 60004160099)

-----------------------------------------

(Raj B Shah: 60004160101)

Date:

**Abstract**

Human Computer Interaction (HCI) technologies are rapidly evolving the way we interact with computing devices and adapting to the constantly increasing demands of modern paradigms. One of the most useful tools in this regard is the integration of Human-to-Human Interaction gestures to facilitate communication and expressing ideas.

Yet we still limit human-computer interaction to cumbersome mice movements. The use of hand gestures in the field of human-computer interaction has attracted new interest in the past several years. Special glove-based devices have been developed to analyse finger and hand motion and use them to manipulate and explore virtual worlds. To further enrich the naturalness of the interaction, different computer vision-based techniques have been brought into use.

Touch less hand gesture recognition systems are becoming important in automotive user interfaces as they improve safety and comfort. Various computer vision algorithms have employed colour and depth cameras for hand gesture recognition, but robust classification of gestures from different subjects performed under widely varying lighting conditions is still challenging.

In this report, we present a hand gesture recognition system with a Creative Senz3D Camera, which operates robustly in uncontrolled environments and is insensitive to hand variations and distortions. Our system consists of one major module, namely, hand gesture recognition.

Different from traditional vision-based hand gesture recognition methods that use colour-markers for hand detection, our system uses both the depth and colour information from the camera to detect the hand shape, which ensures the robustness in cluttered environments.

This process was followed by data collection, a detailed analysis, and an ascertainment of the selected classification algorithm. The major objective of conducting a Systematic Literature Review (SLR), followed by the corresponding analysis of the results is to highlight the state-of-the-art in the context of vision-based gesture recognition with a specific focus on hand gesture recognition (HGR) techniques and enabling technologies.

We propose a classification method for a hand gesture recognition from challenging depth and intensity data by comparing various Machine Learning and Deep Learning techniques such as Logistic Regression, K-Nearest Neighbours, Support Vector Machine (SVM) and various types of Convolution Neural Networks. Our solution combines information from multiple spatial scales for the final prediction.

Our method achieves a correct classification rate of **96.212**% on the Senz3D dataset.

Contents

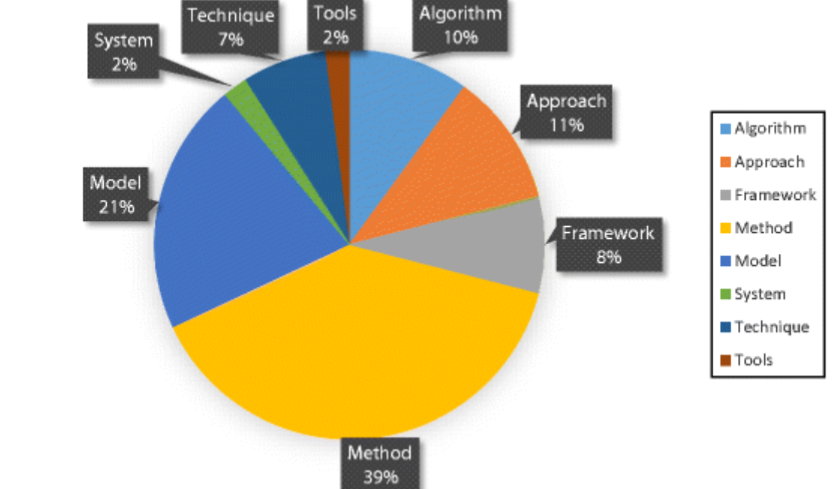
[Introduction 7](#_Toc6392136)

# Introduction

In the last decade, many vision-based dynamic hand gesture recognition algorithms were introduced [11, 16]. To recognize gestures, different features such as hand-crafted spatio-temporal descriptors [23] and articulated models [9], were used. As gesture classifiers, hidden Markov models [20], conditional random fields [24] and support vector machines (SVM) [4] have been widely used. However, robust classification of gestures under widely varying lighting conditions, and from different subjects is still a challenging problem [25, 1, 15].

Figure 1 shows a Statistical analysis of hand gesture recognition solutions according to various contribution types, that have been historically presented. Our approach is a comparative analysis of classification algorithms, as shown in the figure below.

Figure1: Statistical analysis



**Challenges for a Vision-Based System:**

The method must generalize over users and variation in the performance of the gestures. Segmentation of continuous temporal gesture events is also difficult. In particular, generalising gesture recognition and making it independent of all possible volatile environments significantly differs from gesture recognition in the constrained environment of an office.

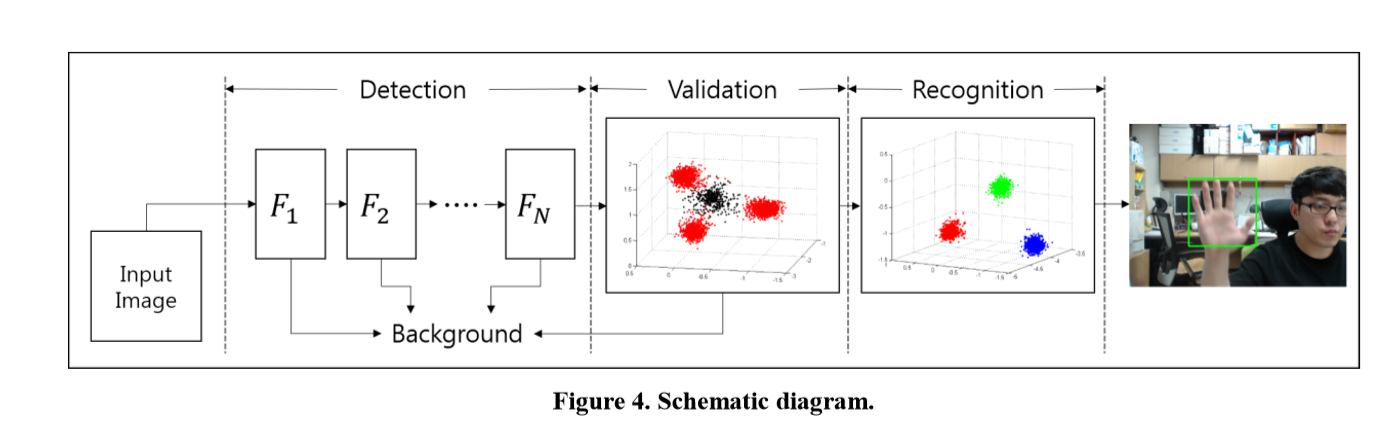
First, the algorithm must be robust to varying global illumination changes and shadow artefacts.

Second, since the Infrared sensor is mounted in front of the user in our dataset and gestures are performed away from the sensor, the hand commonly self-occludes itself throughout the performance of the gestures. Precise pose estimation (as in [103] and [104]) is difficult and was little studied before in settings of harsh illumination changes and large self-occlusion, yet many approaches rely on such pose information for producing the discriminatory features for gesture classification. Finally, fast computation (ideally real time) is desirable.

In our report,we compare various hand gesture recognition classification systems that utilize depth and intensity channels comparing various Machine Learning and Deep Learning techniques such as Logistic Regression, K-Nearest Neighbours, Support Vector Machine (SVM) and different types of Convolution Neural Networks. Our solution combines information from multiple spatial scales for the final prediction.

The Senz3D Hand Gesture Recognition dataset contains gestures performed by 4 different people, each performing 11 different gestures repeated 30 times each, for a total of 1320 samples.

# Literature Review

****

**Description:**

This paper presents a real-time hand gesture detection and recognition method. Proposed method consists of three steps: Detection, Validation and Recognition.

In the detection stage, several areas, estimated to contain hand shapes are detected by random forest hand detector over the whole image. In order to check whether each area contains hand or not, we used Linear Discriminant Analysis.

For validation, computation amount when calculating the distance from each training samples has been reduced by obtaining centroids of clusters. K-means was applied on the samples’ distribution.

Thus, we propose a real time hand

gesture recognition system using a camera. The proposed

method is based on the random forest for hand detection [2],

and the LDA is used for implementing the gesture patterns.

We define three kinds of hand postures as “deterministic

hand posture” (See Figure 1). Because of the ambiguity of

hand postures, we cannot call every posture as its definite

name. For this reason, we also classify every hand gesture

into 3 postures (3 classes). Defined 3-posutres are

represented in Figure 1.

Thus, we propose a real time hand

gesture recognition system using a camera. The proposed

method is based on the random forest for hand detection [2],

and the LDA is used for implementing the gesture patterns.

We define three kinds of hand postures as “deterministic

hand posture” (See Figure 1). Because of the ambiguity of

hand postures, we cannot call every posture as its definite

name. For this reason, we also classify every hand gesture

into 3 postures (3 classes). Defined 3-posutres are

represented in Figure 1.

Thus, we propose a real time hand

gesture recognition system using a camera. The proposed

method is based on the random forest for hand detection [2],

and the LDA is used for implementing the gesture patterns.

We define three kinds of hand postures as “deterministic

hand posture” (See Figure 1). Because of the ambiguity of

hand postures, we cannot call every posture as its definite

name. For this reason, we also classify every hand gesture

into 3 postures (3 classes). Defined 3-posutres are

represented in Figure 1.

Thus, we propose a real time hand

gesture recognition system using a camera. The proposed

method is based on the random forest for hand detection [2],

and the LDA is used for implementing the gesture patterns.

We define three kinds of hand postures as “deterministic

hand posture” (See Figure 1). Because of the ambiguity of

hand postures, we cannot call every posture as its definite

name. For this reason, we also classify every hand gesture

into 3 postures (3 classes). Defined 3-posutres are

represented in Figure 1.

For recognition, the paper proposes a real time hand gesture recognition system using a camera. The proposed method is based on the random forest for hand detection [402], and the LDA is used for implementing the gesture patterns. We define three kinds of hand postures as “deterministic hand posture” (See Figure 1). Because of the ambiguity of hand postures, we cannot call every posture as its definite name. For this reason, we also classify every hand gesture into 3 postures (3 classes). Defined 3-postures are represented in Figure 1.



By the described method, we can read trajectory of hand and hand posture variation in real-time. They can be used as a signal to control electric devices and home appliances. Speed is important to the practical real-time interface. With the combination of random forest and the proposed NPD feature, we achieved more than 30 FPS. In validation and recognition stage, the paper used Linear Discriminant Analysis to classify hand samples and negative (background) samples. By using traditional methods, it takes twice the time and twice the computation. But here, it is solved without face rejection stage, by adding a face image in the negative image set. Face region is also considered as background in the validation stage. Distance between centroids obtained by k-means and test sample can be interpreted as degree of similarity. This validation process helps discarding face region but takes less than 1msec on average.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Number | Name | Author | Publication | Date |
| [7301342](https://ieeexplore.ieee.org/document/7301342) | Hand Gesture Recognition with 3D Convolutional Neural Networks | Pavlo Molchanov, Shalini Gupta, Kihwan Kim, and Jan Kautz | IEEE | 2015 |

**Description:**

Pre-processing: To normalize the temporal lengths of the gestures, each gesture sequence was resampled to 32 frames using nearest neighbour interpolation (NNI) by dropping or repeating frames. Spatial down sampling was also performed.

While the network architectures is quite simple, they perform thorough data augmentation during training. For data augmentation they use:

* reverse ordering of the frames and horizontal mirroring (computed offline, the remaining data augmentations are computed online during training);
* rotation, scaling and translation – spatially;
* spatial elastic deformation;
* fixed-pattern dropout, i.e. setting the same (but randomly selected) pixels across all frames to zero;
* random dropout;
* temporal scaling (of duration) and translation;
* Temporal elastic deformation (elastic deformation extended to the temporal domain.

**Method:**

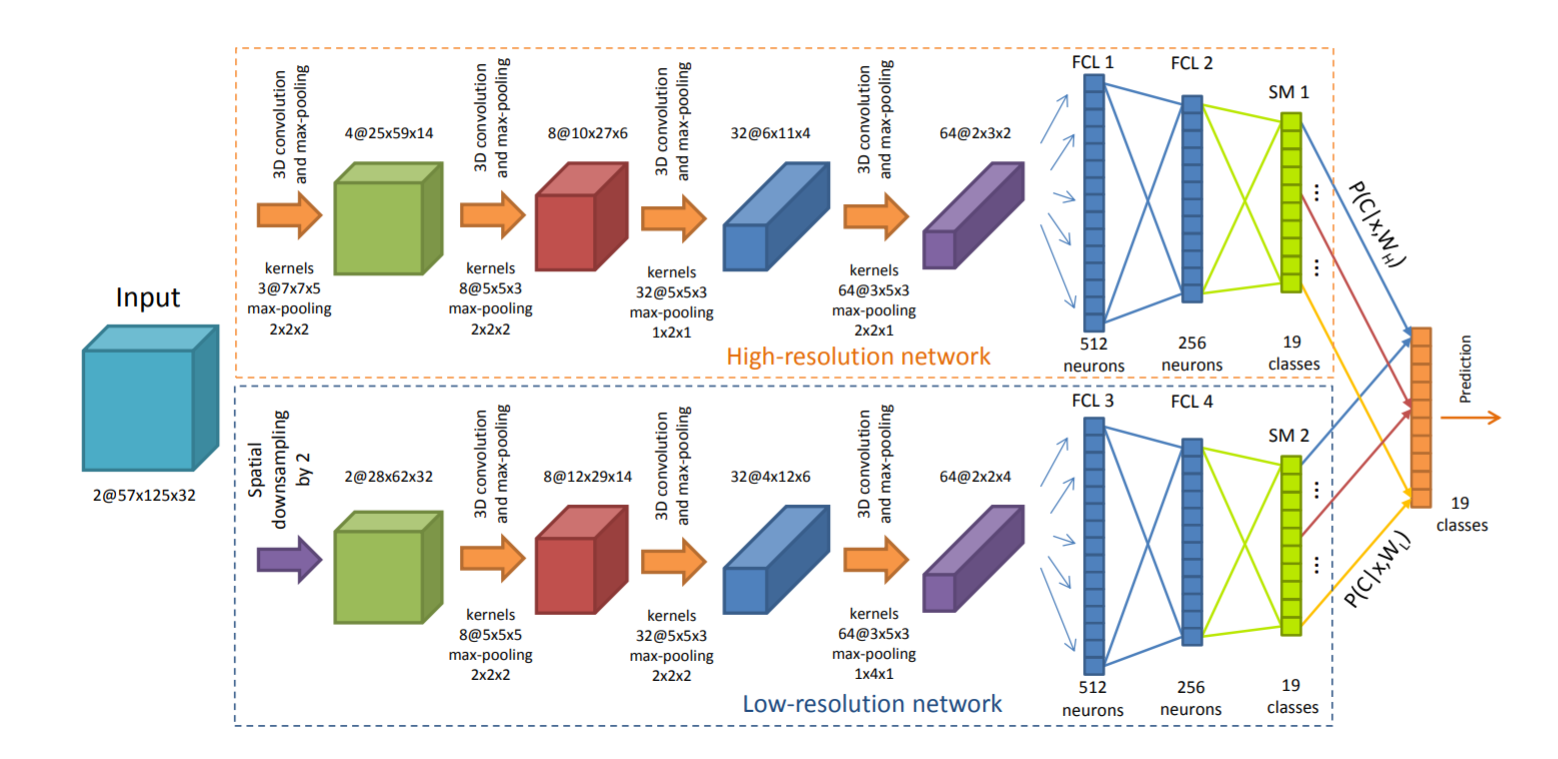
The paper proposes a 3D CNN for hand gesture recognition. The system consists of two networks, a high-resolution network and a low-resolution network – the predictions are multiplied during testing. The architecture is illustrated in Figure 3.

A convolutional neural network classifier consists of two sub-networks:

1. A high-resolution network(HRN)
2. A low-resolution network(LRN)

To avoid over fitting the paper performed offline and online spatio-temporal data augmentation (Since, the dataset contained only 885 images).

Fig 3



In experiments, they show that using depth information alone performs better than using intensity data only. Still the combination outperforms both. They also observe that including pre-computed gradients increases final performance.

For the challenging VIVA dataset, the proposed system achieves a classification rate of 77.5%.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Number | Name | Author | Publication | Date |
| 267872 | Real-Time Hand Gesture Recognition Using Finger Segmentation | Zhi-hua Chen, Jung-Tae Kim, Jianning Liang, Jing Zhang, Yu-Bo Yuan | The Scientific World Journal | June, 2014 |

**Description and Method Overview:**

In this work, a novel real-time method for hand gesture recognition is presented.

Firstly, a hand region is extracted from the background with the background subtraction method.

Then, the palm and fingers are segmented so as to detect and recognize the fingers.

Finally, a simple rule classifier is applied to predict the labels of hand gestures.



**Hand Detection and Fingers Recognition:**

It is easy and effective to detect the hand region from the original image using the background subtraction method. However, in some cases, there are other moving objects included in the result of background subtraction. The skin color (using the HSV model) can be used to discriminate the hand region from the other moving objects.

**Fingers and Palm Segmentation:**

(i) Palm Point: The palm point is defined as the center point of the palm. It is found by the method of distance transform. In the distance transform image, each pixel records the distance of it and the nearest boundary pixel.

(ii) Inner Circle of the Maximal Radius: When the palm point is found, it can draw a circle with the palm point as the center point inside the palm. The circle is called the inner circle because it is included inside the palm.

(iii) Wrist Points and Palm Mask. When the radius of the maximal inner circle is acquired, a larger circle the radius of which is 1.2 times of that of the maximal inner circle is produced. A larger circle instead of the maximal inner circle is used so as to yield a more accurate palm mask for segmentation.

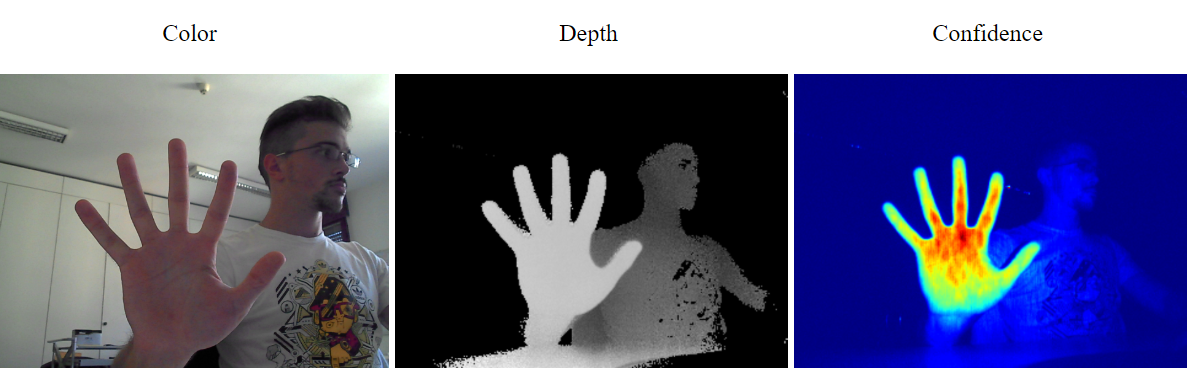
­(iv) Hand Rotation: The hand image rotates synchronously so as to make the hand gesture invariant to the rotation.

(v) Fingers and Palm Segmentation: The part of the hand that is covered by the palm mask is the palm, while the other parts of the hand are fingers.

**Proposed System**

**Scope of the project:**

Our hand gesture classification system, works primarily on the Creative Senz 3D Dataset([901] and [902]). The dataset contains several different static gestures acquired with the Creative Senz3D camera.

The dataset contains gestures performed by 4 different people, each performing 11 different gestures repeated 30 times each, for a total of 1320 samples. For each sample, colour, depth and confidence frames are available. 

Our scope is limited to detecting and classifying these 11 gestures only. For this classifier to be of practical use, we need to expand its scope not only in terms of the no of classes it classifies into, but also the number of images it trains upon.

This system will also give the proposed results if and only if it is tested on images using the Creative Senz 3D camera. There are many other camera and Infrared detectors like the Kinect and LEAPMOTION Camera, but the accuracy of the model will decrease when tested using those images.

**Dataset:**

**With citation**

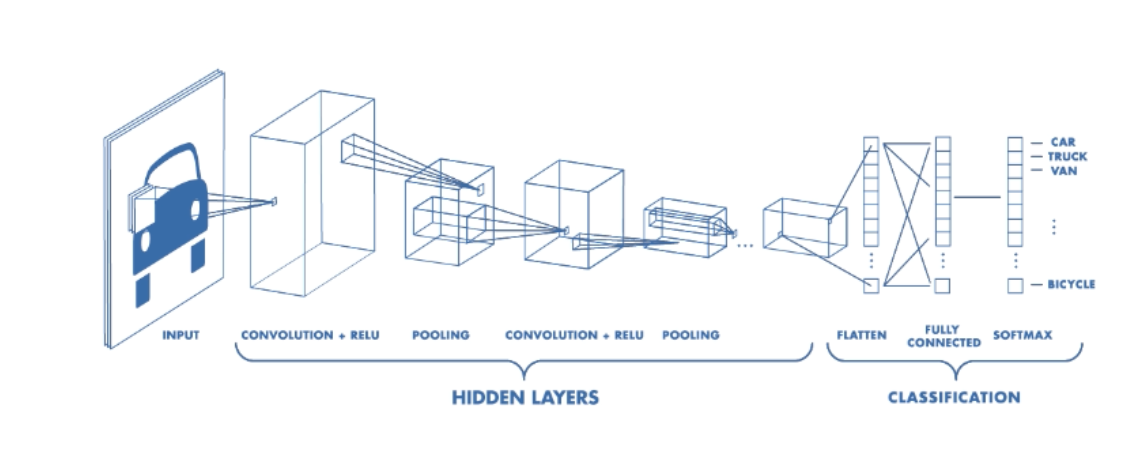
**Implementation Details**

**Whqt all you have tried**

**Convolutional Neural Network**

Convolutional neural networks are deep artificial neural networks that are used primarily to classify images (e.g. name what they see), cluster them by similarity (photo search), and perform object recognition within scenes.

The efficacy of convolutional nets (ConvNets or CNNs) in image recognition is one of the main reasons why the world has woken up to the efficacy of deep learning.

****

A Typical CNN Architecture

We used CNNs for feature extraction and multi-class classification.

The layers that were used were:

-**Fit:** Trains the model for a given number of epochs (iterations on a dataset):

fit(x=**None**, y=**None**, batch\_size=**None**, epochs=1, verbose=1, callbacks=**None**, validation\_split=0.0, validation\_data=**None**, shuffle=**True**, class\_weight=**None**, sample\_weight=**None**, initial\_epoch=0, steps\_per\_epoch=**None**, validation\_steps=**None**, validation\_freq=1)

-**Compile:** Compile defines the loss function, the optimizer and the metrics.

model.compile(optimizer='**rmsprop**', loss='categorical\_crossentropy', metrics=['accuracy'])

-**Conv2D:** 2D convolution layer (e.g. spatial convolution over images)

keras.layers.Conv2D(filters, kernel\_size, strides=(1, 1), padding='valid', data\_format=**None**, dilation\_rate=(1, 1), activation=**None**, use\_bias=**True**, kernel\_initializer='glorot\_uniform', bias\_initializer='zeros', kernel\_regularizer=**None**, bias\_regularizer=**None**, activity\_regularizer=**None**, kernel\_constraint=**None**, bias\_constraint=**None**)

**-MaxPooling2D:** Max pooling operation for spatial data.

keras.layers.MaxPooling2D(pool\_size=(2, 2), strides=**None**, padding='valid', data\_format=**None**)

**Dense**: Just your regular densely-connected NN layer.

keras.layers.Dense(units, activation=**None**, use\_bias=**True**, kernel\_initializer='glorot\_uniform', bias\_initializer='zeros', kernel\_regularizer=**None**, bias\_regularizer=**None**, activity\_regularizer=**None**, kernel\_constraint=**None**, bias\_constraint=**None**)

**-Flatten:** Flattens the input. Does not affect the batch size.

keras.layers.Flatten(data\_format=**None**)

**Confusion Matrix:**

[[ 8 0 0 0 0 0 0 0 0 0 0]

[ 0 12 0 0 0 0 1 0 0 0 0]

[ 0 0 11 0 0 0 0 0 0 0 0]

[ 0 0 0 11 0 0 0 0 0 0 0]

[ 0 0 0 0 11 0 0 0 0 0 0]

[ 0 0 0 0 0 15 0 0 0 0 0]

[ 0 0 0 0 0 0 14 0 0 0 1]

[ 0 0 0 0 0 0 0 13 0 0 1]

[ 1 0 0 0 0 0 0 0 11 1 0]

[ 0 0 0 0 0 0 0 0 0 11 0]

[ 0 0 0 0 0 0 0 0 0 0 10]]

**Classification report of the CNN classifier:**

|  |  |
| --- | --- |
| Precision (%) | Recall (%) |
| 100 | 88.889 |
| 92.308 | 100 |
| 100 | 100 |
| 100 | 100 |
| 100 | 100 |
| 100 | 100 |
| 93.333 | 93.333 |
| 92.857 | 100 |
| 84.615 | 100 |
| 100 | 100 |
| 100 | 91.667 |

The Overall Accuracy of this method is 96.212%

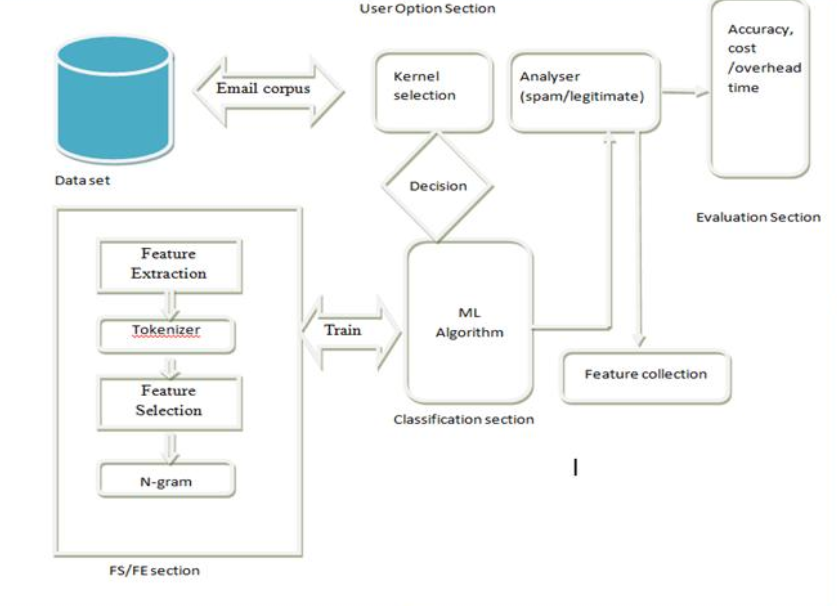
Kappa is 0.958

**Support Vector Machine (SVM)**

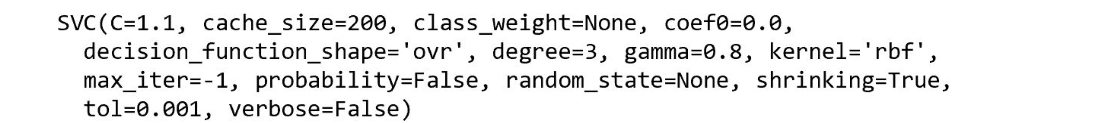
A Support Vector Machine (SVM) is a discriminative classifier formally defined by a separating hyper plane.

In other words, given labelled training data (supervised learning), the algorithm outputs an optimal hyper plane which categorizes new examples. In two-dimensional space this hyper plane is a line dividing a plane in two parts where in each class lay in either side.

A typical SVC architecture



The following is the function of the Support Vector Classifier:



**Confusion Matrix:**

[[10 0 0 0 0 0 0 0 0 0 0]

[ 0 5 0 0 0 0 0 5 0 0 0]

[ 0 0 3 0 0 0 0 10 0 0 0]

[ 0 0 0 2 0 0 0 9 0 0 0]

[ 0 0 0 0 7 0 0 3 0 0 0]

[ 0 0 0 0 0 6 0 6 0 0 0]

[ 0 0 0 0 0 0 4 13 0 0 0]

[ 0 0 0 0 0 0 0 9 0 0 0]

[ 0 0 0 0 1 0 0 8 6 0 0]

[ 0 0 0 0 0 0 0 1 0 9 0]

[ 0 0 0 0 0 0 0 10 0 0 5]]

**Classification report of the Support Vector Machine classifier:**

|  |  |
| --- | --- |
| Precision (%) | Recall (%) |
| 100 | 100 |
| 50 | 100 |
| 23.07 | 100 |
| 18.182 | 100 |
| 70 | 87.5 |
| 50 | 100 |
| 22.222 | 100 |
| 90 | 12.162 |
| 40 | 100 |
| 90 | 100 |
| 33.3333 | 71.429 |

The Overall Accuracy of this method is 49.243%

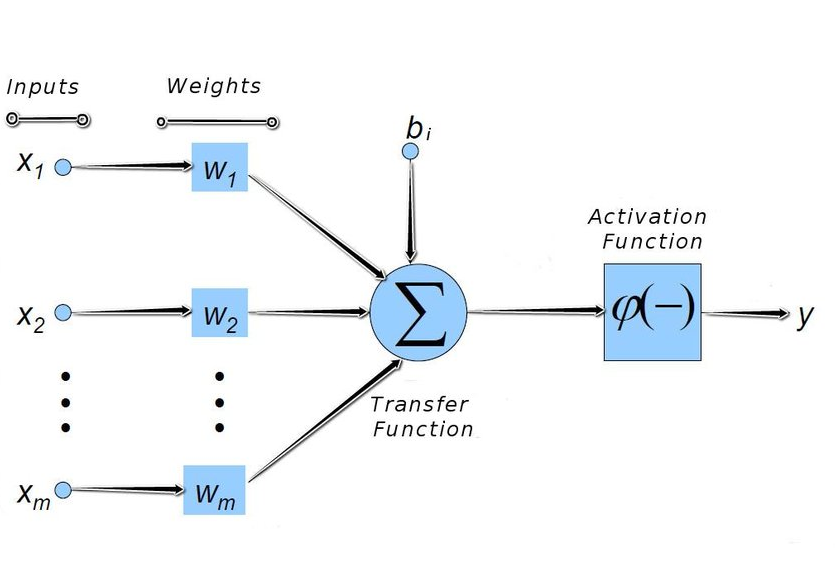
Kappa is 0.448

**K-Nearest Neighbours(KNN)**

K-Nearest Neighbors is one of the most basic yet essential classification algorithms in Machine Learning. It belongs to the supervised learning domain and finds intense application in pattern recognition, data mining and intrusion detection.

It is widely disposable in real-life scenarios since it is non-parametric, meaning, it does not make any underlying assumptions about the distribution of data (as opposed to other algorithms such as GMM, which assume a Gaussian distribution of the given data).

**KNeighborsClassifier**(n\_neighbors=5, weights=’uniform’, algorithm=’auto’, leaf\_size=30, p=2, metric=’minkowski’, metric\_params=None, n\_jobs=None, \*\*kwargs)



**Confusion Matrix of the KNN Classifier:**

[[11 0 0 1 0 0 0 0 1 0 0]

[ 0 15 1 0 0 0 0 0 0 0 0]

[ 0 0 11 0 0 0 0 0 0 0 0]

[ 0 0 0 14 1 0 0 0 0 0 0]

[ 0 0 0 0 10 0 0 0 0 0 0]

[ 0 0 0 0 0 9 0 0 0 0 0]

[ 0 0 0 0 0 1 8 1 0 0 0]

[ 0 0 0 0 0 0 0 15 0 0 0]

[ 0 0 0 0 0 0 0 0 12 0 0]

[ 0 0 0 0 0 0 0 0 0 7 0]

[ 0 0 1 0 0 0 0 0 1 0 12]]

**Classification report of the KNN classifier:**

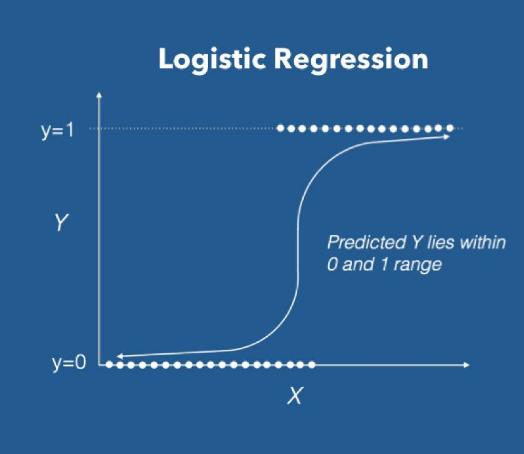
|  |  |
| --- | --- |
| Precision (%) | Recall (%) |
| 84.615 | 100 |
| 93.75 | 100 |
| 100 | 84.615 |
| 93.333 | 93.333 |
| 100 | 83.333 |
| 100 | 90 |
| 80 | 100 |
| 100 | 93.75 |
| 92.308 | 85.714 |
| 100 | 100 |
| 85.714 | 100 |

The Overall Accuracy of this method is 93.233%

Kappa is 0.925

**Logistic Regression:**

Logistic regression is used to describe data and to explain the relationship between one dependent binary variable and one or more nominal, ordinal, interval or ratio-level independent variables.



**Function:**

LogisticRegression(penalty=’l2’, dual=False, tol=0.0001, C=1.0, fit\_intercept=True, intercept\_scaling=1, class\_weight=None, random\_state=None, solver=’warn’, max\_iter=100, multi\_class=’warn’, verbose=0, warm\_start=False, n\_jobs=None)

**Confusion Matrix of the Logistic Regression Classifier:**

[[15 0 0 0 1 0 0 0 0 0 0]

[ 0 8 0 0 0 0 0 0 0 0 0]

[ 0 0 10 0 0 0 0 0 0 0 0]

[ 0 0 1 12 0 0 0 0 0 0 0]

[ 0 0 0 0 7 0 0 0 0 0 0]

[ 0 0 1 0 0 12 0 0 0 0 0]

[ 0 0 0 0 0 0 13 0 0 0 0]

[ 0 0 0 0 0 0 0 11 0 0 0]

[ 0 0 0 0 0 0 0 0 8 0 0]

[ 0 0 0 0 0 0 0 0 0 11 0]

[ 0 0 1 0 0 0 0 0 0 0 21]]

**Classification report of the Logistic Regression classifier:**

|  |  |
| --- | --- |
| Precision (%) | Recall (%) |
| 88.235 | 100 |
| 100 | 100 |
| 100 | 83.333 |
| 92.308 | 100 |
| 100 | 77.778 |
| 100 | 92.308 |
| 86.667 | 100 |
| 100 | 91.667 |
| 88.889 | 80 |
| 100 | 100 |
| 91.304 | 100 |

The Overall Accuracy of this method is **94.108%**

Kappa is 0.935

**Result Analysis**

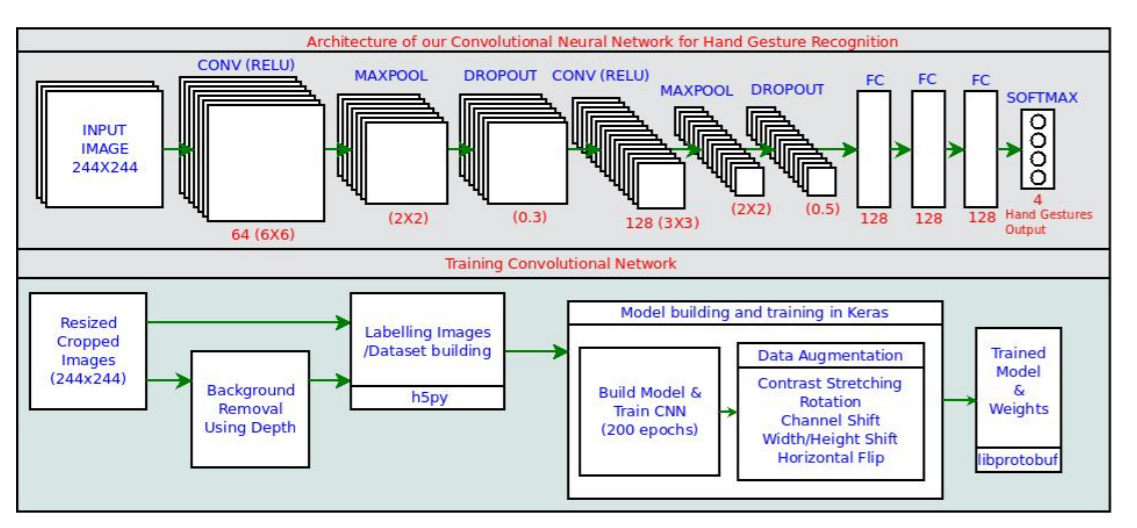
|  |  |
| --- | --- |
| Classifier Type | Accuracy (%) |
| Logistic Regression | 94.104 |
| K-Nearest Neighbours | 93.233 |
| Support Vector Machine | 49.243 |
| Convolution Neural Network | 96.212 |

From the above table, we can see that the Convolution Neural Network gives us the best result for this dataset. Its accuracy is close to that of K Nearest Neighbours and Logistic Regression Classifiers.

However, the Support Vector Machine Classifier is very far behind and does not do a very efficient job of accurately classifying the hand gestures.

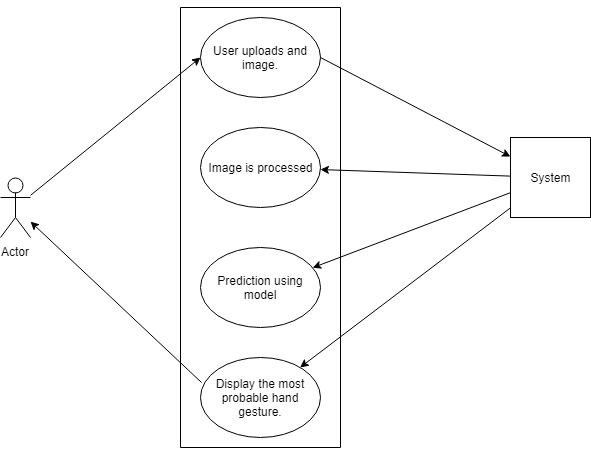
**System Design**

**Block Diagram:**

****

Architecture of our Convolutional Neural Network (top), and block diagram of training and building CNN model (bottom)

**Use Case Diagram:**



**Conclusion and Future Scope**

In our SVM classifier, we can also use GLCM (gray level co-occurrence matrix) along with our colour and histogram of gradients (HOG), this would increase the quality of our feature extraction.

Several authors have emphasized the importance of using many diverse training examples for CNNs [208, 217, 219]. They have proposed data augmentation strategies to prevent CNNs from over fitting when training with datasets containing limited diversity.

* Krizhevsky et al. [208] employed translation, horizontal flipping and RGB jittering of the training and testing images for classifying them into 1000 categories.
* Simonyan and Zisserman [219] employed similar spatial augmentation on each video frame to train CNNs for video-based human activity recognition.

However, these data augmentation methods were limited to spatial variations only.

* To add variations to video sequences containing dynamic motion, Pigou et al. [217] temporally translated the video frames in addition to applying spatial transformations. We have not used these techniques here.
* By integrating our system with a voice recognition system, such as Alexa we can embed it in ROBOTS and make an autonomous sign language system for breaking down the barriers of communication between mutes with the rest of the populace.
* We would also like to handle dynamic image processing and event handling accordingly. This would make air gesture technologies in mobile phones more accurate and further broaden the scope of (HCI) Human Computer Interaction.
* We are also planning on expanding and building upon this project for our Major Project in the next semester.

**References**

1. T. Mantecón, C.R. del Blanco, F. Jaureguizar, N. García, “Hand Gesture Recognition using Infrared Imagery Provided by Leap Motion Controller”, Int. Conf. on Advanced Concepts for Intelligent Vision Systems, ACIVS 2016, Lecce, Italy, pp. 47-57, 24-27 Oct. 2016. (doi: 10.1007/978-3-319-48680-2\_5)
2. [11] S. Mitra and T. Acharya. Gesture recognition: A survey. IEEE Systems, Man, and Cybernetics, 37:311–324, 2007. 1
3. [16] V. I. Pavlovic, R. Sharma, and T. S. Huang. Visual interpretation of hand gestures for human-computer interaction: A review. PAMI, 19:677–695, 1997. 1
4. [23] P. Trindade, J. Lobo, and J. Barreto. Hand gesture recognition using color and depth images enhanced with hand angular pose data. In IEEE Conf. on Multisensor Fusion and Integration for Intelligent Systems, pages 71–76, 2012. 1
5. [20] T. Starner, A. Pentland, and J. Weaver. Real-time american sign language recognition using desk and wearable computer based video. PAMI, 20(12):1371–1375, 1998. 1
6. [9] J. J. LaViola Jr. An introduction to 3D gestural interfaces. In SIGGRAPH Course, 2014. 1
7. [24] S. B. Wang, A. Quattoni, L. Morency, D. Demirdjian, and T. Darrell. Hidden conditional random fields for gesture recognition. In CVPR, pages 1521–1527, 2006. 1
8. [4] N. Dardas and N. D. Georganas. Real-time hand gesture detection and recognition using bag-of-features and support vector machine techniques. IEEE Transactions on Instrumentation and Measurement, 60(11):3592–3607, 2011. 1
9. [25] M. Zobl, R. Nieschulz, M. Geiger, M. Lang, and G. Rigoll. Gesture components for natural interaction with in-car devices. In Gesture-Based Communication in HumanComputer Interaction, pages 448–459. Springer, 2004. 1
10. [1] F. Althoff, R. Lindl, and L. Walchshausl. Robust multimodal ¨ hand-and head gesture recognition for controlling automotive infotainment systems. In VDI-Tagung: Der Fahrer im 21. Jahrhundert, 2005. 1
11. [15] F. Parada-Loira, E. Gonzalez-Agulla, and J. Alba-Castro. Hand gestures to control infotainment equipment in cars. In IEEE Intelligent Vehicles Symposium, pages 1–6, 2014. 1
12. [103] I. Oikonomidis, N. Kyriazis, and A. A. Argyros, “Efficient model-based 3D tracking of hand articulations using Kinect,” in Proc. Brit. Mach. Vis. Conf., 2011, pp. 101.1–101.11.
13. [104] C. Qian, X. Sun, Y. Wei, X. Tang, and J. Sun, “Realtime and robust hand tracking from depth,” in Proc. IEEE Conf. Comput. Vis. Pattern Recog., 2014, pp. 1–8.
14. [208] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In NIPS, pages 1097–1105. 2012. 1, 2, 4
15. [217] L. Pigou, S. Dieleman, P.-J. Kindermans, and B. Schrauwen. Sign language recognition using convolutional neural networks. In ECCVW, 2014. 1
16. [219] K. Simonyan and A. Zisserman. Two-stream convolutional networks for action recognition in videos. In NIPS, pages 568–576, 2014. 1, 4
17. [901] A. Memo, L. Minto, P. Zanuttigh,  "Exploiting Silhouette Descriptors and Synthetic Data for Hand Gesture Recognition", STAG: Smart Tools & Apps for Graphics, 2015
18. [902] A. Memo, P. Zanuttigh,  "Head-mounted gesture controlled interface for human-computer interaction", Multimedia Tools and Applications, 2017
19. https://www.researchgate.net/publication/286331414\_Real\_Time\_Hand\_Gesture\_Recognition\_Using\_Random\_Forest\_and\_Linear\_Discriminant\_Analysis
20. [402] 2. Fisher, R. A. The use of multiple measurements in taxonomic problems. Annals of eugenics 7, 2 (1936), 179-188.