International Journal of Recent Technology and Engineering (IJRTE)
ISSN: 2277-3878, Volume-8 Issue-5, January 2020

Hand Calculator System using Convolutional Neural Networks

Y. Jhansi, U. Harish

***Abstract:** Computer vision has great attention in recent years as it identifies and similarly processes images that human vision does and they provide suitable output. In computer vision, hand gesture recognition is one of the important and fundamental problems. The hand gesture recognition system has gained significant importance in the recent few years because of its manifoldness applications. This paper aims to give a new approach for vision-based, fast and real time hand gesture recognition, a new light that can be used in many HCI applications. The proposed algorithm first detects and segments the hand region and by using our innovative approach, it finds the fingers and classifies the gesture. The proposed algorithm is invariant to orientation, hand position or distance from the webcam. Based on this proposed algorithm we have progressively developed a gesture-based mathematical tool (calculator) as a practical application.*

***Keywords:** computer vision, Human-computer interaction (HCI), Gesture recognition, Convolutional neural network (CNN)*

I. INTRODUCTION:

HCI – Human-Computer Interaction is the new name, where the user can access and control the digital devices on the bases of the human body like an eye, head, and hand gestures which makes the physical world and digital world to come nearer. The aim of this is to make computers recognize human interactions and develop user-friendly Human-Computer Interfaces (HCI). Some steps related to this are - making a computer understand speech, facial expressions, and human gestures. Therefore, the idea of gesture recognition research is to form a system that can detect and understand specific human gestures automatically and employ them to convey information or to control devices [1]. Hand gestures can be movable or immovable wherein static gesture recognition the hand shape, size of palm, length, and width of fingers are important. Spatiotemporal information is required to track hand for movable hand gestures [2].

In any hand gesture recognition system, the initial step is Hand detection and tracking. For finding out human hand gestures, different methods have been proposed and implemented. Some methods make use of external devices like gloves and some make use of skin color to segment the hand features.

The hand gesture recognition study is classified into three categories. They used a sensor which is usually an acoustic or a magnetic attached to the glove for hand posture determination and an additional sensor for the position of the hand. This approach is referred to as "Glove based Analysis".

The second approach is "Vision-based Analysis" that users get information from their surroundings. Several cameras attached to this model to determine parameters corresponding for matching images of the hand, palm orientation, and joint angles to achieve hand gesture classification. Based on a three-dimensional hand skeleton model using 27 degrees of freedom was developed by Lee and Kunii. To diminish the model parameter space search they joined five key constraints centered on the human hand kinematics. To simplify the model matching, specially marked gloves were used [3]. The Third implementation is "Study of drawing gesture" use a stylus as an input device. These drawing analysis lead to the recognition of written text. Mechanical sensing work has used for hand gesture recognition at a massive level for direct and virtual environment manipulation. Mechanically sensing hand posture has many problems like electromagnetic noise, reliability, and accuracy.

Full American Sign Language recognition systems (words, phrases) incorporate data gloves. Using a joint angle and hand orientation coding technique data glove-based system could identify 34 of the 46 Japanese gestures (user dependent). To provide data for principal component and cluster analysis the test user made each of the 46 signs 10 times. For each of the alphabets, the user has generated a different test from five iterations, with each gesture separated in time but they suffer from a lack of training [4],[5].Classification of gestures is centered on wavelet transforms and temporal motion templates used a temporal illustration of gesture-based on alterations among successive images by building motion templates [6]. The preliminary idea is inspired by the point that in low-level resolution images a human observer can directly identify gestures deprived of great effort [7]. A Binary Motion Region, the BMR image is figured prominently to act as an index that typically defines the spatial distribution of motion for a given angle and a specified gesture in the gesture library.

Manuscript received on January 02, 2020.
Revised Manuscript received on January 15, 2020.
Manuscript published on January 30, 2020.

* Correspondence Author
Dr.Y.Jhansi*, CSE Department, GITAM Institute of technology, Visakhapatnam, India, jhansi.yellapu@gitam.edu.
U. Harish, IT Department, GITAM Institute of technology, Visakhapatnam, India, harish.fragger98@gmail.com

*Retrieval Number: D9106118419/2020©BEIESP
DOI:10.35940/ijrte.D9106.018520
Journal Website: www.ijrte.org*

Hand gesture recognition is also used to control TV interfaces [9]. For the classification of different gestures, the hand was separated into 21 dissimilar regions and the SVM classifier is used for training to model the joint distribution of these regions for different hand gestures. In [8] they improved the medical service through hand gesture recognition.

The HCI recognition system of the intelligent wheelchair includes five hand gestures and three compound states. Their system performs reliably in the environment of indoor and outdoor and in the condition of lighting change.

II. METHODOLOGY

In this section, we elaborate on our hierarchical architecture that enables the CNN models to be used in recognizing gestures as efficiently as possible. Here we give a brief description of the Architecture of the gesture recognizing the system, the pre-processing techniques that are applied on the image before prediction, The CNN architecture used to train the dataset and finally how the calculation is done.

A. *Architecture*

Fig 1 depicts the architecture of the classifier which is a deep CNN to classify the detected gesture and then with the gesture a virtual calculator is developed. This system depends totally on human-computer interaction, where the computer tries to interpret what the user is trying to show with the knowledge it has gained from the dataset provided. The process simply starts with a user showing a gesture in front of a video cam where each frame is potentially an input. A region of interest (a green square box) on screen is displayed where the user is supposed to place his/her hand so that the preprocessing techniques can be applied to that particular region of the frame to fit it into the CNN architecture**.** Once the pre-processing of a region of interest is done the image in the form of numpy array is passed to the CNN model for prediction. The output of the prediction is a numpy array where the array consists of only one element that is the one having the highest weight. The output is stored and used as a digit or an operator for calculation.

B. *Input Image Preprocessing*

A person who is sitting in front of the camera needs to make sure before starting the process that in order to predict accurately the background of region of interest must be plain and the hand region that is being shown should be only the palm as the model is trained on a dataset that has only palm region of the hand. When the video cam is turned on, a 300x300 green square box is displayed on the screen where the user is supposed to place his/her hand showing the gesture. Every frame from the video cam is an input image and every frame is preprocessed for prediction. There are only 2 preprocessing techniques that are being applied on a frame that is resized and reshaped. The Region of interest is of size 300x300 but our CNN model takes only 100x100 images as input. So using resize function we transform the image to a 100x100 size. Then reshape is done to the resized numpy array which makes sure the data is not changed. Fig 2 shows us how the clipping of a frame is done to create a region of interest.

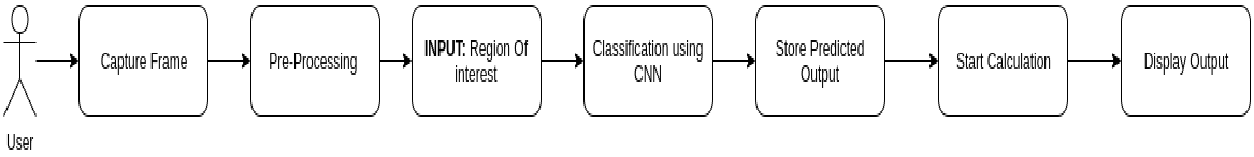


fig.1 Architecture of the proposed system

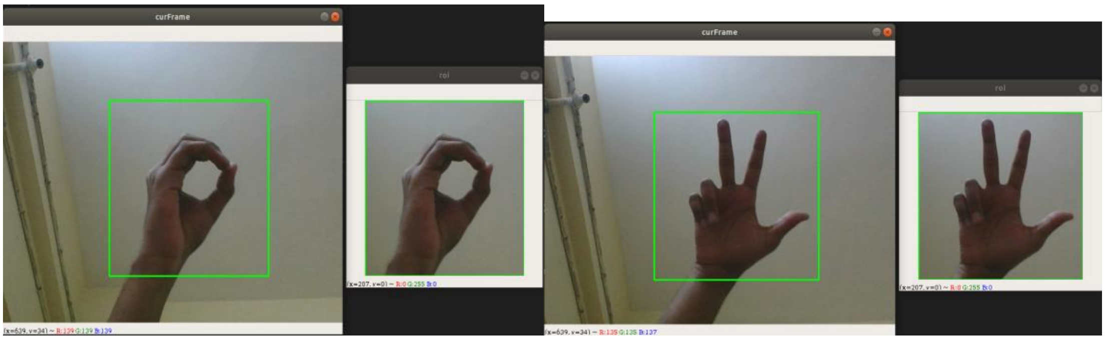


fig. 2 User showing gesture for number 0 and 3

C. *CNN Architecture*

Convolutional Neural Network (CNN) is a Deep Learning algorithm and an optimistic algorithm for image recognition, and image classifications. [14]. In present days, adequate applications are centered on CNN models, as for example are face recognition [10],[13], semantic parsing, text classification, question answering, information extraction[11],[12], Real-time parking management, traffic control [15],[16], video recognition [17], and Sketch-based image retrieval [18].

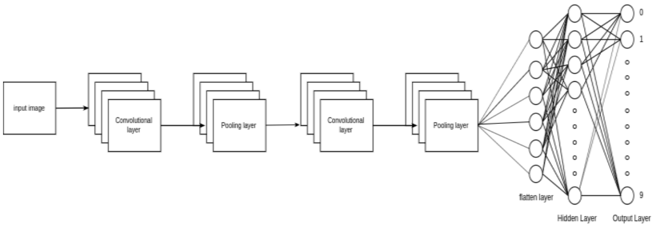


Fig 3 Convolutional Neural Network model

Human beings are liable for five senses to understand the actions around us. Out of these five senses, perception is by default the utmost used for the everyday tasks we perform. Deep learning methods have attained state-of-the-art results in computer vision problems such as object detection, image classification, and face recognition.

CNN consists of 4 parts that are the Convolutional layer, the pooling layer, the flattening layer and the Neural network layer as shown from fig 3. Different features from the individual image are extracted by the convolutional layer. Convolution layer preserves the correlation between pixels by learning image features through small squares of input data. Several mathematical combinations of 2 functions are done to produce a 3rd function. A convolution operation is done on the input image and the feature detector to produce a feature map. Several feature maps are created for an image and these feature maps together make a convolutional layer. The spatial size of representation is reduced using the pooling layer which in turn decreases the parameters and amount of computation process. There are several types of pooling ex. Max-pooling, Min-pooling, Mean pooling and average pooling in which we have used the max-pooling technique. Reducing the spatial dimensions (width x height) of the input volume for the next convolutional layer is the primary utility of the max-pooling technique. The input values are transformed into representative values using the pooling layer which takes a sliding window or a certain region that is moved in stride across the input. The method of transformation can be performed in any of the two ways by taking the average of the values or by taking the maximum value from the values observable in the window (called "max pooling"). Due to its better performance characteristics, max pooling is mostly preferred. Feature map from the Convolutional layer is the input matrix for pooling, Max-pooling is responsible for selecting the largest element from a 2d matrix (2x2, 3x3, etc.) which is placed in the feature map and moves right with a stride value. The output of the Pooling layer is another 2D matrix called pooled feature map. This process helps the model to recognize features even if they are tiled, zoomed, different colors, etc. The Pooled feature map is then transformed into a 1D array that is sent for further processing into the Neural network layers. The flattened layer takes the output from previous layers and classification is accomplished through the features that have been extracted by the convolutional layers. The ANN consists of the hidden layers which are responsible for performing computations on the weighted inputs and produce net input which is then applied with activation functions to produce the actual output. The nodes of the output layer are the classes that are given weights by the hidden layer after the complete process of the ANN each class in the output layer will have a value and the class with the highest value is the prediction output.

In this paper, we have used an architecture that consists of 2 convolutional layers with 32 filter layers each. The 1st convolutional layer is max pooled with a pooling size of 2x2 matrix. The output of max-pooling is fed to another convolutional layer with 32 filters and again the feature maps are pooled using a 2x2 matrix. Flattening is done to the pooled matrix and then connected to the Artificial neural network. The model has a dense layer with 128 hidden neurons and the output layer consists of 10 classes ie. 0 – 9.

An activation function called a rectified linear unit also known as Relu is applied on the convolutional layers and the hidden layer. Relu is a rectifier which is used to increase the non-linearity in our images. Fig 4 shows us the relu function graph.

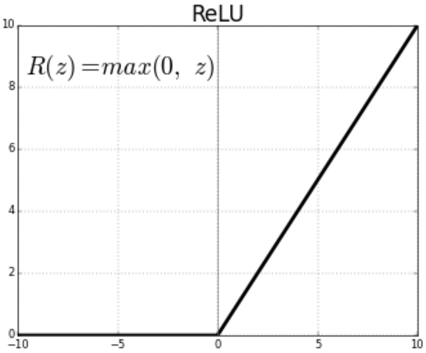
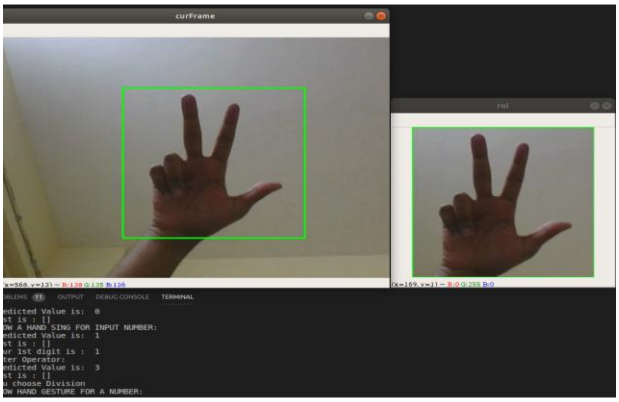


fig 4. Relu Activation function

D. *Calculation*

To perform calculations, the user is supposed to place his/her hand in the region of interest and press key 's' to start the process. As soon as the key is pressed the tool starts to predict every frame. For 100 frames predicted values have appended a list and from those 100 predicted values, the value that occurs the most is considered as the final prediction. The calculator has 4 operators that are Addition, subtraction, multiplication, and division. For choosing the operator a gesture has been assigned a number ie. Gesture 1 represents addition, 2 stands for subtraction, 3 means multiplication and 4 used for division an example was depicted in fig 5. The CNN model can predict in a total of 10 classes ie 0-9.


curFrame
roi
(x=568, y=13) ~ R:138 G:135 B:126
(x=169, y=1) ~ R:0 G:255 B:0
OBLEMS 11
OUTPUT
DEBUG CONSOLE
TERMINAL

```
edicted Value is:  0
st is : []
OW A HAND SING FOR INPUT NUMBER:
edicted Value is:  1
st is : []
ur 1st digit is :  1
ter Operator:
edicted Value is:  3
st is : []
u choose Division
OW HAND GESTURE FOR A NUMBER:
```

fig 5 Showing gesture for 3 for choosing division operation

III. RESULTS AND DISCUSSION

In this section we'll be discussing the dataset we used to train the model, what results we obtained after training which includes validation accuracy and loss and some experimental results.

A. *Dataset*

The concept of convolutional neural networks and artificial neural networks were there from the 1980s but were not very much useful because of the lack of data. In earlier days' registers were maintained to keep different all sorts of records and information. Very less digital data was available which is the key aspect for CNN to work. In the early 2000's data started to take digital form, everything started to get automated and computerized.

The dataset that we have used to train our model is of hand Signs from 0-9. The 'Kaggle' Sign Language digits' dataset [18] of hand gestures has been used to evaluate the framework. The Sign Language digits' database has 10 classes of digits. There are in total of 1534 training images and 548 test images. Every class of hand sign has around 150 images for training and 50 images for testing. Fig 6 is a sample dataset of sign language digits.

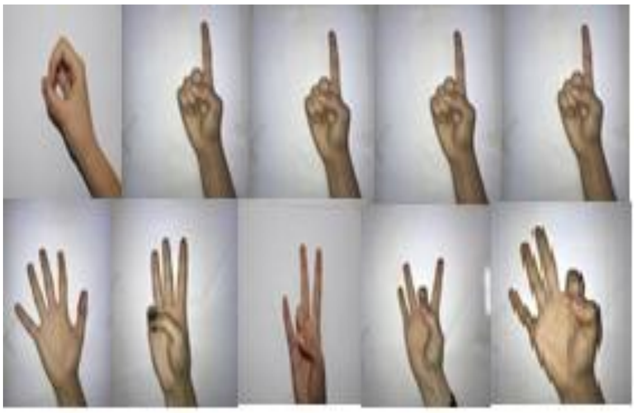
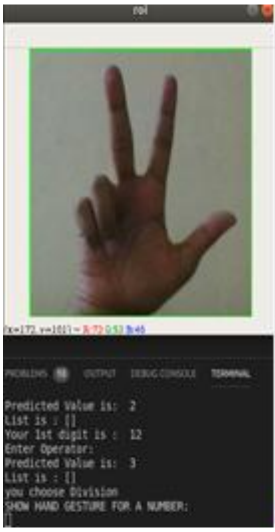


fig. 6 Sample Sign Language digits' dataset

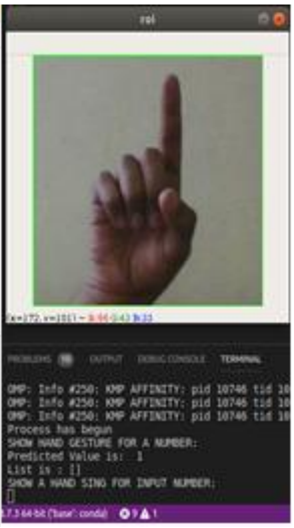
B. *Experimental Results*

During the calculation process, the user is asked to enter a digit by displaying a hand sign, as fig 7 shows the user has input 1 as the 1st digit. similarly, the 2nd is also taken as input in the same way. For choosing an operator the user is supposed to show a hand sign between 0-4 where 0 is for addition, 1 is for subtraction, 3 for division and 4 for multiplication. Fig 7(c). shows us that the user chooses the division operation.

In Figure 7 (a) the user is showing hand sign 1 which is taken as the first digit. Similarly, in figure 7(b) the user is displaying a hand sign for number 2 which is taken as the 2nd input for our 2-digit number. To choose an operator user is showing hand sign 3 to select the division in figure 7(c). To generate the 2nd digit the user shows hand sign 8 and 9 in figure 7(d) and figure 7(e) respectively.

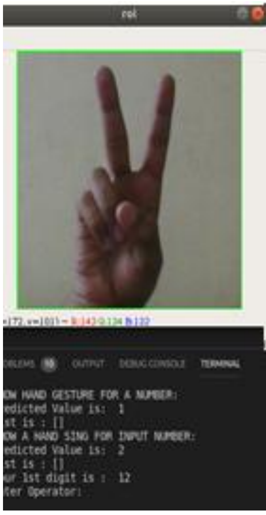


```
Predicted Value is:  2
List is : []
Your 1st digit is :  12
Enter Operator:
Predicted Value is:  3
List is : []
you choose Division
SHOW HAND GESTURE FOR A NUMBER:
[]
```

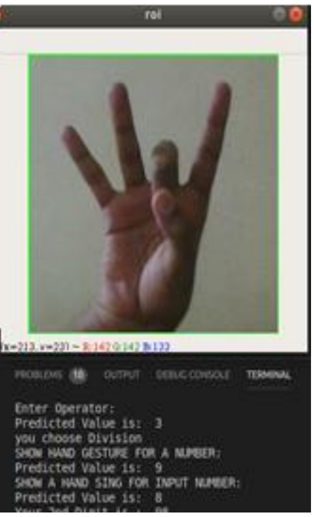


```
OMP: Info #250: KMP AFFINITY: pid 10746 tid 10
OMP: Info #250: KMP AFFINITY: pid 10746 tid 10
OMP: Info #250: KMP AFFINITY: pid 10746 tid 10
Process has begun
SHOW HAND GESTURE FOR A NUMBER:
Predicted Value is:  1
List is : []
SHOW A HAND SING FOR INPUT NUMBER:
[]
```

fig 7 (a) Hand sign 1 for 1st digit fig 7 (b) hand sign for digit 2

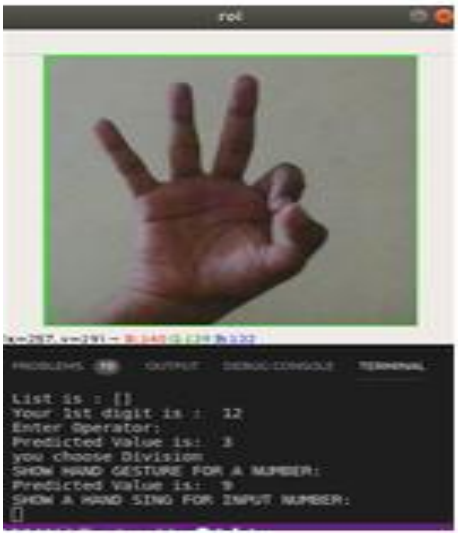


```
OW HAND GESTURE FOR A NUMBER:
edicted Value is:  1
st is : []
OW A HAND SING FOR INPUT NUMBER:
edicted Value is:  2
st is : []
ur 1st digit is :  12
ter Operator:
```



```
Enter Operator:
Predicted Value is:  3
you choose Division
SHOW HAND GESTURE FOR A NUMBER:
Predicted Value is:  9
SHOW A HAND SING FOR INPUT NUMBER:
Predicted Value is:  8
Your 2nd Digit is :  98
```

fig 7 (c) hand sign for operator fig 7 (d) hand sign for digit 9



```
List is : []
Your 1st digit is :  12
Enter Operator:
Predicted Value is:  3
you choose Division
SHOW HAND GESTURE FOR A NUMBER:
Predicted Value is:  9
SHOW A HAND SING FOR INPUT NUMBER:
[]
```

fig 7 (e) hand sign for digit 8

Training of the dataset which consists of 10 total classes with 1534 training images and 548 testing images have obtained a validation accuracy of 95%. Table- I shows us each epoch's validation accuracy and Loss.

Table- I: validation accuracy and Loss of different epochs.

	Validation Loss	Validation Accuracy
1	0.1990	0.9478
2	0.2120	0.9521
3	0.2244	0.9519
4	0.2458	0.9516
5	0.2700	0.9529
6	0.2290	0.9551
7	0.2270	0.9559
8	0.2321	0.9491
9	0.2643	0.9504
10	0.2858	0.9525
11	0.2990	0.9534
12	0.2751	0.9566
13	0.2162	0.9569
14	0.2666	0.9553
15	0.3425	0.9503

IV. CONCLUSION:

This paper proposes an approach to use Convolutional Neural Networks for classification of hand signs and use the predicted value to build a calculator that can calculate 4 arithmetic operators ie. Addition, subtraction, division, and multiplication. Using the power of Convolutional neural networks for classification we have applied preprocessing techniques on each image such as resize and reshape to fit each image to our Neural network. The training and testing dataset consists of 10 classes each with 1534 and 540 images respectively. The validation accuracy obtained was 95%. The convolutional neural network consists of 2 Convolutional layers attached with a pooling layer each, 1 flattening layer, 1 dense layer, and an output layer. Based on the predicted outputs of images the calculator programs start the calculating process where the user's hand signs taken as input numbers and operator is selected based on 4 hand sign those are from 0-4 that correspond to addition, subtraction, multiplication, and division.

REFERENCES:

1. M. K. Bhuyan., et al. “A novel set of features for continuous hand gesture recognition,” Journal on Multimodal User Interfaces., vol. 8, no. 4, pp. 333–343, 2014.
2. El-Baz AH., et al. “An efficient algorithm for 3D hand gesture recognition using combined neural classifiers”. Neural Computer Applications 22(7–8): pp. 1477–1484,2013
3. Kay M. Stanney “HANDBOOK OF VIRTUAL ENVIRONMENTS Design, Implementation, and Applications, Gesture Recognition” Chapter #10
4. W Chung., et al. "A real-time hand gesture recognition based on Haar wavelet representation". IEEE Conf. Robot.Biomimetics. 2009: pp.336-341.
5. L. Dipietro., et al. “A survey of glove-based systems and their applications,” IEEE Trans. Systems, vol. 38, no. 4, pp. 461–482, Jul. 2008.
6. Sanjay Kumar., et al. “Visual Hand Gestures Classification using Temporal Motion Templates and Wavelet Transforms”, 10th International Multimedia Modelling Conference, IEEE Computer Society,2004
7. James William Davis, “Appearance-Based Motion Recognition of Human Actions”, MIT Media Lab, TR 387,1996
8. J. Zeng., et al. "A natural hand gesture system for intelligent human-computer interaction and medical assistance," 3rd Global Congress on Intelligent Systems, pp.382–385, November 2012.
9. A. Shimada., et al. "Hand gesture-based TV control system—towards both user—&machine friendly gesture applications," 19th Korea Japan Joint Workshop on Frontiers of Computer Vision , pp.121–126, Feb 2013.
10. S. S. Farfade, et al. "Multi-view face detection using deep convolutional neural networks." 5th ACM on International Conference on Multimedia Retrieval, pp. 643–650, New York, USA, 2015.
11. X. Hu., et al. "Mandarin speech recognition using a convolution neural network with augmented tone features." Chinese Spoken Language Processing, International Symposium pp. 15–18, Sept 2014.
12. J. Huang., et al. “An analysis of convolutional neural networks for speech recognition.” Acoustics, Speech and Signal Processing IEEE International Conference on, pp. 4989–4993, April 2015.
13. M. Khalil-Hani., et al. "A convolutional neural network approach for face verification." High-Performance Computing Simulation, pp. 707–714, July 2014.
14. A. Krizhevsky., et al. “ImageNet classification with deep convolutional neural networks.” Advances in Neural Information Processing Systems 25, pp. 1097–1105. Curran Associates, 2012.
15. M. M. Lau., et al. "Malaysia traffic sign recognition with a convolutional neural network." IEEE International Conference on Digital Signal Processing, pp.1006–1010, July 2015.
16. C. Wang., et al. “Deep people counting in extremely dense crowds.” 23rd ACM International Conference on Multimedia, pp. 1299–1302. ACM, 2015.
17. R. Wang., et al. "A pedestrian and vehicle rapid identification model based on convolutional neural network." 7th International Conference on Internet Multimedia Computing and Service, pp. 32:1–32:4, New York, USA, 2015. ACM.
18. Jhansi Y, Reddy E S.," A new approach for Sketch-Based Image Retrieval using FISH SWARM optimization with the aid of Optimal score level fusion," In Indian Journal of Science and Technology, Vol 9, issue 41, November 2016, pp. 1-9
19. [online] Available: https://www.kaggle.com/ardamavi/sign-language-digits dataset/version/2#.

AUTHORS PROFILE

Dr.Y.Jhansi is currently working as an Assistant Professor in the Computer science engineering Department, GITAM Institute of Technology, Visakhapatnam. She was awarded Ph.D. From Acharya Nagarjuna University in the year 2019. Her research interests are focused on image processing and computer vision. She has published 07 publications in reputable international journals. She has a total of 13years of teaching experience.

Harish is currently pursuing his final year B. Tech in Information Technology from GITAM Institute of Technology, Visakhapatnam. His area of interest is towards computer vision.