International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 07 Issue: 05 | May 2020 www.irjet.net p-ISSN: 2395-0072

# Numbers and Alphabets Recognition in the Form of Gestures through Deep Learning

**Mr.D.Bikshalu[1], Dr.K.Kranthi Kumar[2], Bejjanki Anuhya[3], Jabba Aswanth [4], Jhade Kalyan Sreekar [5], Kallem Pavan [6]**

*[1, 2]Associate Professor, Department of Information Technology, Sreenidhi Institute of Science and Technology, Telangana, India*
*[3,4,5,6]B. Tech Student, Department of Information Technology, Sreenidhi Institute of Science and Technology, Telangana, India*

---

**Abstract –** *A Human focused human-PC collaboration innovation will in general supplant the conventional PC focused innovation with its developing materialness to a wide assortment of utilizations. Going about as a method for most normal correspondence among human and machine, vision based hand signal acknowledgment is turning into the quest for human-PC communication. General vision based hand signal acknowledgment by and large comprises of test catching, picture reprocessing, include extraction and arrangement. Among these systems include extraction expects to identify and remove includes that can be utilized to decide the significance of a given hand motion. The separated highlights ought to have the capacity to portray motion extraordinarily and be strong to the move and turn of hand motion so as to accomplish a dependable acknowledgment. In this task, we convert the picture to a grey scale picture and after that dim scale picture or a highly contrasting picture to double picture by applying edge esteem. Where dark speaks to 0 and white speaks to 1. We utilized Pytorch and Fastai, Resnet-50 in Deep Learning to prepare the dataset. This strategy is straightforward, proficient and free from signal bearing and position*

***Key Words***: **Pytorch, Grey Scale, Fastai, Deep Learning, Resnet-50.**

## 1. INTRODUCTION

Signals are the development of anyone part used to pass on the significant data. Correspondence through motions has been broadly utilized by people to express their musings and sentiments. Signals acknowledgment alludes to the way toward recognizing motions performed by human with the goal that machine can play out the relating activity. A large portion of the collaboration with PC frameworks has been done using understood cooperation gadgets.

Correspondence should be as generic as possible, and should copy unilateral interactions sensibly. For example, pointing at something to select it or using the hand action of having a thing to move it from one place to the following can be seen as natural. "People in need of a hearing aid use moves to offer every day. It is charming to direct these flags along these lines towards a written dedication to applications. It would later encourage the mapping of these movements to a general commitment for intelligent applications, as would the development of ongoing frameworks of understanding between spoken and gestural communication" [1]. Vision based procedures [2] utilizes one or cameras to catch the hand pictures. Different kind of cameras utilized for catching picture can be stereo cameras, monocular cameras, fish eye cameras, time of flight cameras, infrared cameras, and so on.

So as to accomplish this few detecting innovations can be connected, including information gloves, vision based frameworks, and electromyogram (EMG) sensors. On 2013 Thalmic Labs propelled the Myo armband, which incorporates on a remote gadget inertial and EMG sensors. The Myo armband is light and simple to wear, just as moderate and delivered around the world. Along these lines the Myo armband is viewed as available and important outcomes on this gadget can possibly achieve a wide use and effect on a specific application area. Hand signals are a characteristic type of correspondence among individuals, and as opposed to gadgets, for example, joysticks or consoles they stay instinctive, touchless, non-intrusive methods for human-PC, or human robot, correspondence. In any case, regardless of many years of research in the area, the hand worked gadgets are not normally utilized in our day by day lives since the as of now created frameworks offer sensible dependability just in a controlled lab condition. The number and noteworthiness of limitations forced on such frameworks is as yet significant. "These days, new gadgets, for example, time-of-flight (ToF). ToF camera and a RGB camera to permit hands covering with the face, be that as it may, adjustment is essential for a mapping from profundity information to RGB picture." [3] "Uses various Kinects to follow the hand and a HD shading sensor to fragment the hand for stable motion acknowledgment. Cameras or sensor Kinect

that offer new conceivable outcomes have been created. They give 3D data about watched scenes yet because of their particular nature they require new calculations and preparing plans. As per Seo Yul Kim the waveforms can be utilized for motion acknowledgment with AI, for example, CNN" [4]. "The use of Electromyography (EMG) (Saponas, etal., 2008" [5]; "Wolfetal., 2013" [6] are additionally especially valuable for hand motion acknowledgment. So we developed a system which will work better than these currently existing technologies. Our system gave us an accuracy of 96.4%. We applied deep learning into gestures recognition and we got good results. We trained the model with our own dataset containing 2900 images of 100 images for each gesture.

## 2. RELATED WORK

The current works in the writing on the point of Sign Language Recognition are principally isolated in three classes, in view of how the information is procured from the clients. An extraordinary number of works depend on uncommon gloves containing sensors equipped for estimating the finger positions. "Most of the rest of the works use cameras and Computer Vision methods for obtaining and handling pictures of the performed signals. The primary issue with this methodology is that the outcomes depend extraordinarily on ecological conditions, for example, enlightenment and shades of the articles in the scene. There are similar works which depend on the utilization of electromyogram (EMG)" [7] signals which measure the electrical movement of muscles, to give highlights to characterization. "Measures all the three kinds of withdrawals. This methodology has as of late begun being investigated, and has focal points over the two options. EMG sensors are not badly arranged to wear, in contrast to sensor gloves, and are not touchy to ecological conditions, in opposition to cameras. We contemplated profound learning exercises gave in the fastai.com site" [8]. What's more, we saw how the convolutional neural system work. We have demonstrated quickly how convolutional neural system work.

### 2.1 Convolutional Neural Networks

In this paper, the precedent that I will take is identified with Computer Vision. Notwithstanding, the fundamental idea continues as before and can be connected to some other use-cases. Every neuron gets a few sources of info, takes a weighted entirety over them, go it through an initiation work and reacts with a yield.

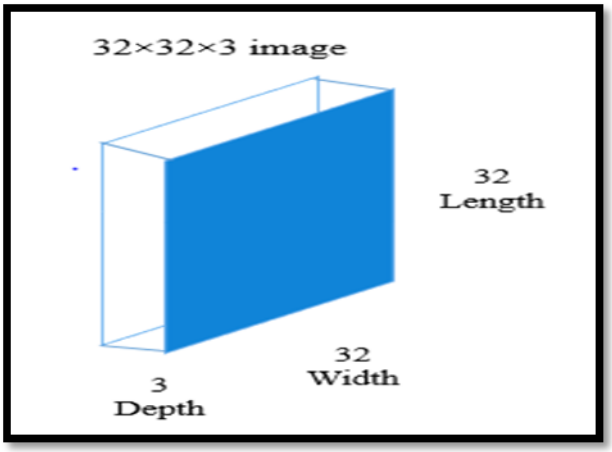


**Fig 1.** Example of a RGB image

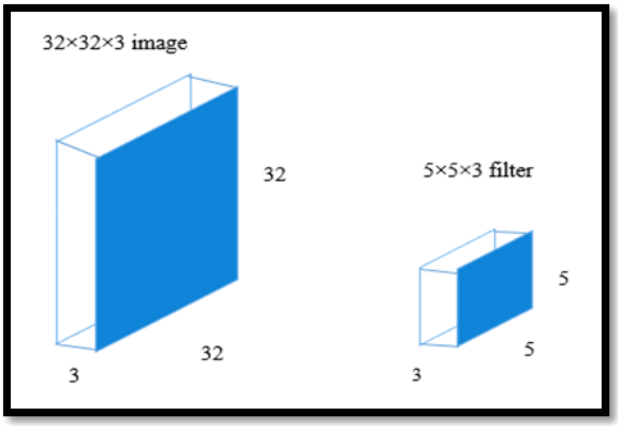


**Fig 2.** Convolving an image with a filter

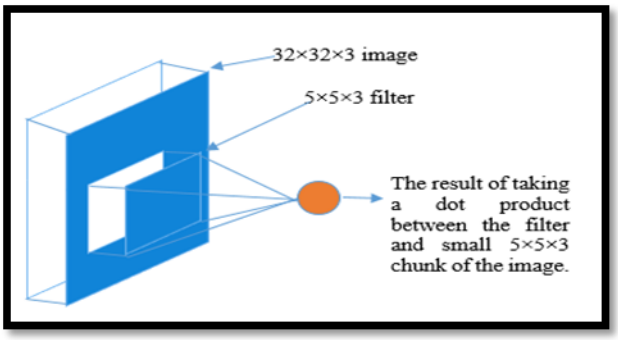


**Fig 3.** Acquiring result of dot product

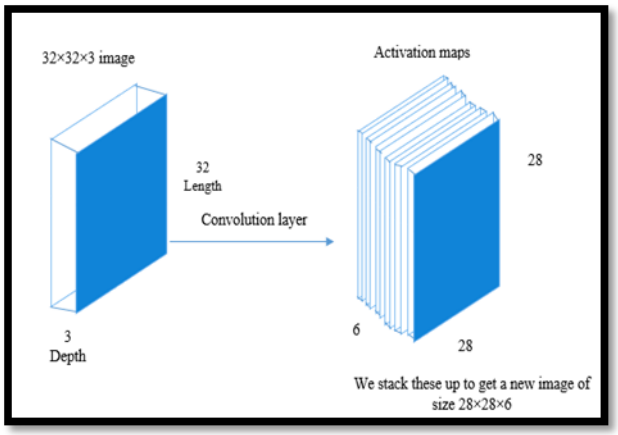


**Fig 4.** Convolution layer

The convolution layer is the primary structure square of a convolutional neural system. The convolution layer will be having a lot of free channels (6 in the precedent appeared). Each channel is freely convolved with the picture and we end up with 6 include maps of shape 28*28*1.

Investigate the channels in the absolute first layer (these are our 5*5*3 channels). Through back spread, they have tuned themselves to move toward becoming masses of shaded pieces and edges. As we go further to other convolution layers, the channels are doing speck items to the contribution of the past convolution layers. In this way, they are taking the littler shaded pieces or edges and making bigger pieces out of the All these channels are instated arbitrarily and turned into our parameters which will be found out by the system consequently. I will demonstrate to you a case of a prepared system. Investigate picture 4 and envision the 28*28*1 matrix as a framework of 28*28 neurons. For a specific component map (the yield got on convolving

the picture with a specific channel is known as an element map), every neuron is associated just to a little lump of the info picture and every one of the neurons have a similar association loads. So again returning to the contrasts among CNN and a neural system.

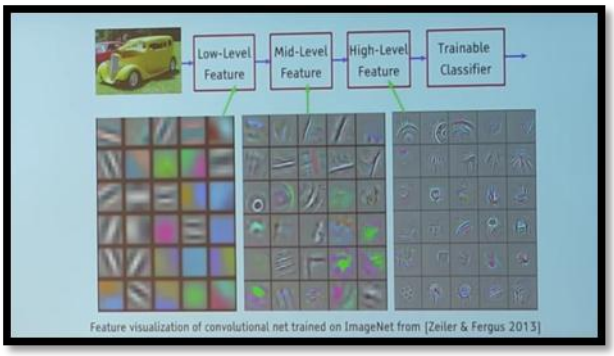


**Fig 5.** Filters in a trained network

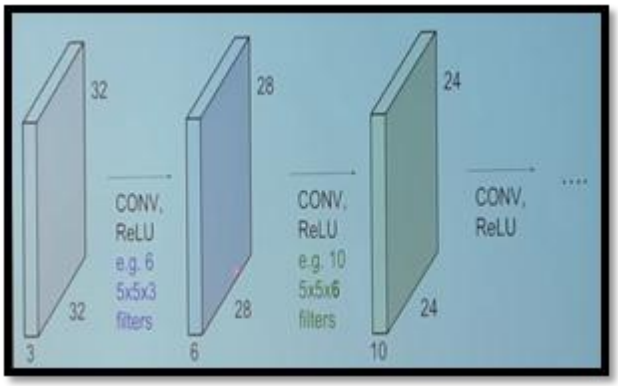


**Fig 6.** Convolution layers in sequence

## Typical Architecture of a CNN

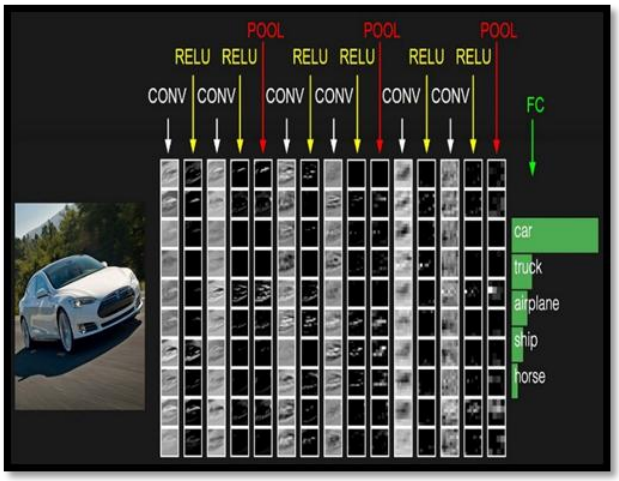


**Fig 7.** Typical architecture of CNN

### 2.2 Dataset Collection

We collected 2900 images through an Opencv python program. There are 100 images are present in our database for each gesture. We wrote the program in Spyder IDE which was there in Anaconda. We wrote a program in such a way that it will capture the gestures given by the user. It will convert those images into gray images then those gray images are converted into binary images by applying threshold value. The images in the database are of size 214/240 size. These gestures are taken by the opencv python program. Which will capture the images frame by frame. We used four people for showing the gestures. And also took the images at different distances from the web camera of laptop. The gestures were taken at various inclinations. The dataset contains small gesture images as well as large gesture images. This is because we have taken the images at various distances from the web camera. We took only 22 Alphabets. And 4 Alphabets are left because they have the similar structure like numbers gestures. The left over characters are D, F, Q, V, and W, X, Z. The reason why we do not considered Q, V, W, Z because the gesture image Q looks same like the gesture image of G. The gesture image of V looks same like that of gesture image of 2. The gesture image of W looks same like that of gesture image of 3. The gesture image of Z looks same like that of gesture image of 1.


**Fig 8.** Dataset

### 3. IMPLEMENTATION

In addition to the exposure to a dense and rich variety of natural images, one important property of the primary brain is its hierarchical organization in layers of increasing processing complexity, this is an architecture that has inspired Convolutional Neural Networks.

The collected data will undergo for transformation. Then that data will be sent to the model for training. Then after that we will place the model at the location of server. Actually on the monitor we will be see an interface which will ask us to put your hand in a box. The hand gesture will be captured and converted into a gray colored image. That gray colored image is converted into a binary image by applying some threshold value. This binary image will be sent to the server. The server based on the trained model it will classify the class to which the gesture belongs to. Then response will be shown on the user interface.

### System Architecture

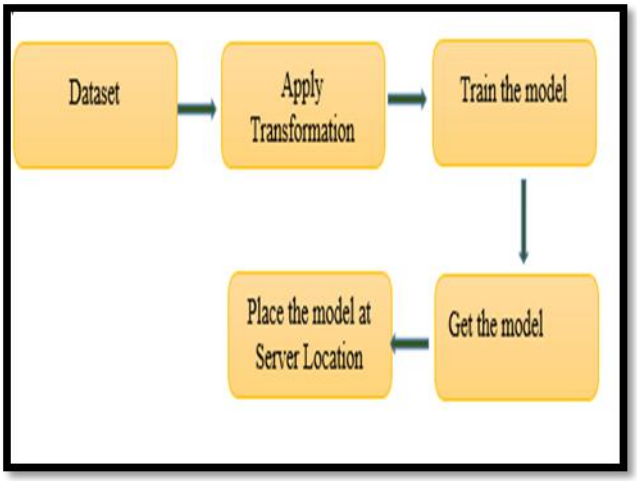


**Fig 10.** Training the Data

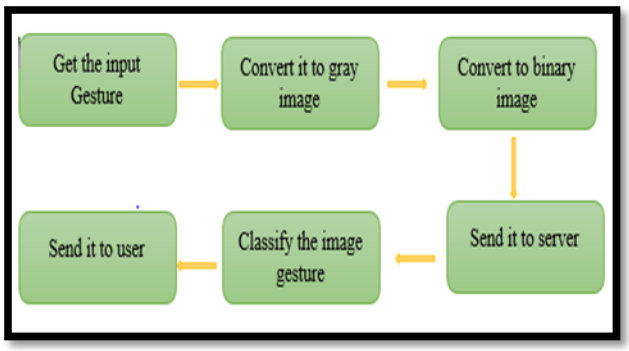


**Fig 11.** Testing in real time

In order to implement our project we need Jupyter notebook which is the package in Anaconda Software. Let's build up from the basics, what is a Jupyter Notebook? Well, you are reading one. It is a document made of cells. You can write like I am writing now (markdown cells) or you can perform calculations in Python (code cells) and run them like this:

| |
|---|
| IN [] : 1+1 |
| OUT [] : 2 |

This combination of prose and code makes Jupyter Notebook ideal for experimentation: we can see the rationale for each experiment, the code and the results in one comprehensive document. In fast.ai, each lesson is documented in a notebook and you can later use that notebook to experiment yourself.

### Getting Image Files from Folders

To get the images from the folders we use the function written below.

```
data = (ImageList.from_folder ('hands')
      .split_by_rand_pct()
      .label_from_folder()
      .transform(get_transforms(), size=56, resize_method = ResizeMethod.SQUISH)
      .databunch(bs=16))
data.normalize(imagenet_stats)
```

ImageList.from_folder will return a data object. Anything that you model with will be a data object in Fastai. This Data object generally contains 2 or 3 datasets-it includes your training data, validation data, and other optional test data. It includes the images and labels for each of those, the texts and labels, or the tabular data and labels, or so on. And that all sits there (i.e. data)

in this one location. ResizeMethod. SQUISH) this technique is used to reduce the image size without losing the details. In a little bit, something we'll hear more about is normalization. In almost all machine learning ventures, however, we usually have to make all our data to the same "scale"-they are basically around the same mean and standard deviation. So there will be a normalize function that we can use to normalize our data bunch in that way to the same size. data.show_batch (rows=3, figsize= (15, 11)). This function is used to show the data and also to check the data. It will be very useful to check whether any part of images are cropped or some unwanted noise has deposited on the images. Or the gestures that we gave to the model or looking same like that of our gestures or not. And also the images got over fitted or not.

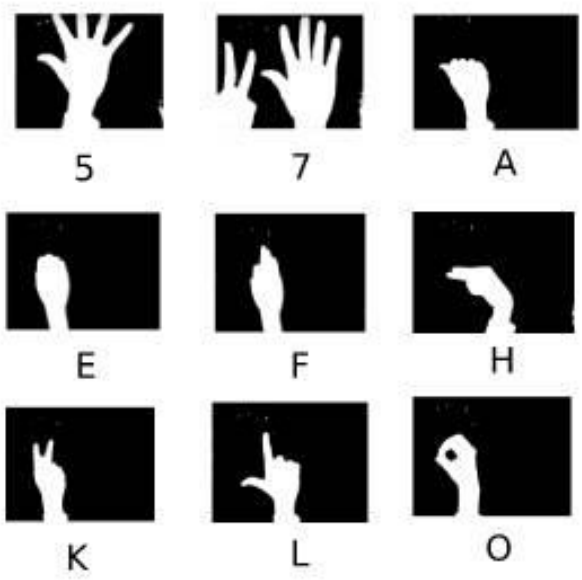


**Fig 12.** Images after applying transformation

One of the most important things about being a very good practitioner is being able to access our results. So, remembering to go to data.show batch and look at the data is very important. Its curious how much we've been given when we actually look at the dataset that we find it has peculiar black borders on it, some of the items have text covering up some of it, or some of it is rotated in odd ways. So make sure you take a look. A model is trained in Fastai using something called a “learner”. Fitting the model so now we have a ConvLearner, we can fit it. You can just use a method called fit but in practice, you should nearly always use a method called fit_one_cycle.

We will be knowing that this number, 4, basically decides how many times do we go through the total dataset, how many times do we show the dataset to the model so that it can learn from it. Each time it sees a picture, it's going to get a little bit better. But it's going to take time and it means it could over fit.

The collected dataset will undergo go Transformation like reducing the size of the image. In our model we reduced the image size to 56/56. We can see the images after transformation. Because of reducing the size of the image, there may be the chance to lose the data in the gesture. We also checked whether any data has lost or not. But nothing has lost from the images. Then the next phase we for is training the model. The prebuilt function cnn_learner in Fastai will help us to train the model. After training has completed then we will save the model. And we will place the trained model path at server location.

After the training has completed then we go for testing the model. We capture the gestures and convert those gestures to gray images. And then converting those gray images binary images. And we then send that image to the server the server with the trained model will classify the image class and finally it will give us response.

**4. RESULTS**

Firstly we have trained our model for 5 epochs. This consumed a time of 34:12 seconds. Here in the Table 1 we can see the epochs from o to 5 it means 5 epochs. 1 epoch is one complete cycle. In 1 epoch the System looks all the images in the database and gets the similar patterns in a particular class. For first five cycles we have got 0.16 error rate. So we will train for three more cycles and see if the error rate reduces or not.

**Table 1.** Finding error rate

| Epoch | Train_ loss | Valid_ loss | Error_ rate | time |
|---|---|---|---|---|
| 0 | 2.403331 | 1.318557 | 0.394850 | 06:47 |
| 1 | 1.682517 | 0.827041 | 0.259657 | 06:45 |
| 2 | 1.224003 | 0.775936 | 0.263949 | 06:43 |
| 3 | 1.094361 | 0.576195 | 0.180258 | 06:49 |
| 4 | 0.961269 | 0.542890 | 0.169528 | 07:06 |

We just trained a model. We don't know exactly what that involved or how it happened but we do know that with 3 or 4 lines of code, we've built something which smashed the accuracy at the time of 2012. 16% error certainly sounds like pretty impressive for something that can recognize different hand gestures.

Let's make our model better. We can make it better by using fine-tuning. So far we fitted 5 epochs and it ran pretty quickly. The reason it ran pretty quickly is that there was a little trick we used. These convolutional networks, they have many layers. We'll learn a lot about exactly what layers are, but for now, just know it goes through a lot of computations. What we were doing was adding a few extra layers to the end and we were only training those. We left most of the model basically exactly as it was, so that's very easy. If we are trying to build a model at something similar to the original pre-trained model (similar to the ImageNet data in this case), that works pretty good. But what we want to do really is go back and train the entire model. This is why we pretty much always use this two stage process. Naturally, when we call fit or fit_one_cycle on a ConvLearner, it'll just fine-tune these few extra layers added to the end and it will run very fast. It will basically never over fit but to really get it good, you have to call unfreeze. Unfreeze is the thing that says please train the whole model. Then I can call fit_one_cycle again. In the Table 2 we can observe that we got an error rate of 0.03 that means we got an accuracy of 96.4%. Learn. Fit_one_cycle (3), this method is used to train the model for 3 more cycles. Here in the Table we can see we have got error_ rate of 0.036481.it means that our model is 96.4% accurate. Today in 2019, with basically about three lines of code, we got 96.4% (i.e. 5% error). So that gives you a sense of how far we've come with deep learning, and particularly with PyTorch and Fastai, how easy things are.

**Table 2.** Finding reduced error rate

| Epoch | Train_ loss | Valid_ loss | Error_ rate | time |
|---|---|---|---|---|
| 0 | 0.964080 | 0.660398 | 0.2188854 | 10:08 |
| 1 | 0.788910 | 0.291939 | 0.111588 | 09:42 |
| 2 | 0.433597 | 0.141201 | 0.036481 | 09:36 |

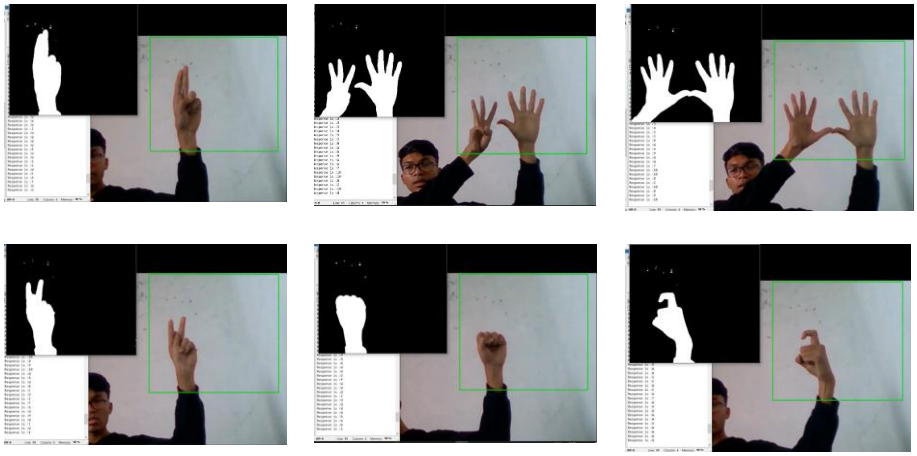

**Fig 13.** Gesture U, 8, 10,
K, S, X recognised by the model

## 5. CONCLUSION

In this paper the hand gesture recognition system is developed using Deep learning model. This system is tested with 36 gestures for 100 different poses per gesture and showed us 98.57% accuracy and average recognition time of 0.098251 secs. Our research will help the medical experts to give the instruction to the robots remotely to add the accuracy in the operations. Our model is also capable of working with the images containing hands of other than skin color. The proposed system will test the images clicked in certain light colors where the hand gestures are clicked and the system only operates for static gestures. In the future, the system can be modified to accept dynamic gestures and the system can be used to create an application for monitoring medical operations. In the current system can be implemented with various techniques for efficient human computer interaction.

## REFERENCES

[1]. Arcana S. Ghotkar and Dr. Gajanan K. Kharatem, "Study of Vision Based Hand Gesture Recognition Using Indian Sign Language", International Journal on Smart Sensing And intelligent Systems, vol. 7, no. I, pp. 96-115, March 2014.

[2]. HaithamHasan, Sameem Abdul-Kareem, "Human-computer interaction using vision-based hand gesture recognition systems: a survey", Neural Comput& Applic, 2013.

[3]. Yi Li. 2012. Hand gesture recognition using Kinect. In Software Engineering and Service Science (ICSESS), 2012 IEEE 3rd International Conference on. IEEE, 196–199.

[4]. Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, “Gradient-Based Learning Applied to Document Recognition.” Proc. IEEE, vol. 86, no. 11, pp. 2278-2324, Nov. 1998.

[5].Wolf,M.T.,Assad,C.,Stoica,A.,You,K.,Jethani,H.,Vernacchia,M.T.,Fromm (2013).Decoding static and dynamic arm and hand gestures from the jplbiosleeve. In AerospaceConference, 2013 IEEE, pages1–9.

[6] Y. Bengio. Learning deep architectures for ai. Foundations and trends

R in Machine Learning, 2009.

[7]. https://course.fast.ai/videos

[8]. J. Gordon and C. Ghez, “EMG patterns in antagonist muscles during isometric contraction in man: Relations to response dynamics,” Exp. Brain Res., vol. 55, pp. 167–171, 1984.