**ACKNOWLEDGEMENT**

This is to thank all those without whom the project would not have been a success. Firstly, I would like to express my gratitude to our guide Mrs. Supriya Aras ma’am and Mrs. Prajakta Soman ma’am for their guidance and support in completing my project. I would like to extend my thanks to BDA department who gave us the golden opportunity to do this project “Virtual Cricket using Hand Gestures”.

Secondly, I would like to thank my group members for their fabulous effort to explore this very topic and support me throughout the completion of project. As this project was mostly completed by staying at home it’s also important to thank our parents and family for the co-operation and providing facilities to us. Last but not least I would like to thank Rohan Kulkarni and Alvyn Abranches for their constant help.

Aadesh Dalvi

Roll no: 11

Contents

[Introduction 1](#_Toc37324346)

[Motivation 2](#_Toc37324348)

[Problem Statement 2](#_Toc37324349)

[Literature Survey 3](#_Toc37324350)

[Solution Design 4](#_Toc37324351)

[Solution Approach 4](#_Toc37324352)

[Technology Stack 5](#_Toc37324353)

[Design Model 7](#_Toc37324354)

[Solution Implementation and Results 8](#_Toc37324355)

[Obtaining Data 8](#_Toc37324356)

[Dataset Details 9](#_Toc37324357)

[Pre-Processing 13](#_Toc37324358)

[Model Used 14](#_Toc37324359)

[Results 18](#_Toc37324360)

[Conclusion and Future Work 19](#_Toc37324361)

[Limitation 19](#_Toc37324362)

[Conclusion 19](#_Toc37324362)

[Future Work 20](#_Toc37324363)

[References 21](#_Toc37324364)

**INTRODUCTION**

## This project “Virtual Cricket using Hand Gestures” uses deep learning technique to recognize the hand gesture and to count fingers in the frame to build an interactive and fun game. By analysing hand signs only, this game can categorize the signs of the users, thus allowing no need of the users for the button, keyboard, mouse, or any special device for the input interface, but simple hand gestures are required to pass user input. Such human machine interaction provides more sense of reality and better interaction during game play between human and machines. Upon the recognition of user's hand gesture, this finger guessing game uses the simple method to map it against the sign randomly selected by the computer, and further determine the game result. As indicated by test result, this motion hand gesture recognition technology allows the users to interact and play games with system in ways that are more realistic and natural, thus providing more enjoyment for people in an effective way.

The second part of the game focuses on chatbot. Chatbot is software used in entertainment industry, businesses and user support. There are different ways to develop a chatbot. Based on the requirement and resources available, we can select any method of implementation. Chatbot which gives responses based on the context of conversation tends to be more user friendly. Here, we have used cosine similarity technique to understand input from user and uses natural language Processing (NLP) techniques to maintain the context of the conversation. This type of chatbots can be used in small industries or business for automating customer care as user queries will be handled by chatbots thus reducing need of human labour and expenditure. This chatbot helps user to understand the rules of the game.

**MOTIVATION**

This is an era of intelligent machines. With the advancement in artificial intelligent, machine learning and deep learning, machines have started to impersonate as human. We wanted to develop a game with less complexity and interactive way so that user of any age group can appreciate it. Simultaneously, we wanted to apply the knowledge we have and explore new technologies to build an end to end system.

**PROBLEM STATEMENT**

In this project, we have developed a Game which uses finger count detection. User and system will generate a random number from 1 to 6 at a same time. If both the number are same game will stop and user will lose otherwise it will continue till 6 rounds. Scoreboard is maintained and result will display at the end of the game. Chatbot will help user to understand rules of the game. User need to ask a query to the bot. Bot will try to understand the query and return the appropriate answer. A web page will be provided to user where he/she can play the game and ask queries to chatbot.

# **LITERATURE SURVEY**

The performance of deep learning neural networks often improves with the amount of data available for preprocessing [1]. As the amount of data increases, deep learning model tends to provide better results hence data plays a very important role while building any deep learning or machine learning model.

Image data augmentation is perhaps the most well-known type of data augmentation and involves creating transformed versions of images in the training dataset that belong to the same class as the original image [2]. Transforms include a range of operations from the field of image manipulation, such as shifts, flips, zooms, and much more.

There are different chatbots available. Some of them use technologies like deep learning or machine learning where as some of them are rule based. Rupesh Singh and co-author used deep learning approach to develop the chatbot. This method uses Tensorflow for developing the neural network model of the chatbot and uses the NLP techniques to maintain the context of the conversation [3]. It can be used for small scale businesses. Drawback of this technique is, it requires large amount of data to learn itself as it is using neural network.

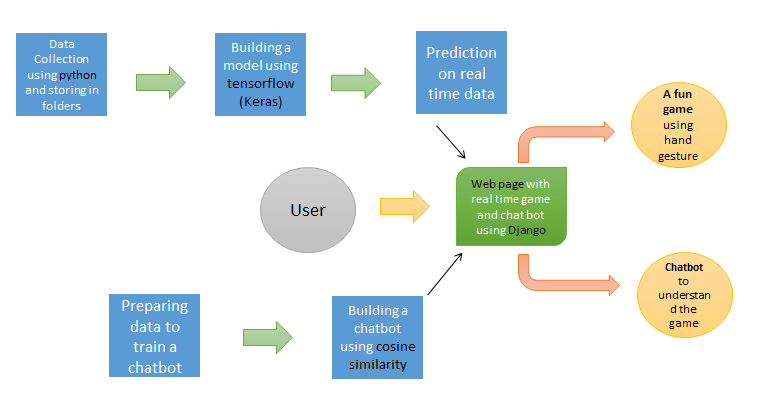
Human machine interaction is a study looking at the transmission and communication of information and emotion between human and machines. In order for human to sense the realistic and comfortable relation when interacting with machines, an interface for natural and intuitive interactions will be important for the bridging of relation between human and machines. Hand gesture, apart from language, is a kind of body language that allows the expression of human emotion and intention. The body movements for signing in finger guessing game can be the most natural language of all for communication and interaction [4].

The human hand gestures are detected and recognized using convolutional neural networks (CNN) classification approach. This process flow consists of hand region of interest segmentation using mask image, fingers segmentation, normalization of segmented finger image and finger recognition using CNN classifier [5]. P. S. Neethu, R. Suguna and Divya Sathish in their paper “An efficient method for human hand gesture detection and recognition using deep learning convolutional neural networks” stated CNN classification approach with enhancement technique achieves high performance with state-of-the-art methods.

# **SOLUTION DESIGN**

## Solution Approach

There are 3 different module of this project. 1. Model to detect hand gesture 2. Chatbot to understand user queries 3. A web page where user can play and ask queries. Based on the study done and available resources we proposed the following solution approach for our problem statement.



There are different approaches for data collection:

1) Download the images from Google and train model on those images. We can use web scraping to download images but we need to delete unrelated images manually. It was a time consuming task. So we decided to go with second approach.

2) Generating our own dataset to build model:

We have used python programming language for this. For data collection, we have used OpenCV library by python to capture the images and OS library to store folder wise images of each sign on our local machines.

There are many efficient pre-trained models used for object detection like Yolo (You only look once) and Mobilenet SSD (Single shot detection). YOLO is object detection system for real-time data processing. It divides image into frames & then forms bounding box around object & predicts class. Mobilenet SSD is designed for mobile embedded apps. It extracts feature map & applies convolution. It is fast and it has simple architecture.

In order to learn and explore more about neural networks, we built our own model for hand sign recognition. To build this model we have used Keras which is an open-source neural-network library written in Python. After building an acceptable model, using OpenCV and the model, we were able to recognize hand sign on real time data.

To build the chat, we used cosine similarity method. It requires less data, space, time and processing power. In this method, we check similarity between two strings and return a numeric values. This technique helps to understand the intent of user. We have provided data to chatbot, which has different intent along with potential patterns/ questions asked by user and expected answers from the bot. using this data, we were able to understand the user input and give proper response.

The last module of this project is web integration. We have used Django framework to this. Django is a high-level Python Web framework that encourages rapid development and clean, pragmatic design. There are other frameworks too like flask. We opt for Django over flask as it allows database connections. Flask is single page application preferable for personal blogs, forums etc. So, we choose Django as its complete package for web development.

## Technology Stack

* Python

Python is a high level, interpreted programming language.

Python has a simple syntax similar to the English language and it also allows developers to write programs with fewer lines than some other programming languages. Python runs on an interpreter system, meaning that code can be executed as soon as it is written.

Python can be treated in a procedural way, an object-oriented way or a functional way.

* OpenCV

OpenCV (Open Source Computer Vision Library) is an open source computer vision and machine learning software library mainly aimed at real-time computer vision. It is cross-platform library and free for use under the open-source BSD license.

* OS

The OS module in python provides functions for interacting with the operating system. It comes under Python’s standard utility modules. This module provides a portable way of using operating system dependent functionality.

* JSON

JSON is a text format that is completely language independent but uses conventions that are familiar to programmers. I JSON, Data is in name/value pairs and separated by commas. Curly braces hold objects and Square brackets hold arrays

* Keras

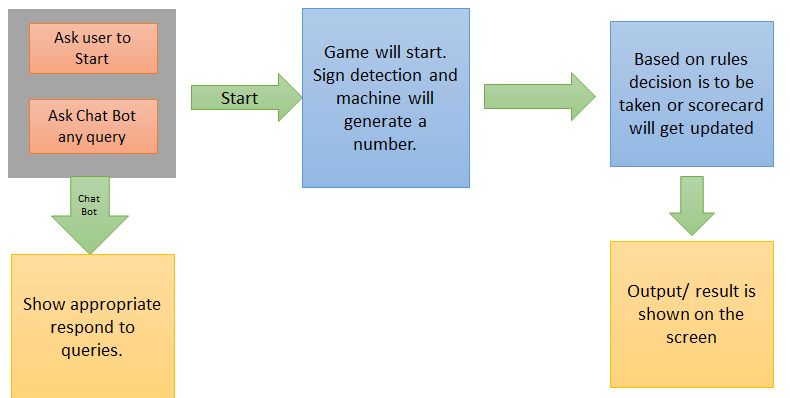
Keras is an [open-source](https://en.wikipedia.org/wiki/Open-source_software" \o "Open-source software) [neural-network](https://en.wikipedia.org/wiki/Artificial_neural_network" \o "Artificial neural network) library written in [Python](https://en.wikipedia.org/wiki/Python_(programming_language)" \o "Python (programming language)). It is capable of running on top of [TensorFlow](https://en.wikipedia.org/wiki/TensorFlow" \o "TensorFlow). It is designed to enable fast experimentation with deep neural networks; it focuses on being user-friendly, modular, and extensible.

* Django

Django is a high-level Python Web framework that encourages rapid development and clean, pragmatic design. It Helps developers take applications from concept to completion as quickly as possible and avoid many common security mistakes with ability to scale quickly and flexibly

## Design Model

We will provide a web page to user where he/she can play game or understand rules of the game by asking questions to chatbot. On the home page, user can start playing game or ask question in the text box given. When he/ she ask any question, it will try to understand the question and return appropriate response using the model built. Once user starts the game, he/she has to make hand signs defined in the region of interest. Model will recognize the signs and simultaneously generate a number. Based on the rules of the game, system number and signs of the user, game will stop or continue till 6 rounds. In the end, result will be displayed on the screen itself. User need to refresh the web page in order to start the game again.



# 

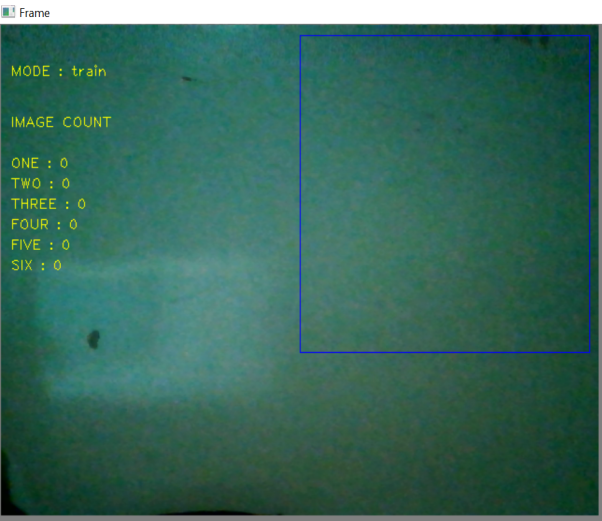
# **SOLUTION IMPLEMENTATION**

## Data Collection

The data was collected by compiling a data collection function (Code) where separate folders for every sign (finger count) where generated. On running the code a frame was generated along with ROI (Region of Interest). So the respective sign was to be shown in the given ROI. Strictly while collecting the data background was kept as plain surface without any distraction to obtain maximum accuracy.



To store the images there was a requirement of pressing corresponding keys on keyboard to record the data. To record sample of “1” press “1”,”2” press “2” and so on till 6. Esc key was used to quit or exit from the frame.

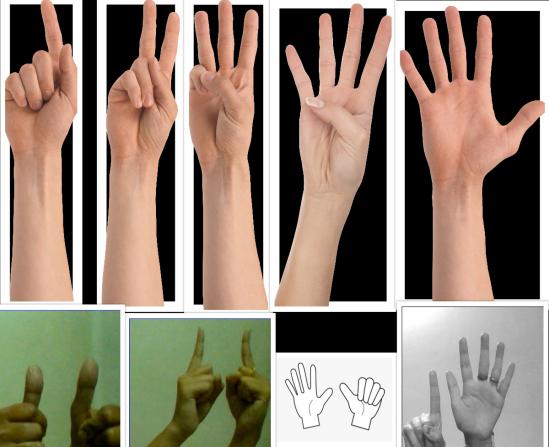


## Dataset Details

There were approximately 1000 such images collected for each sign and further the data folders where split as train, test, validation. Total 6240 images were collected. Size of data collected is 187Mb.

Some basic things to be considered while collecting images are as follows:

* Plain Background in ROI.
* Finger sign to be shown in ROI (Region of Interest) i.e. the blue frame shown in upper image.
* Only hand gesture is to be fitted within the boundaries of ROI , No other body-part or object should be included in ROI while capturing. Gestures
* Only respective key should be pressed while collecting image for that sign. Example: 1 for One, 2 for Two and so on till 6.

Only set of gestures to be used as shown in image above.

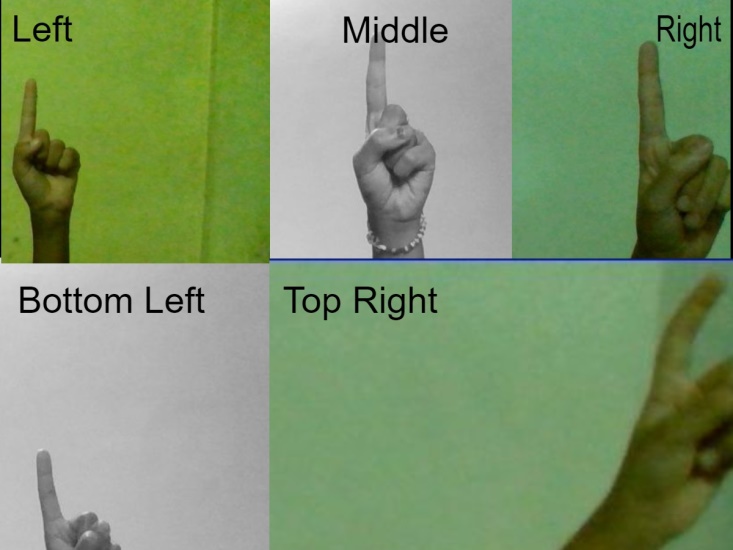
To improve the model accuracy we considered invariance while collecting the data.

Invariance means, you can recognize an object as an object, even when its appearance varies. This is generally a good thing, because it preserves the object's identity, category, (etc.) across changes in the specifics of the visual input, like relative positions of the viewer/camera and the object.

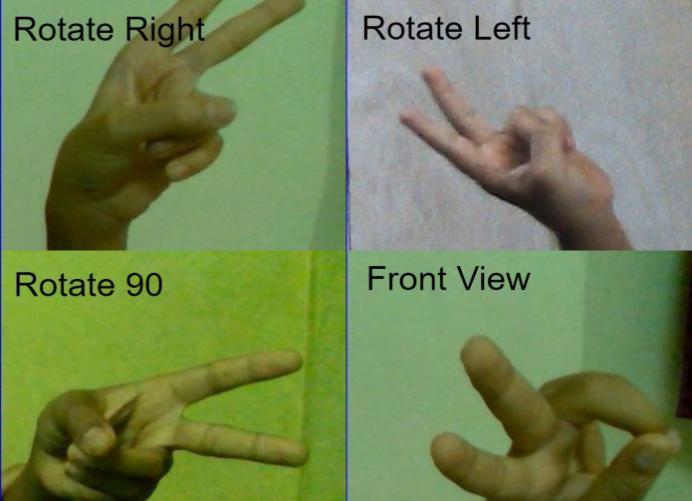
Note: Here the hand/Fingers sign is the object/target.

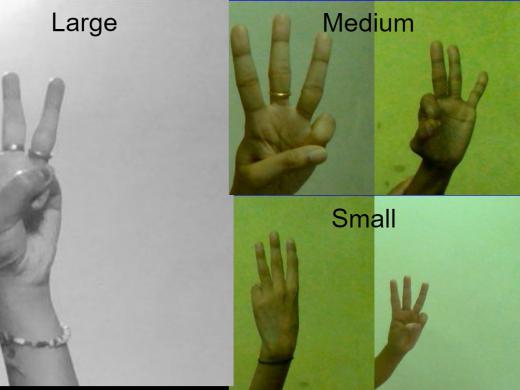
1) Positional / Transnational Invariance:

Ability to detect positional shifts, or translations of the target in the image.

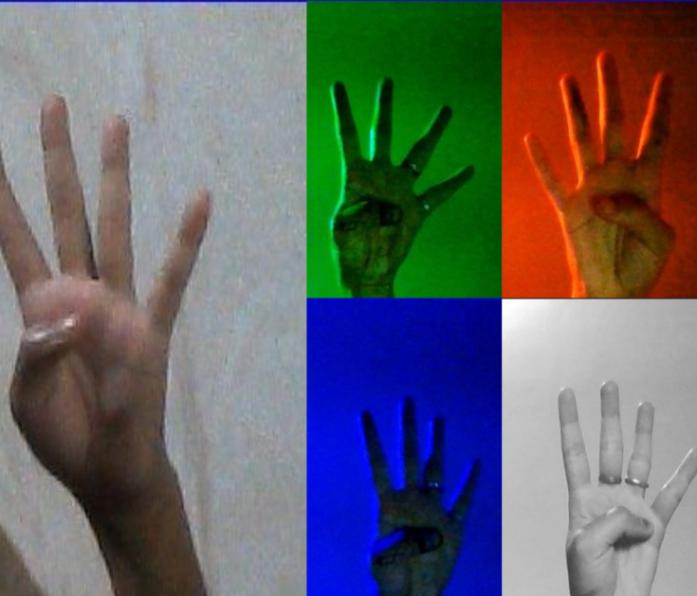


2) Rotation / View-Point Invariance:

Ability to detect circular movement of an object around a center (or point) of rotation.It’s change in view-point of seeing the object.

1. Size Invariance:

Ability to detect change in the size of the target in the image.

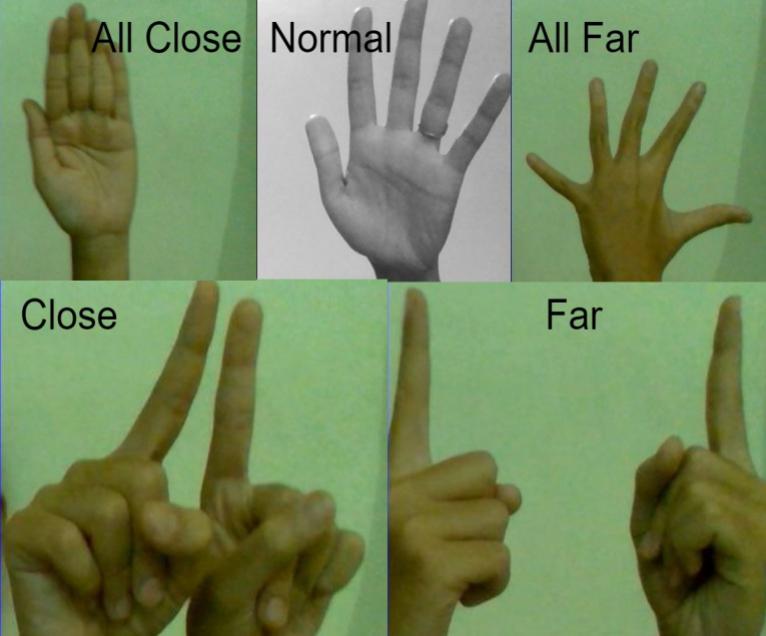


4) Illumination Invariance:

Ability to detect the target in the image even if visibility is low due to light intensity or shadows etc.

5) Space between the Fingers:

Ability to detect the sign even if fingers are close or wide apart.



## Pre-Processing

Data pre-processing plays a vital role while building any model. In data pre-processing, we deal with imbalance data, noise in data if present and we prepare our data to feed to the model.

For this project, the following data pre-processing has been done-

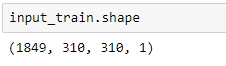
1. Manually deleting inappropriate images.

We have created our own dataset for this project. Images of palm showing different signs were collected for this project. We made sure that only images of palms are captured. We manually checked images and deleted the images with noise.

1. Resize images to (310, 310, 1) if needed; 310 height, 310 width and 1 channel. Before providing input to the model, we made sure that dimensions of each images is same i.e. (310, 310, 1). We have used grey scale images with 310 height and 310 width.
2. As CNN accepts input in 4D format, we created a batch of all images as an input data.



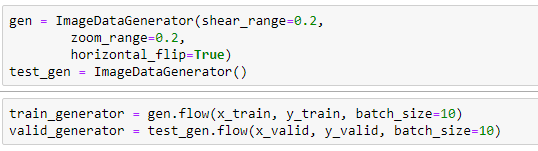
Thus, dimensions of our input data was (1894, 310, 310, 1) i.e. (Batch size, Height, Width, Channel). Similar process has done for validation data.



1. As CNN accepts input of float data type, we changed the data type of our data into float.
2. Normalization helps in efficient computation and gives better performance in terms of time and accuracy. Value of each pixel can be from 0 to 255, thus to normalize this data, we divided our data by 255.



1. Image data generator- In order to make the most of our few training examples, we will "augment" them via a number of random transformations, so that our model would never see twice the exact same picture. This helps prevent over-fitting and helps the model generalize better. In Keras this can be done via the keras.preprocessing.image.ImageDataGenerator class.

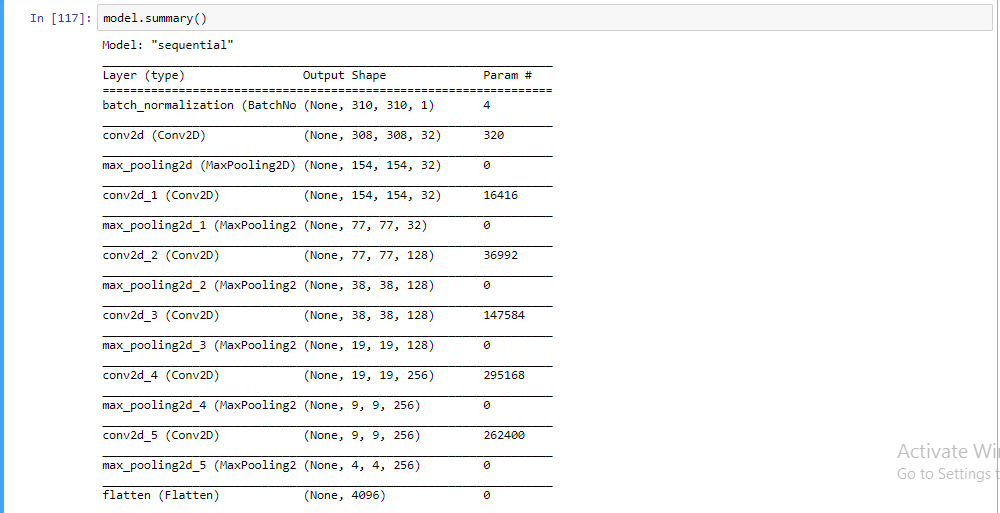


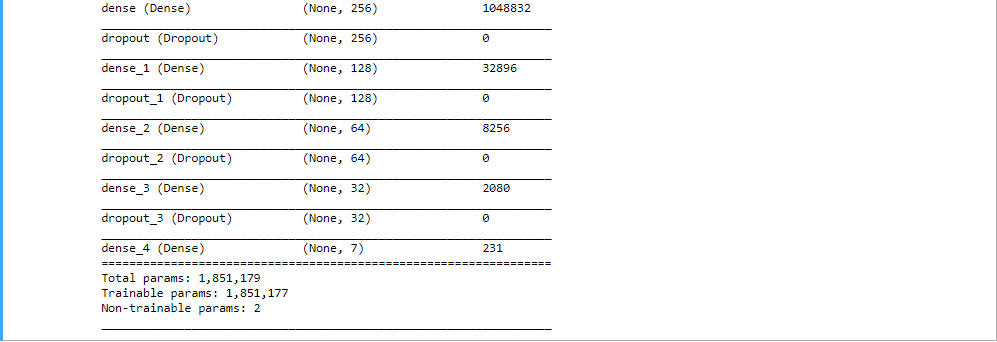
## Model used

We have used Keras to build a model for hand sign recognition.There were many different models with different combinations of layers and number of neurons built. There are different hyper parameters need to be consider while building any neural network. This is the structure of our model used for hand sign recognition.

Our model architecture consists of a Batch Normalization layer followed by Convolution layers.

We have used 6 convolution 2D layers (number of filter being 32,32,128,128,256,256 and kernel size as 3,4,3,3,3,3) each accompanied by activation function ‘relu’ and MaxPooling layer.  
We then used flattening layer so as to give input to the feed forward network. Total 5 dense layers were used having 256, 128, 64, 32, 7 neurons that take ‘relu’ as an activation function. Dropout layers with rate as 0.2 go along with each of these dense layers for regularization.





Pre-requisite - GPU is used to build the model, for this Google Colab is used.

* Sequential: A Sequential model is appropriate for a plain stack of layers where each layer has exactly one input tensor and one output tensor.

In the project, input tensors of size (n\_batches, 310, 310, 1) and output vector of size (7, 1) is used.

* Batch Normalization:  It is used to normalize the output of the previous layers. The activations/filters scale the input layer. Using batch normalization learning becomes efficient also it can be used as regularization to avoid over-fitting of the model.

1 batch normalization layer is used just before the CNN layers.

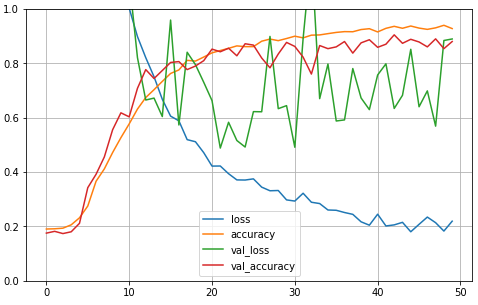
* Convolution Layers: Important layer that is used to do image recognition, image classification etc. Convolution is the first layer to extract features from an input image. Convolution preserves the relationship between pixels by learning image features using pixels of input data. It is a mathematical operation that takes 2 inputs such as image matrix and a filter or a kernel.

In the project, 6 convolution layers are used. The motive is to keep adding layers until over-fit, after which the regularization techniques could be used for generalization.

The convolution layers were accompanied by activation function ‘relu’, the reason is that the images are naturally non-linear, so the rectifier function was preferred.

* Kernels: The kernel is a filter that is used to extract the features from the images. In project, 32, 128, 256 numbers of kernels are used in different Convolution layers. These different convolution kernels each act as a different filter creating a channel/feature map representing something different. Kernels of size 4, 3 were used because the benefit that smaller kernel size provides that it reduces computational costs and weight sharing and extract the more granular features as we move along the layers. Padding function used is ‘same’, doing this improve performance as it retains the information at the borders.
* Pooling-layer Parameters: The objective is to down-sample an input representation (image, hidden-layer output matrix, etc.), reducing its dimensionality by keeping the max value (activated features) in the sub-regions binned. For the project, 6 max-pooling layers (size: 2) were used after every CNN layers.
* Dense Layers: Dense layers are keras’s alias for Fully connected layers. These layers give the ability to classify the features learned by the CNN. For the project 5 dense layers are used (neurons as 256, 128, 64, 32, 7). Activation function ‘ReLU’ is used for hidden layers while ‘Softmax’ is used for the output layer to classify the 7 outcomes (0-6).
* Regularization: Keep adding layers until over-fit. As once we achieved a considerable accuracy in our training set we can use regularization components like l1 / l2 regularization, dropout, batch norm, data augmentation etc. to reduce over-fitting. For the project, Dropout layers are used by switching off 20% of neurons in Dense layers.
* Adam - The superiority of the Adam optimizer lies in its adaptive learning rate and favoured due to its relatively less parameters tuning.
* Loss function: Categorical cross entropy is used in the project as we have multiple classes where each example belongs to a single class.

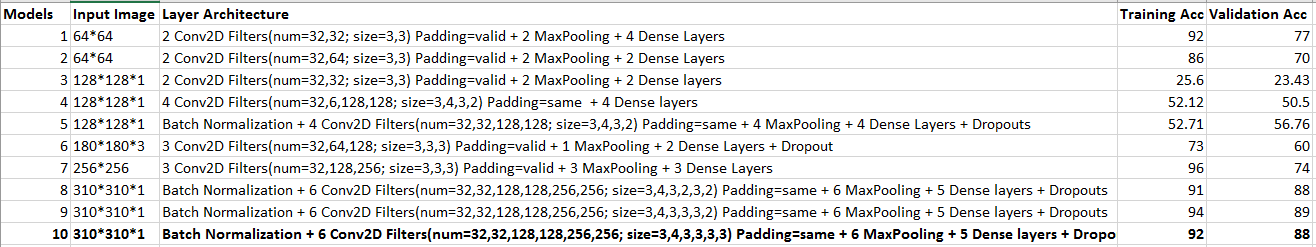
Performance Graph of Model



## **Results**

Below is the table-description for the models created for the project:

Model 10 was chosen as the final one as it stands out for the best performance in terms of accuracy and complexity.



Most of the models were disregarded because of their input size; higher the image resolution, better the accuracy.

Some were set aside because of the over-fitting that caused the higher error for the unknown inputs. (Model 7, 1, 2, 6)

Some of them were overlooked because they completely under-fit the training data. (Model 3, 4, 5).

Model 10 was chosen for the final evaluation because of the fine stability between bias and variance. (Model 9 is equally preferable though)

**CONCLUSION AND FUTURE WORK**

## Limitation

There are few limitation related to this system. To recognize the hand sign, background should be clear or plain. Background with any other object or unclear background does not work for this project.

User can only use some pre-defined specific hand gesture. Hand gesture other than those signs will not work.

User need to show only palm in specified ROI. Any other body part than palm is not allowed in ROI otherwise it might predict the wrong signs. We are looking forward to increase the scope in future.

## Conclusion

This study uses the analysis on human hand movement at time of signing to determine the signing hand gesture and completion timing, and further to perform hand gesture recognition, not only effectively reducing the frequency of hand gesture recognizing in each cycle of finger guessing for the improvement of performance efficiency, but also reducing the rate of misjudging the signing result. We have used machine learning approach to create a bot in this project. Chatbots based on machine learning does not understand the meaning of sentences. It learns how to respond based on previous experience. Though we have used some NLP functions but the actual process through which response is generated is using machine learning. As said earlier, we created the model and trained it with the intent file thus more diverse the intent file more accurate will be the result.

## Future Work

The system can be further extended to Game Intelligence where the machine has the ability to predict the number generated by human in ROI. It will be challenging for user to defeat bot.

A voice module can also be added in order to increase the game ability and agility. In this module, rather than making hand signs, user needs to utter any random number.

The change of background with increasing its ability to detect off a variation can be made available. This will help to overcome the limitation of plain background.

Web page can contain more information like user score per delivery and high score till now etc. A scoreboard can be further integrated to increase a creative aspect of the system.

# **References**

1. https://keras.io/api/preprocessing/image/
2. https://machinelearningmastery.com/how-to-configure-image-data-augmentation-when-training-deep-learning-neural-networks/
3. Singh, R., Paste, M., Shinde, N., Patel, H., & Mishra, N. (2018). Chatbot using TensorFlow for small Businesses. 2018 Second International Conference on Inventive Communication and Computational Technologies (ICICCT). doi:10.1109/icicct.2018.8472998
4. Tseng, K.-T., Huang, W.-F., & Wu, C.-H. (2006). Vision-Based Finger Guessing Game in Human Machine Interaction. 2006 IEEE International Conference on Robotics and Biomimetics. doi:10.1109/robio.2006.340271
5. P. S. Neethu, R. Suguna, Divya Sathish, “An efficient method for human hand gesture detection and recognition using deep learning convolutional neural networks”, 23 march 2020

https://techtutorialsx.com/2018/06/02/python-opencv-converting-an-image-to-gray-scale/

https://www.plsight.com/guides/importing-image-data-into-numpy-arrays

https://www.learnopencv.com/playing-rock-paper-scissors-with-ai/

https://github.com/DarshNaik/Hand-Detection-Finger-Counting

https://www.youtube.com/watch?v=jERbK8OS2yA

https://youtu.be/SgKOTgrGnXI