COMPUTER VISION SOLUTION FOR HEARING-IMPAIRED

MACHINE LEARNING PROJECT

NIRANJAN SHRESTHA AND PRAJIN BAJRACHARYA

OBJECTIVES

- To provide the platform to interpret the series of signs to speech such that it would be much more helpful and convenient for the use specially focusing on Hearing-Impaired people all around the globe.
- To make conversation easier and understandable for people with or without disability and to save the time of both parties

BENEFITS

- Able to predict different hand signs using the real time video interaction.
- Easy in interpreting the meaning and information of various sign languages such that every user becomes aware and know the implementation of those languages in real life as well

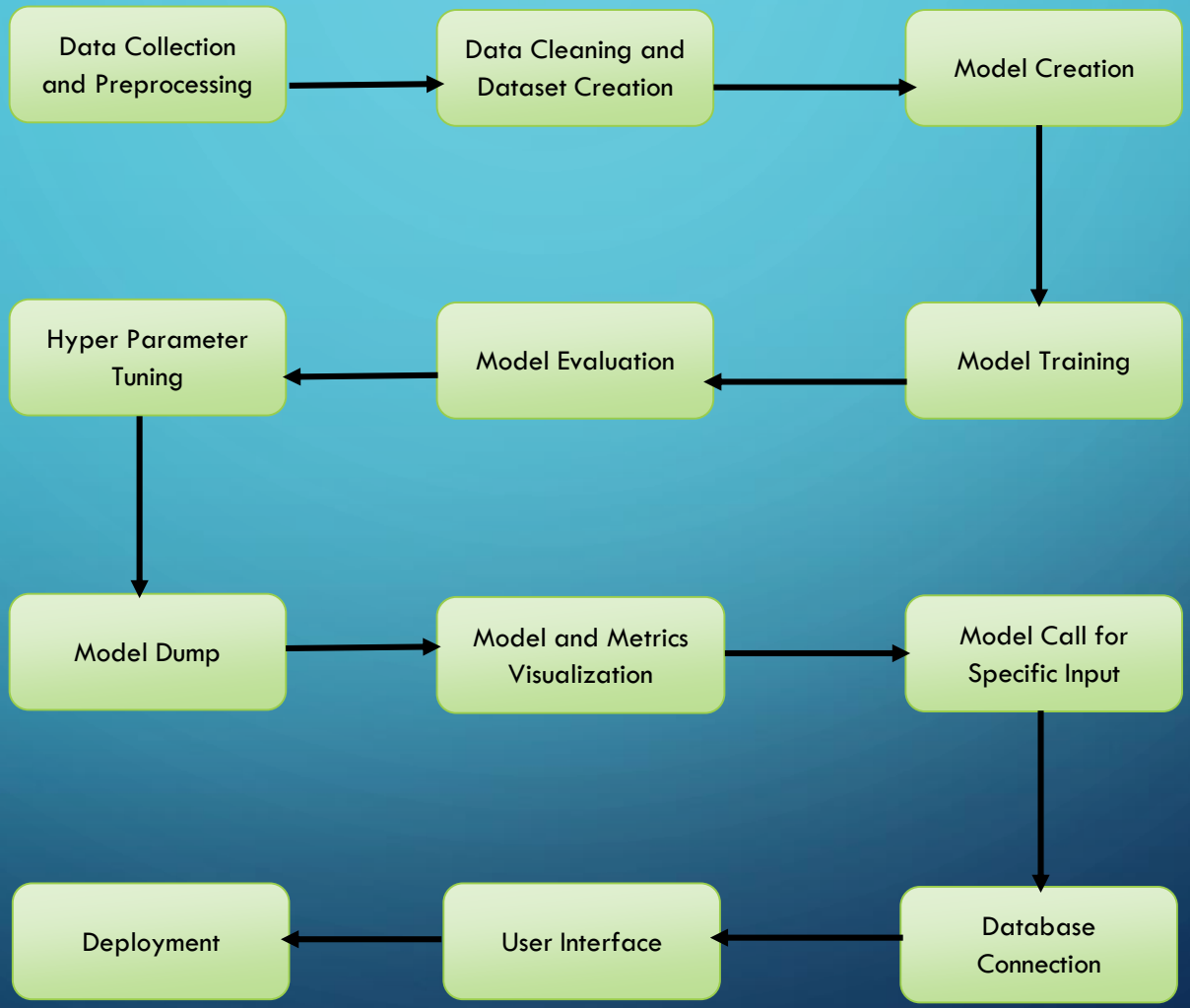
DATA DESCRIPTION

As data must be preprocessed using various methods of OpenCV which are listed below:

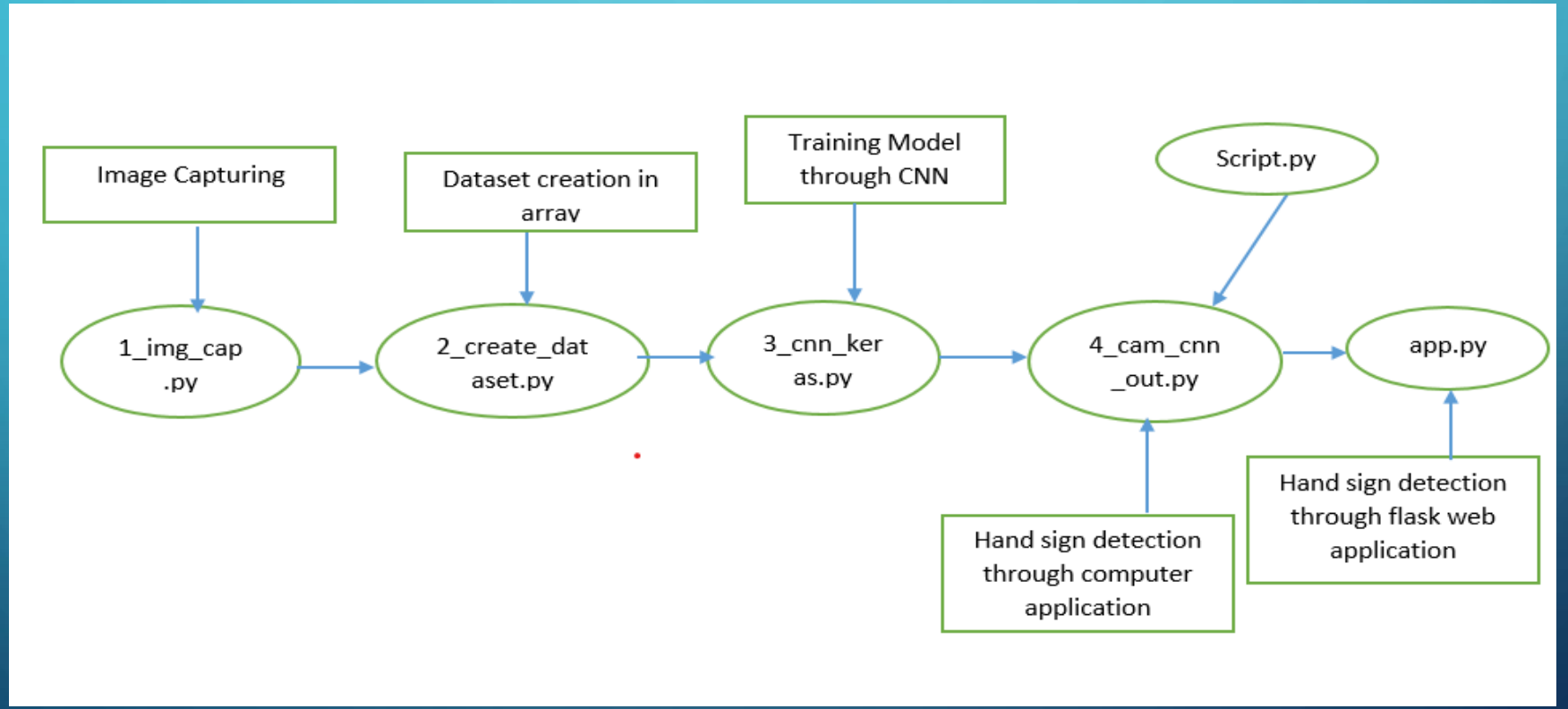
- Transformation
- Smoothing
- Edge Detection
- Segmentation
- Thresholding
- Contours Detection

- For training and testing the model, we used the dataset that are prepared by ourselves in that it includes:
  - 26-class fingerspelling recognition
  - 2-extra function recognition
  - 6-word class fingerspelling recognition
- Then the captured images are converted into pickle file by using pickle library such that it can be easily loaded as input dataset into the neural network.

ARCHITECTURE



METHODOLOGY



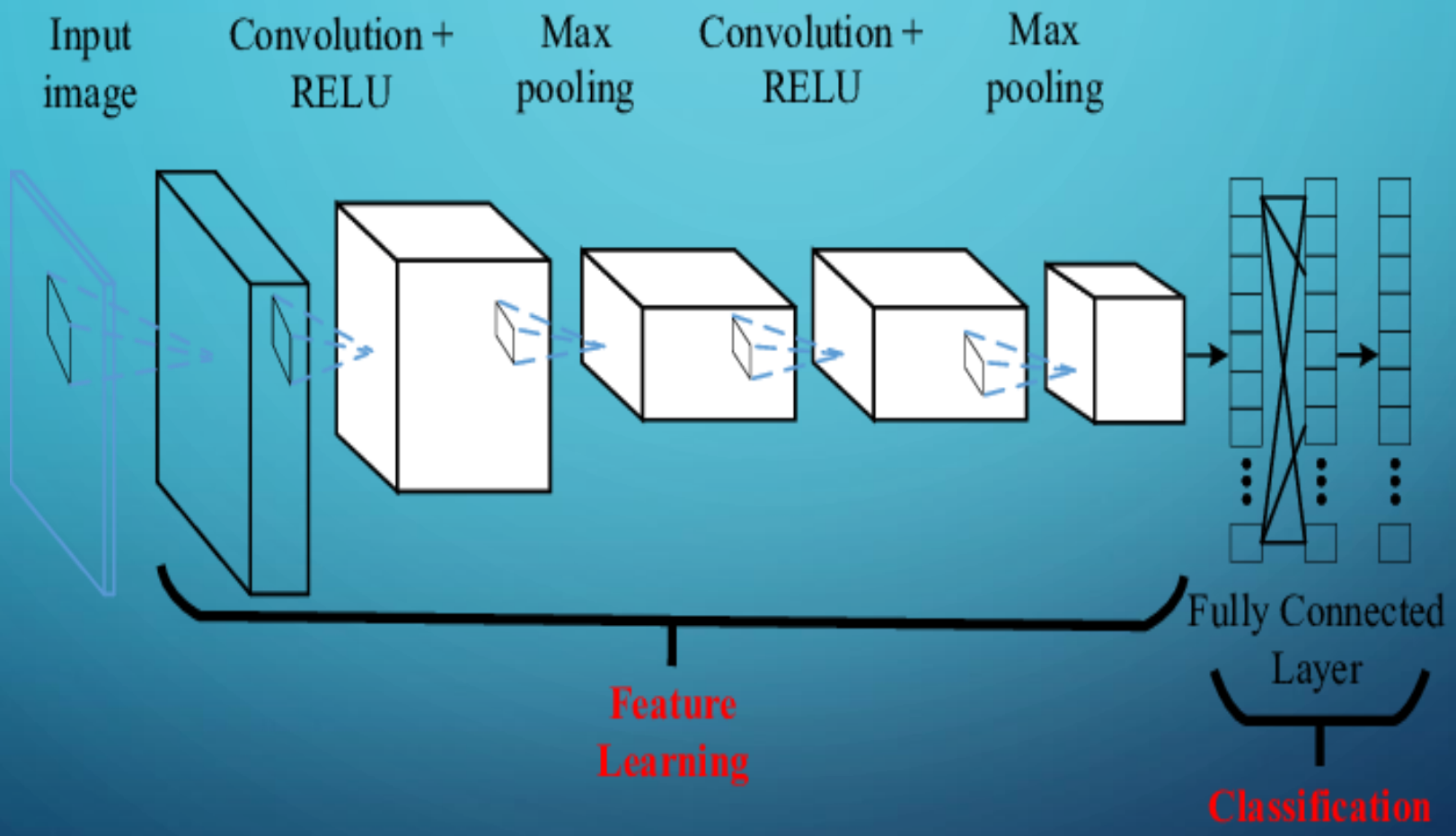
DATA COLLECTION

- All the dataset were made from the img_cap.py
- Images were captured at size of 256*256
- The ROI(region of interest) was thresholded and hence it captured the data set in form of binary image
- Total of 68000 images were created and used

DATA PREPROCESSING

- Images was scaled at its original size
- Total Captured images were divided into 80% Training images and 20 % test images and again 20% of total images were divided as Validation images.
- Thus divided images were pickled and dumped into pickle files

Convolution Neural Network(Overview)

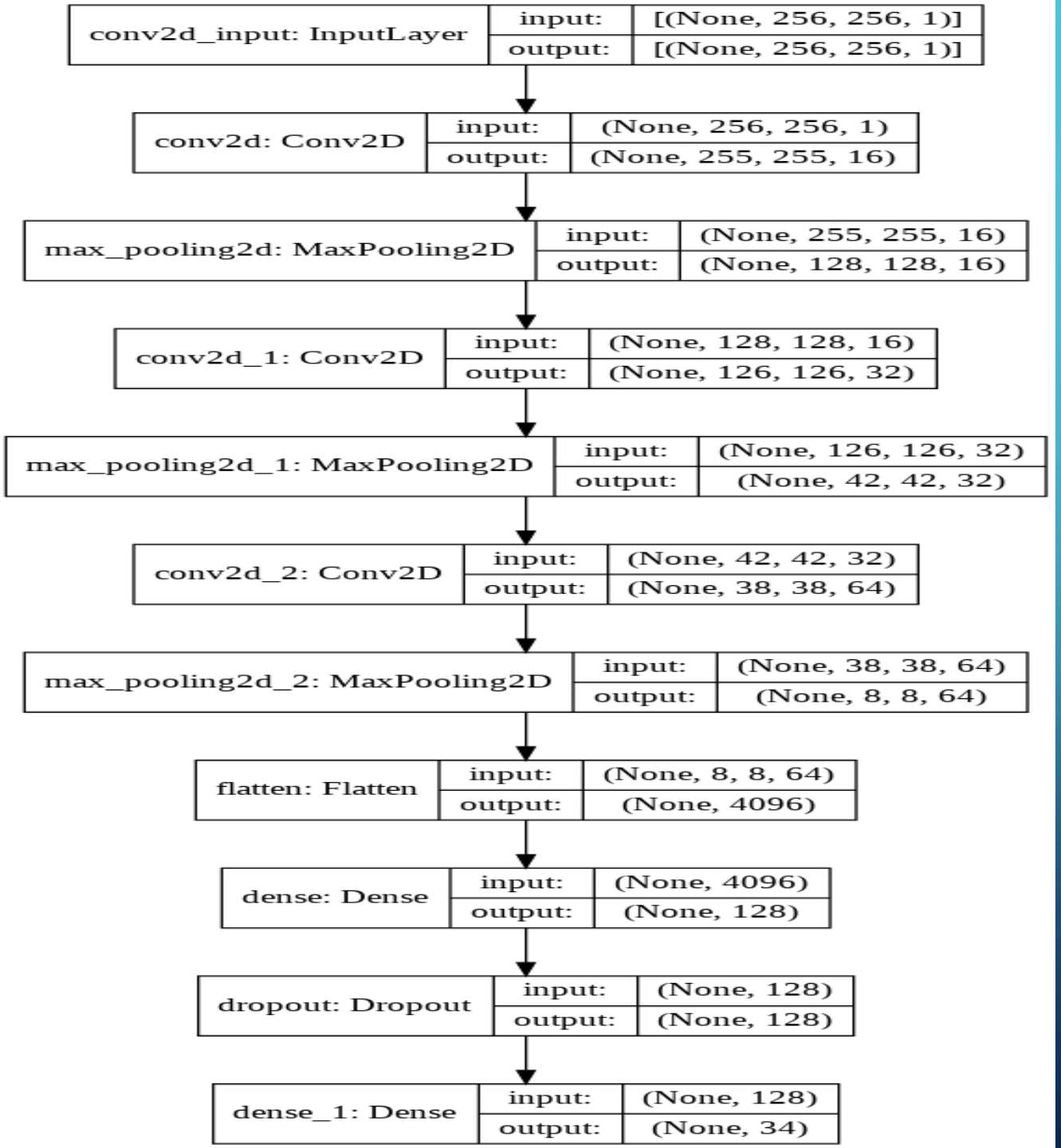


CNN LAYERS

- Convolutional Layer: Filters input signal and extract additional image features
- Pooling Layer: Reduces size of an input by down sampling and controls overfitting.
- Fully-Connected Layer – produces output of neural network

MODEL ARCHITECTURE

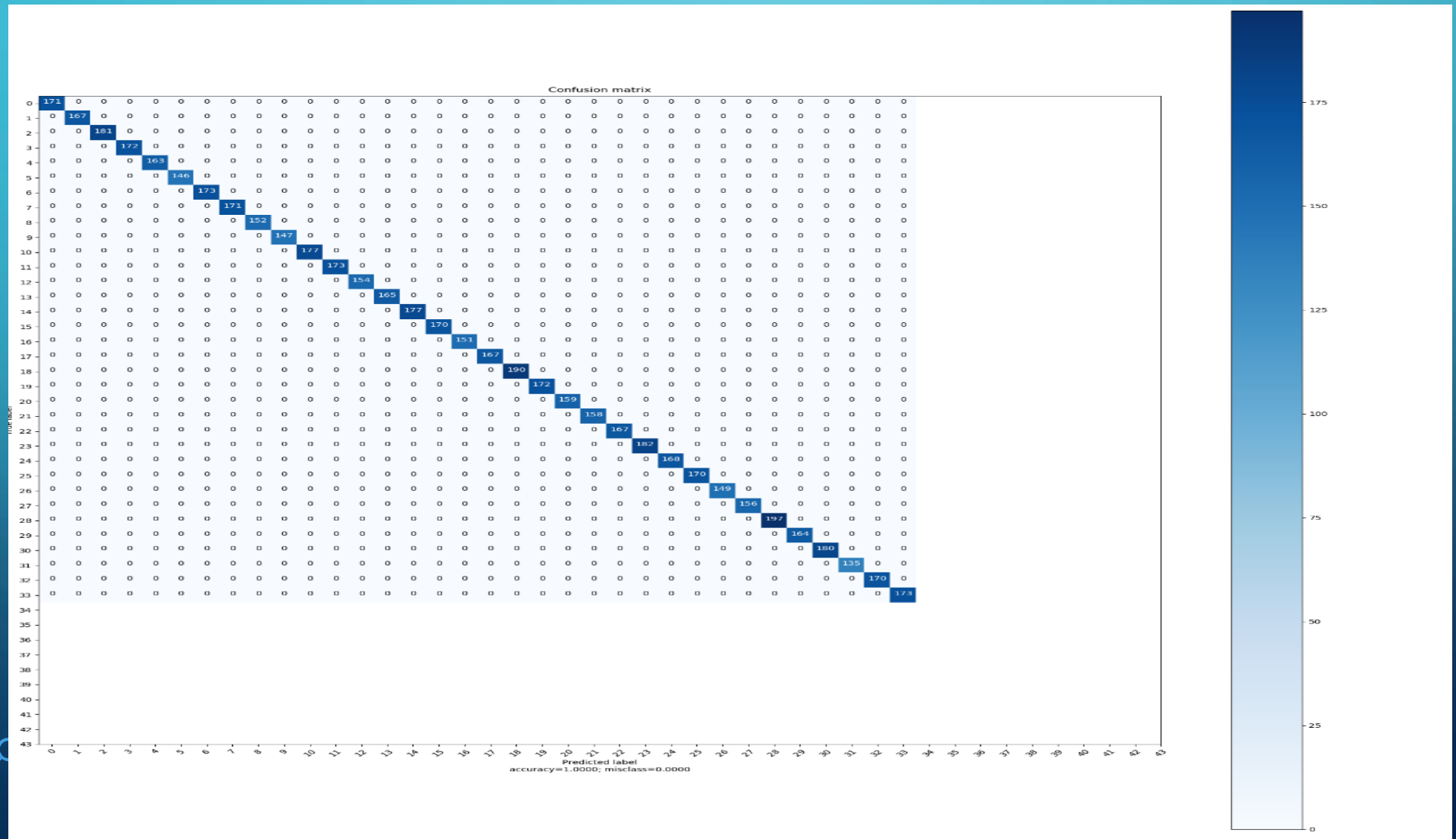
- 256*256*1 sized image for convolution
- 3 convolution layers
- 3 pooling layers
- Activation function – ReLu and Softmax
- Padding(Same)
- Strides
- 1 Flatten layer
- 2 Dense layer(one with 128 neurons and another with 34 neurons)
- Regularization using 1 drop out layer
- Adam Optimizer
- Categorical_crossentropy as loss function
- Best result obtained after 3 epochs till 30 epochs


conv2d_input: InputLayer
input: [(None, 256, 256, 1)]
output: [(None, 256, 256, 1)]
conv2d: Conv2D
input: (None, 256, 256, 1)
output: (None, 255, 255, 16)
max_pooling2d: MaxPooling2D
input: (None, 255, 255, 16)
output: (None, 128, 128, 16)
conv2d_1: Conv2D
input: (None, 128, 128, 16)
output: (None, 126, 126, 32)
max_pooling2d_1: MaxPooling2D
input: (None, 126, 126, 32)
output: (None, 42, 42, 32)
conv2d_2: Conv2D
input: (None, 42, 42, 32)
output: (None, 38, 38, 64)
max_pooling2d_2: MaxPooling2D
input: (None, 38, 38, 64)
output: (None, 8, 8, 64)
flatten: Flatten
input: (None, 8, 8, 64)
output: (None, 4096)
dense: Dense
input: (None, 4096)
output: (None, 128)
dropout: Dropout
input: (None, 128)
output: (None, 128)
dense_1: Dense
input: (None, 128)
output: (None, 34)

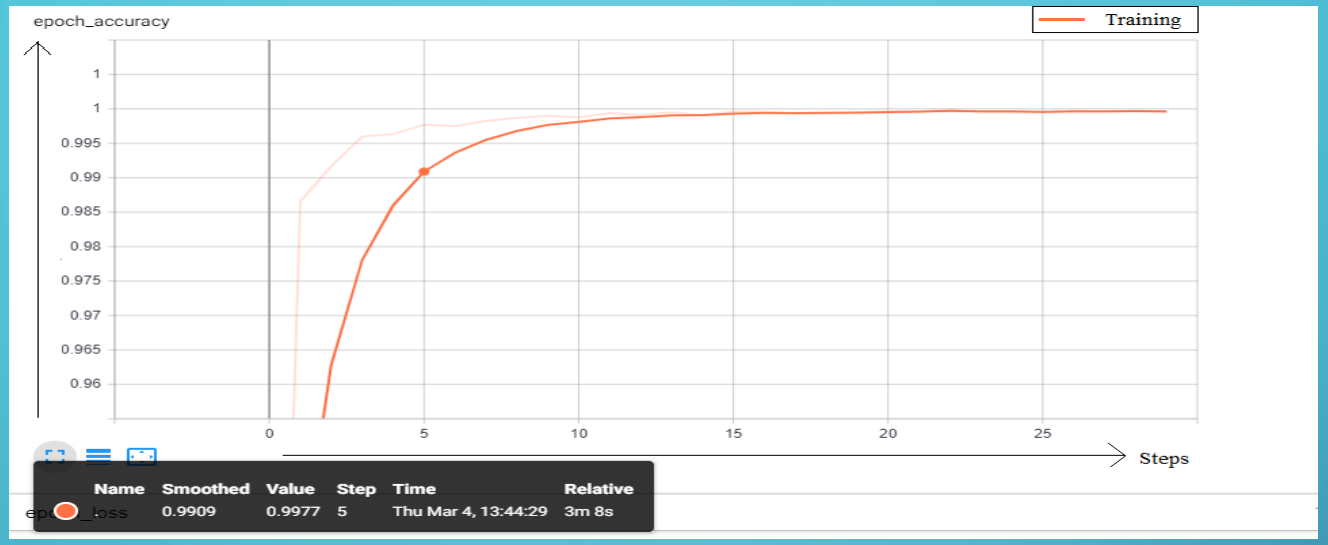
MODEL TRAINING

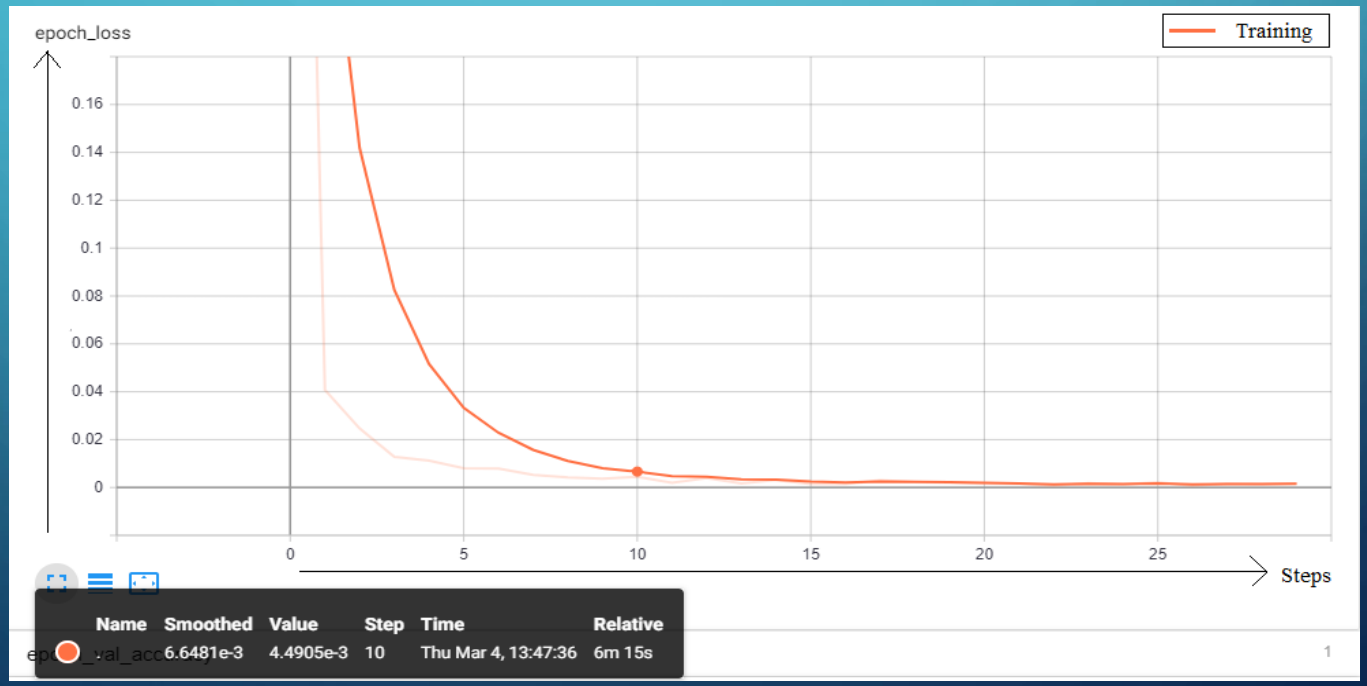
```
Train on 56666 samples, validate on 5667 samples
Epoch 1/30
56666/56666 [==============================] - 40s 704us/step - loss: 0.6420 - accuracy: 0.8617 - val_loss: 0.0069 - val_accuracy: 0.9984
Epoch 2/30
56666/56666 [==============================] - 38s 664us/step - loss: 0.0386 - accuracy: 0.9879 - val_loss: 0.0032 - val_accuracy: 0.9993
Epoch 3/30
56666/56666 [==============================] - 37s 656us/step - loss: 0.0206 - accuracy: 0.9930 - val_loss: 0.0015 - val_accuracy: 0.9995
Epoch 4/30
56666/56666 [==============================] - 37s 655us/step - loss: 0.0154 - accuracy: 0.9953 - val_loss: 3.0699e-04 - val_accuracy: 0.9998
Epoch 5/30
56666/56666 [==============================] - 37s 656us/step - loss: 0.0080 - accuracy: 0.9977 - val_loss: 1.0312e-05 - val_accuracy: 1.0000
Epoch 6/30
56666/56666 [==============================] - 37s 656us/step - loss: 0.0097 - accuracy: 0.9974 - val_loss: 3.6661e-04 - val_accuracy: 1.0000
Epoch 7/30
56666/56666 [==============================] - 37s 656us/step - loss: 0.0057 - accuracy: 0.9983 - val_loss: 6.7759e-04 - val_accuracy: 0.9998
Epoch 8/30
56666/56666 [==============================] - 37s 655us/step - loss: 0.0063 - accuracy: 0.9984 - val_loss: 7.8254e-06 - val_accuracy: 1.0000
Epoch 9/30
56666/56666 [==============================] - 37s 654us/step - loss: 0.0031 - accuracy: 0.9991 - val_loss: 0.0040 - val_accuracy: 0.9993
Epoch 10/30
56666/56666 [==============================] - 38s 665us/step - loss: 0.0047 - accuracy: 0.9987 - val_loss: 2.7464e-05 - val_accuracy: 1.0000
Epoch 11/30
56666/56666 [==============================] - 37s 656us/step - loss: 0.0035 - accuracy: 0.9989 - val_loss: 3.6098e-06 - val_accuracy: 1.0000
Epoch 12/30
56666/56666 [==============================] - 37s 656us/step - loss: 0.0028 - accuracy: 0.9992 - val_loss: 4.7450e-05 - val_accuracy: 1.0000
Epoch 13/30
56666/56666 [==============================] - 37s 655us/step - loss: 0.0032 - accuracy: 0.9991 - val_loss: 1.6126e-04 - val_accuracy: 1.0000
Epoch 14/30
56666/56666 [==============================] - 37s 655us/step - loss: 0.0030 - accuracy: 0.9992 - val_loss: 1.0702e-06 - val_accuracy: 1.0000
Epoch 15/30
56666/56666 [==============================] - 37s 654us/step - loss: 0.0036 - accuracy: 0.9992 - val_loss: 0.0052 - val_accuracy: 0.9988
```

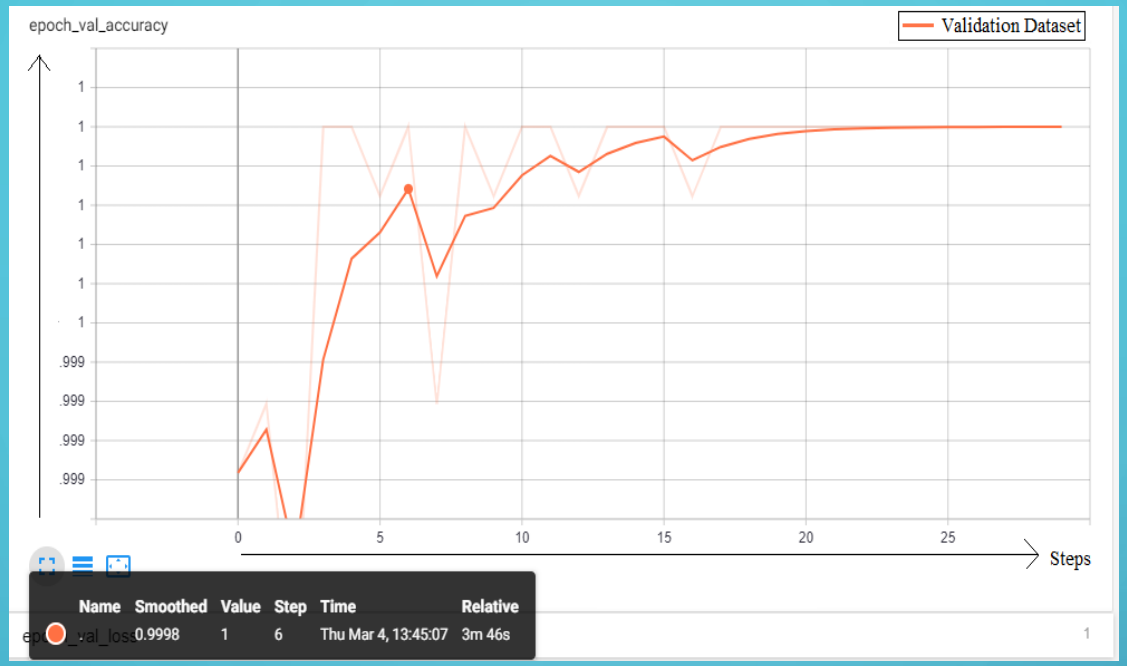
CONFUSION MATRIX

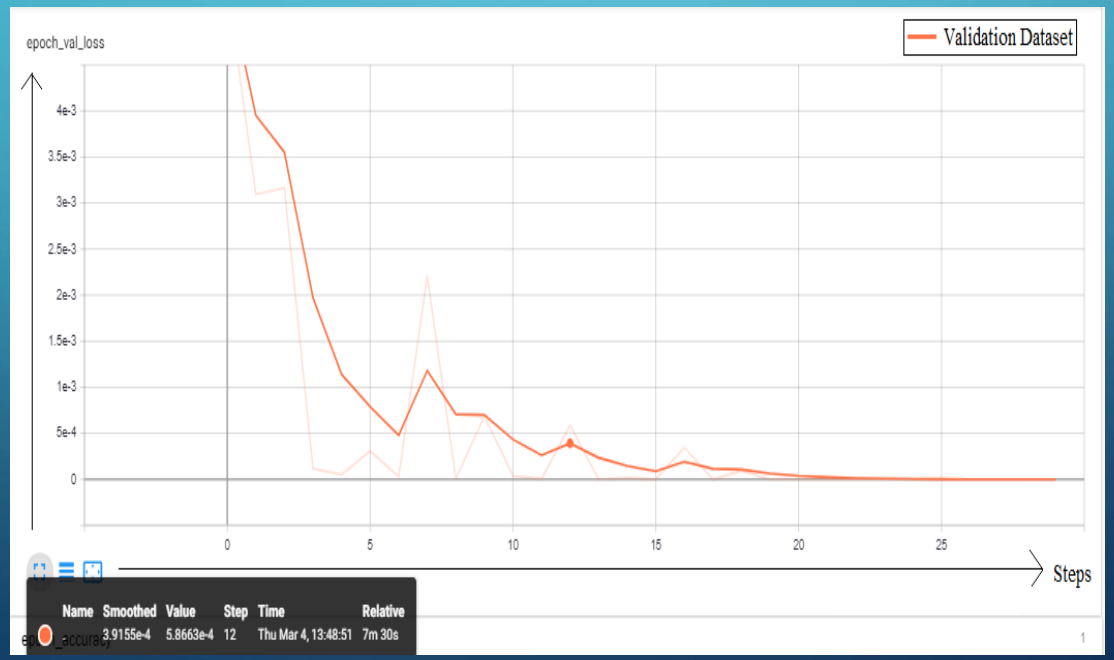


GRAPHS


epoch_accuracy
Training
1
1
0.995
0.99
0.985
0.98
0.975
0.97
0.965
0.96
0
5
10
15
20
25
Steps
Name Smoothed Value Step Time Relative
0.9909 0.9977 5 Thu Mar 4, 13:44:29 3m 8s


epoch_loss
Training
0.16
0.14
0.12
0.1
0.08
0.06
0.04
0.02
0
0
5
10
15
20
25
Steps
Name Smoothed Value Step Time Relative
6.6481e-3 4.4905e-3 10 Thu Mar 4, 13:47:36 6m 15s


epoch_val_accuracy
Validation Dataset
1
1
1
1
1
1
1
.999
.999
.999
.999
0
5
10
15
20
25
Steps
Name Smoothed Value Step Time Relative
0.9998 1 6 Thu Mar 4, 13:45:07 3m 46s


epoch_val_loss
Validation Dataset
4e-3
3.5e-3
3e-3
2.5e-3
2e-3
1.5e-3
1e-3
5e-4
0
0
5
10
15
20
25
Steps
Name Smoothed Value Step Time Relative
3.9155e-4 5.8663e-4 12 Thu Mar 4, 13:48:51 7m 30s

DATABASE CONNECTION

- **Database creation and connection:** Create the database with the key space name passed (if It's not already present). Connect with the database.
- **Table Creation:** we have to create a table in Database in that key space(if it's not already present) to insert valid data files, if Table is already created then we need to insert data into database
- **Insertion:** All valid files are inserted into the tables

PREDICTION

- A user must register or login in the web app such that the user can access the camera for sign prediction along with speech.
- User also can access Speech To Text page for converting speech to text corpus.
- For prediction hand signs user are advised to wear gloves so that accurate result are obtained and also the background should be like a wall or with less objects.

RESULT AND DISCUSSION

- During training the model, best result was obtained with 3 epochs till 30 epochs and training accuracy of 90% was achieved.
- Test accuracy achieved was 85 %.
- Even though Train accuracy of 90% and Test accuracy of 85%. The real time video capture multiple number of frames so accuracy is 75 to 85%.
- Considering the dataset, the performance proposed model is satisfactory.

QUESTION & ANSWERS

Question 1: Explain about the Project and your day to day task:

Answer: Computer Vision Solution for Hearing-Impaired may be the solution of the difficulties occurred for conversing with the Hearing-Impaired people with implementation of real-time prediction of sign languages used for various specific signs. However, this project cannot be used for day to day conversation like in normal real life between the people with hearing disability as they use their languages so quickly such that it becomes difficult to predict those languages.

As a machine learning engineer, I was involved in each and every phase of the project. My responsibility consisted of creation of suitable model which predicted the outcomes with greater accuracy and also the presenting the model in Web application and also the deployment of the model as well. Similarly, I was responsible for monitoring the deployed model for any issues. Mixed in are calls, stand ups and the attending Scrum meeting.

Question 2: How Logs Are Managed?

Answer : We are using different logs as per the steps that we flow in validation and Modeling Like File Validation Log, Data preprocessing, model training logs, prediction log etc.

Question 2: What is the source and size of data?

Answer: The image we captured through web cam is of size 256*256 so the total size of dataset is about GB but due to use of pickle the data are splitted in Training and Testing Dataset ,also is compressed in MB.

Question 3: How Prediction was done?

Answer: The real time dataset was provided in the model through the use of webcam and also with the help of OpenCV which prediction was done.

Question 4 : What is AUC Curve?

Answer: AUC stands for "**Area under the ROC Curve**" .AUC measures the entire 2D area underneath the entire ROC curve.

Question 5 : What is the type of Data?

Answer: The data is the Multimedia type of data.

Question 6: What techniques are you using for data pre-processing?

Answer: For data pre-processing we are using a library known as OpenCV and we used following techniques of OpenCV:

- Transformation
- Smoothing
- Edge Detection
- Segmentation
- Thresholding
- Contours Detection

Question 7: Does Your Dataset Show Normally Distributed Or Not? IF Not Then Which Techniques You will Use To Make IT normal?

Answer : Yes, These Data set Shows Normally Distributed as the data are captured through use of OpenCV and each and every image are similar to each other in the consent of Size and dimensions.

Question 8: Which Tool You Used for Implementation of This Model?

Answer: 1) IDE: VScode

2) Database: Cassandra

Question 9: How were you training those dataset?

Answer: For the training phase we used Google Colab because it contained the free ram and gpu such that the dataset were properly trained and model was created with higher accuracy. Similarly, during the training phase we had done model training in our pc but due to shortage of ram and less advanced Gpu we had to switch to Google Colab.

Question 10: In which technology you are most comfortable?

Answer: I have worked in almost all the fields of Artificial Intelligence such as Computer Vision, Natural Language Processing(NLP) etc. But I personally prefer Computer Vision.

Question 11: What kind of Challenges have u faced during the project?

Answer: There are lot of challenges that had came during the various phases of project development. Some of the biggest challenges are:

- Training the Model with appropriate resources such that accuracy would be greater
- Managing the lighting issues from the palm of the hand which would make error in prediction.

Question 12: What is Accuracy?

- Answer: Accuracy Is One Metric For Evaluating Classification Models.

Accuracy = Number Of Correct Predictions /Total Number Of Predictions

Question 13: What will be your expectations?

Answer: I expect to work on different projects to enhance my technical skills and explore the world of Artificial Intelligence.

Question 14: What is your future objective?

Answer: My future objective is to learn and explore the field of AI and also want to contribute all the knowledge and skills which I have learnt in research purposes and also for finding new things related to AI.

Question 15: How did you optimize your solution?

Answer: We optimized the solution by following ways:

1) Train with better data or do data-preprocessing in efficient way.

2) Using Manual Search Hyper-parameter Optimization.

3) Using High End GPU and lots of Ram for feeding the dataset to neural and training the neural network.

4) Using hand glove in hand for predicting hand signs such that no reflection occurs and no lighting issues occurs.

Question 16: At what frequency are you retraining and updating your model?

Answer: The model was trained once only with appropriate hyper-parameter such that it predicted it with reasonable accuracy.

Question 17: Which is more important to you model accuracy or model performance?

Answer: Actually, model accuracy is sub division of model performance. As a result model accuracy is directly proportional to model performance and hence better the performance of the model, more accurate are the predictions

Question 18: What is Overfitting, and How can You Avoid it?

Answer :

Overfitting is a situation that occurs when a model learns the training set too well, taking up random fluctuations in the training data as concepts. These impact the model's ability to generalize and don't apply to new data.

When a model is given the training data, it shows 100 percent accuracy technically a slight loss. But, when we use the test data, there may be an error and low efficiency. This condition is known as overfitting.

There are multiple ways of avoiding overfitting, such as:

1. Regularization. It involves a cost term for the features involved with the objective function
2. Making a simple model. With lesser variables and parameters, the variance can be reduced
3. Cross validation methods like k folds can also be used
4. If some model parameters are likely to cause overfitting, techniques for regularization like LASSO can be used that penalize these parameters

Question 19: What is Bias-Variance Trade Off ?

Answer: If our model is too simple and has very few parameters then it may have high bias and low variance. On the other hand if our model has large number of parameters then it's going to have high variance and low bias. So we need to find the right/good balance without overfitting and under fitting the data.

This tradeoff in complexity is why there is a tradeoff between bias and variance. An algorithm can't be more complex and less complex at the same time.

Question 20: Question 21 : Explain the Confusion Matrix with Respect to Machine Learning Algorithms ?

Answer: Answer :

A confusion matrix (or error matrix) is a specific table that is used to measure the performance of an algorithm. It is mostly used in supervised learning; in unsupervised learning, it's called the matching matrix.

The confusion matrix has two parameters:

1. Actual
2. Predicted

It also has identical sets of features in both of these dimensions.

Question 22 : How Will You Know Which Machine Learning Algorithm to Choose for Your Classification Problem?

Answer : While there is no fixed rule to choose an algorithm for a classification problem, you can follow these guidelines:

- If accuracy is a concern, test different algorithms and cross-validate them
- If the training dataset is small, use models that have low variance and high bias
- If the training dataset is large, use models that have high variance and little bias