**PERCEPTION OF GESTURE TO COMMUNICATE WITH DEAF AND DUMB**



***PROJECT BY:***

**SAI MAHESH MUDAVAT**

**RAJESH NEMANI**

**HARSHA VARDAHAN GANGINENI**

**TARUNENDRA MALEPATI**

**Objectives and Goals:**

1. Motivation:

The motivation for this project was that we can understand the depth about the image recognition and also, we can help disabled with these learning in a more accurate fashion.

1. Significance:

The Significance of this project even though having many recognitions models the RESNET framework stood up first. So, we are using the RESNET model with wide range of filters, which will make the model better and even more reliable.

1. Objectives:

The Objective of this project is to extract features by employing filtering techniques such as scale-invariant feature transform, accelerated robust features, histogram of directed gradients, edge detection, Gabor filter, and spatial filter. The output of the filters is then analyzed using graph models such as Receiver Operating Characteristics, Confusion Matrix, Classification Report, and Area Under Curve.

1. Features:

The main features we got extracted from different filters we used. With the help of SIFT, SURF and ORB etc. we are extracting the features and provide it to the RESNET for model building based on those features.

**Introduction**:

This project, which we are currently creating, is primarily concerned with the image's feature extraction. The motions are presented by the people, and the model must recognize them properly. As a result, identifying them requires the extraction of characteristics. The features are retrieved using a variety of approaches available for feature extraction, and the model is trained using prediction models. We can gather a good amount of characteristics for the model this way, and then pick the most suited model to attain high accuracies.

**Background**:

This project is one-of-a-kind since it is developed on Residual Network. Utilizing technology provides several additional benefits, such as greater speed and accuracy.

The Residual Network technology was designed to alleviate the deterioration problem. As a result, the framework makes it easier to train networks that are far deeper than previous frameworks.

There are 152 layers in this framework. The layers are similar to Convolutional Neural Networks, except they skip a few levels to reduce the framework depth. This also includes ImageNet, which has predetermined weights for the layers, which aids in enhancing the framework's performance.

**Architecture and workflow:**

Diagram

Description automatically generated

RESNET is used to improve the accuracy and to overcome the drawbacks in the convolution network. In RESNET the input image is sent as feedback to the output image or the processed image. Let’s suppose that f(x) is the output of the CNN for the original image x then, output of RESNET is h(x), which is f(x)+x

Diagram

Description automatically generated

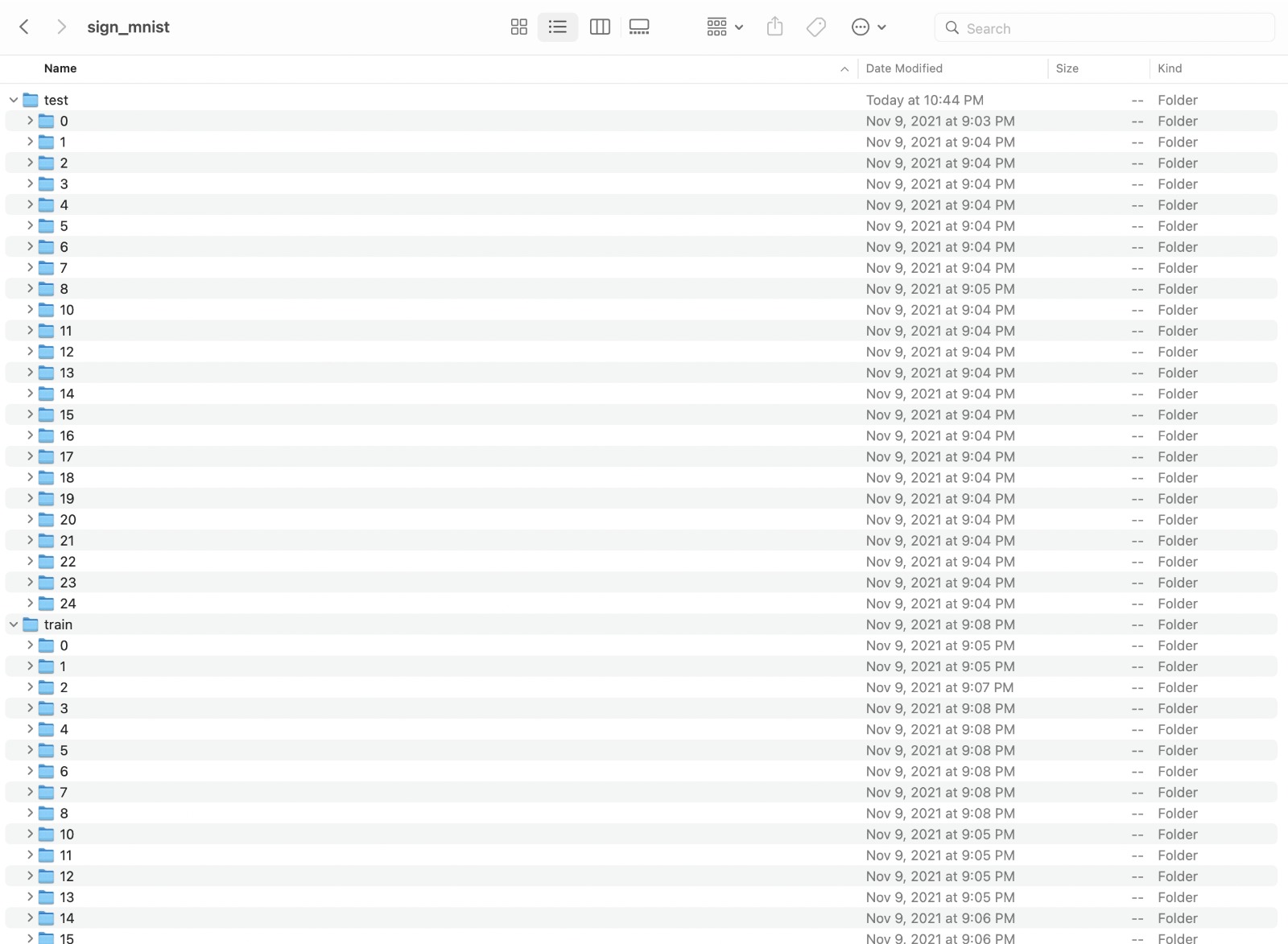
The above the work flow diagram of RESNET.

**Data set**:-

In this project we used MNIST dataset from Kaggle, this is by far the best dataset we could find, as it has different scenarios that come across the regular projects like sharpness levels, angles, blur ratios, pixel levels and brightness level. By this dataset, we can train the model in a better way to get the better accuracy.

A collage of a person's hand

Description automatically generated with low confidence

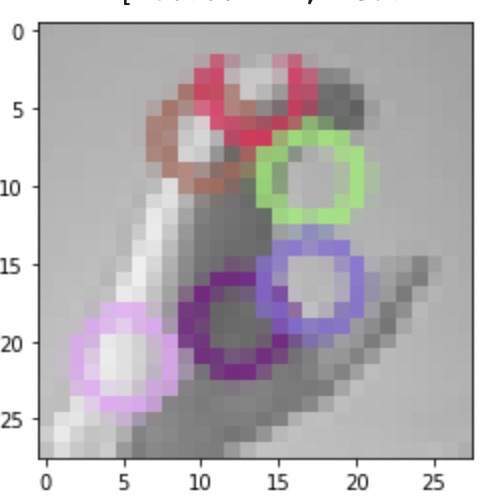


Above is the folder structure of the dataset used. This contains train data and test data in 2 different folders, each folder in the train and test contains different hand gestures in different angles, in different brightness and sharpness levels.

**FEATURES**:

1. **Scale - Invariant Feature Transform(SIFT)**:

This filter solves the problems of its predecessors, which were that feature extraction was not feasible for a comparable picture that was rotated or an image obtained from a different perspective. In this filter, we employ Gaussian difference, which is significantly quicker and more precise. The input picture is gaussian convoluted with increasing deviations, then downscaled and convoluted again until the least point is found. The picture key points are then retrieved using the Difference of Gaussian, and the similarity for two distinct images is tested using the Euclidean distance.



1. **GABOR Filter:**

When a Gabor filter is applied to a band pass image, it delivers the greatest results towards the borders and at texture transition points. At the feature's spatial position, the filter has a separate value. Gabor filters are subclasses of bandpass filters that accept a certain 'band' of frequencies while rejecting others.

Chart, histogram

Description automatically generated

1. **Oriented FAST and Rotated BRIEF:-**

This is the combination of both FAST key point detector and BREIF descriptor. Then several modifications are made to the module to improve the performance. In this we would apply FAST to images first then apply Harris corner measure to find the top N points among all the key points displayed/detected.

1. **Speeded Up Robust Filter**:

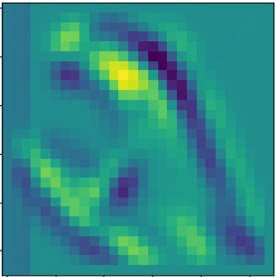
Box filters are used to estimate the Difference of Gaussian. Instead of Gaussian averaging, squares are used to approximate the picture since convolution with squares is much quicker than convolution with the integral image. To generate the SURF feature descriptor, a neighborhood surrounding the key point is picked and partitioned into subregions, with wavelet responses collected and represented for each subregion. The Laplacian sign, which was earlier computed in the detection, is used for underlying interest sites.

1. **SPATIAL:**

Spatial filtering modifies a picture by substituting the value of each pixel with a function of the values of the pixel and its neighbors. If the action executed on the image pixels is linear, the filter is referred to as a linear spatial filter. The filter is, in any case, a spatial nonlinear filter.

1. **SOBEL:**

The Sobel filter is a type of edge detection filter. Edges denotes a sharp change in color or intensity. The kernel for this filter is two-dimensional, with positive attributes in the very first columns and negative values in the last, resulting in the edges as the output when convoluted. This might produce either positive or negative results. The high values are interpreted as having high intensities and are printed. Below is the original image along with the filter applied image of the original image.



**Analysis of Data:-**

Though the data has several features, before feeding the data in to model, it is best practice to preprocess. Here the preprocessing would include extraction of features from the dataset, for example we may have unnecessary noise in the images. By applying edge detection we could filter out unwanted noise from the image. The key points can be extracted by using SIFT or SURF filters which will focus on subject excluding the noise.

**Implementation:-**

**Algorithm/Pseudo code and Explanation:**

1. Reading the image and applying the filter to the image

Preprocess\_img(img):

Img = img \*ftr;

Return img;

1. Splitting the image dataset into train dataset and validation dataset using image data generators

Train , validation = Imagedatagenerator(img,Preprocess\_img)

1. Applying RESTNET model on the dataset with the preprocessed image weights

Conv\_model = resnet50.Restnet50(weights,include\_top,include\_shape)

X=keras.layers.Dense(shape,activation=’relu’)

Predictions = keras.layers.Dense(shape,activation=’softmax’)

Full\_model.compile(loss,optimizer,metrics)

1. Printing the accuracy of the filter and adjusting the epochs accordingly

Full\_model.fit\_generator(train,validation,workers,epochs)

Full\_model.predict()

1. Using the predict model class we will predict the outputs and plotting the classification report, confusion matrix.

Classification\_report(actual,predict)

Confusion\_matrix(actual,predict)

**Results**:

On applying different filters to the dataset and feeding the data to the model provides the accuracy and loss for both training and validation phases.

**ORB**:-

Chart, line chart

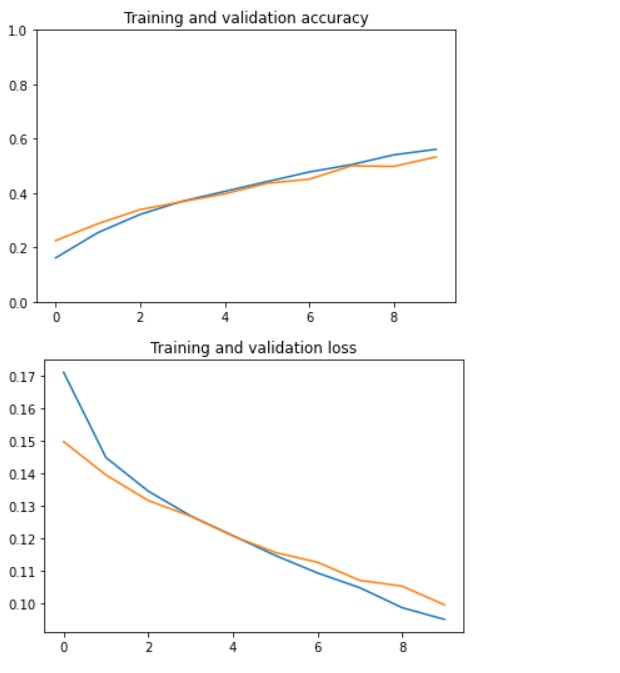
Description automatically generated

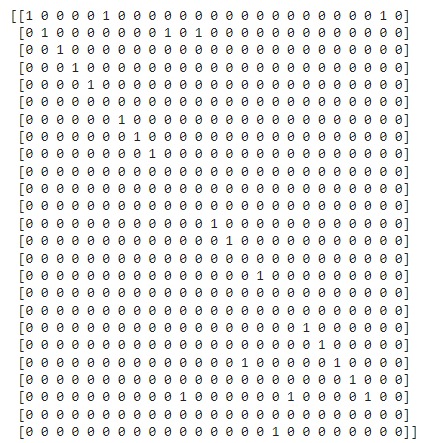
Table

Description automatically generated A picture containing dog, tiled

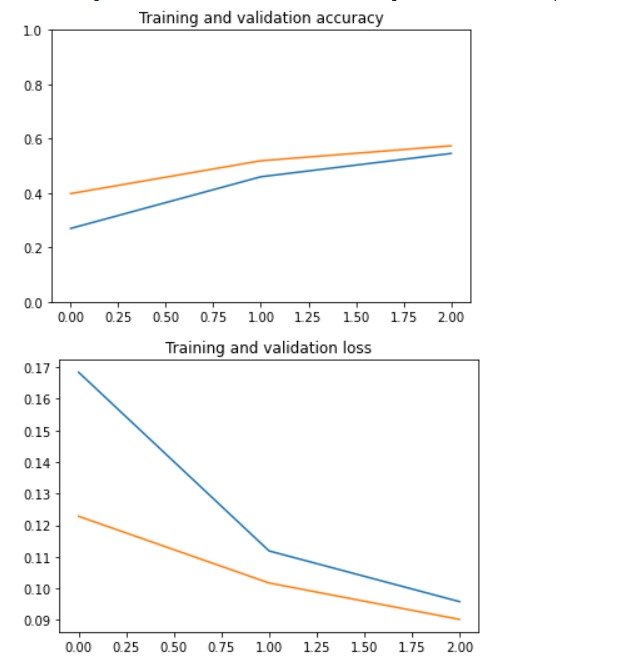
Description automatically generated

**GABOR**:-

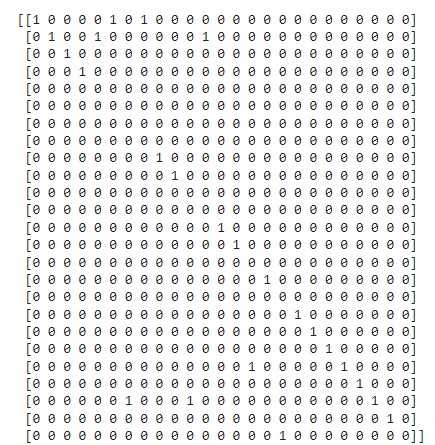
 Table

Description automatically generated

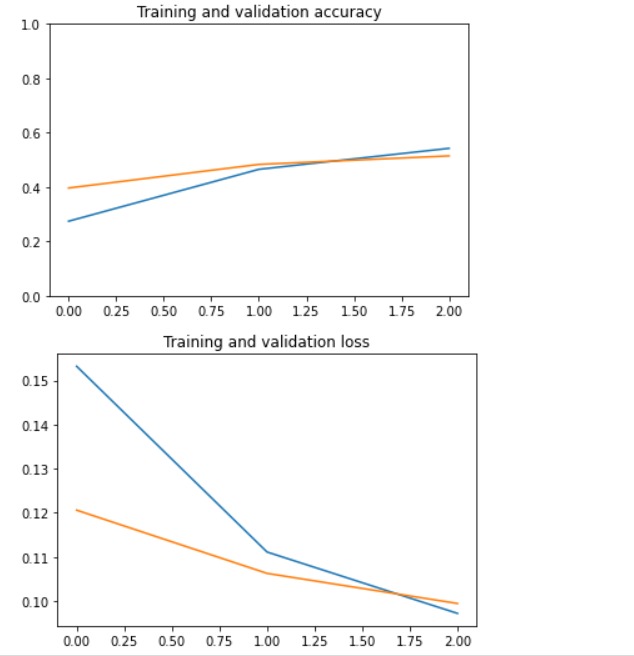
**SIFT**-



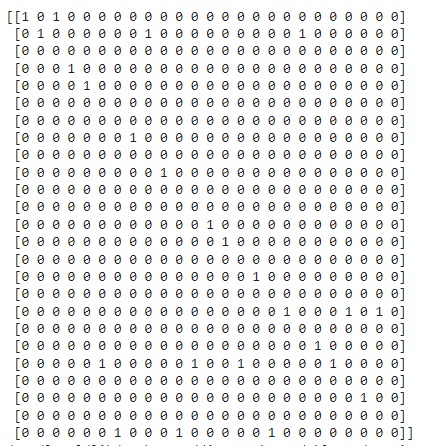
A picture containing text, receipt

Description automatically generated 

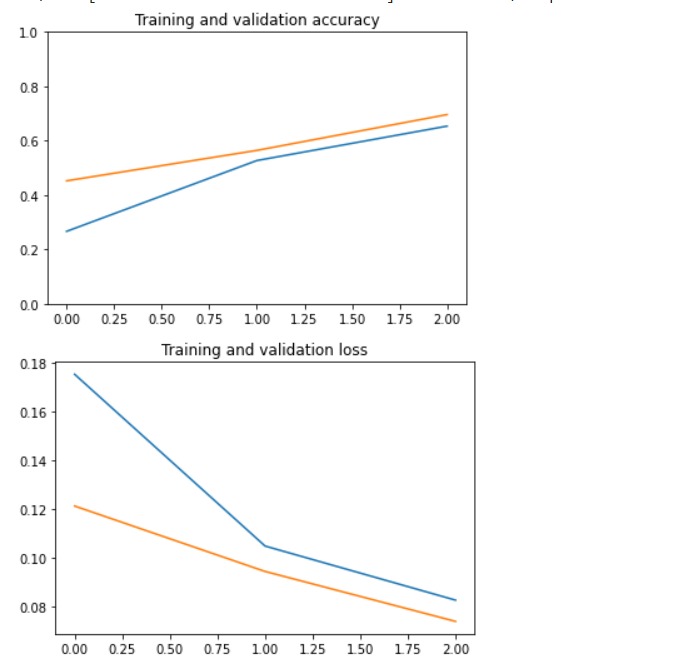
**SPATICAL** Filters:-



Table

Description automatically generated 

**SOBEL FIlters:**



Table

Description automatically generated 