1. Data Pre-processing:

   We have obtained the database from kaggle, which is in the form of a .csv file. Each entry represented an image of 28 x 28 size. We have loaded the train and test files, and have been able to visualize the data i.e., conversion of the pixel values to image format.

   We have also looked into the distribution of data as the count of the number of entries of each class corresponding to the alphabet in sign language. As it turned out that there are an average of 1143 images in each class and all the classes have a sufficient number of training examples.

2. Pipeline:

   From the loaded data we have normalized the values of pixels so that everything falls within the range. We have splitted the training data, which are pixel values and the labels into x_train and y_train.

   The y_train labels is encoded using LabelBinarizer

   The data is reshaped so as to fit in the model we are using.

   Augmentation:

   We are using an image generator for the data augmentation. In which we are doing the following operations

   - 10 - Randomly rotate images in the range (degrees, 0 to 180)
   - 0.1 - Randomly zoom image
   - 0.1 - Randomly shift images horizontally (fraction of total width)
   - 0.1- Randomly shift images vertically (fraction of total height)
   - We are not doing the vertical and the horizontal flipping because the flipping leads to misclassification of the images as in sign language orientation is important.

   Modeling:

   We built a CNN from scratch using keras library. By using Sequential models in keras we are adding the layers such as convolution layer, max pool layer, batch normalization and dropout layers to build a CNN. These CNN layers are for the feature

extraction which we think has done a good job. After these layers we are using Dense and flatten layers to get the predicted output.
The results obtained from this model are:
Training loss: 0.0126 - Training accuracy: 0.9963
Validation loss: 0.1514 - Validation accuracy: 0.9480

We suspect that the model that we designed is undergoing overfitting as the training accuracy is very large.

3. (i) We will be looking into the matter of overfitting by using more layers and augmentation and also hyperparameter tuning.
(ii) We will be looking for more resources for the data set of sign language from the links that were shared and try to incorporate them in our project.
(iii) We will try to incorporate some OpenCV if possible so as to detect the real images of sign language.
(iv) We will try to build more models using CNN architecture by changing the layers and tuning the hyperparameters and try to get the best CNN model. We also try to incorporate other techniques like ensembles for getting better results.
(v) Atlast, the main goal for the final submission is to decrease the overfitting and to get a better accuracy of the prediction and to understand the CNN architecture building.

Contributions:

1. P V Damoadram : Data augmentation, model building
2. Muttana Jashraj: Data loading and visualization, model building