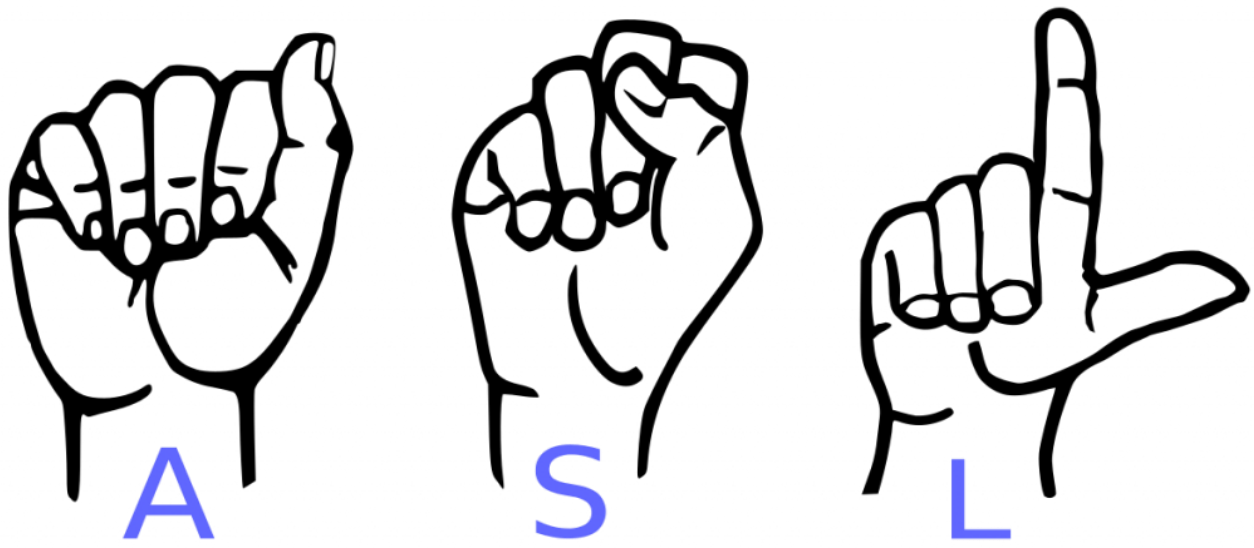
TECHNOCOLABS DATA SCIENCE INTERNSHIP

PROJECT REPORT

TITLE: ASL Recognition with Data Science and Machine Learning



AIM:

The main purpose of this project is to classify the images of ASL (American Sign Language) letters using Convolutional Neural Network.

ABSTRACT:

Hand gesture is one of the methods used in sign language for non verbal communication. Being a significant part in nonverbal communication, hand gestures are playing a key role in our daily life. A lot of recent progress has been made towards developing computer vision systems that translate sign language to spoken language.
This project helps in recognizing the hand gestures of a person and helps to communicate with the person in much effective manner.

INTRODUCTION:

American Sign Language (ASL) is the primary language used by many deaf individuals in North America, and it is also used by hard-of-hearing and hearing individuals. The language is as rich as spoken languages and employs signs made with the hand, along with facial gestures and bodily postures.

The result obtained after completion of the project is to evaluate how fast and accurately the model classifies the hand gestures.

OVERVIEW:

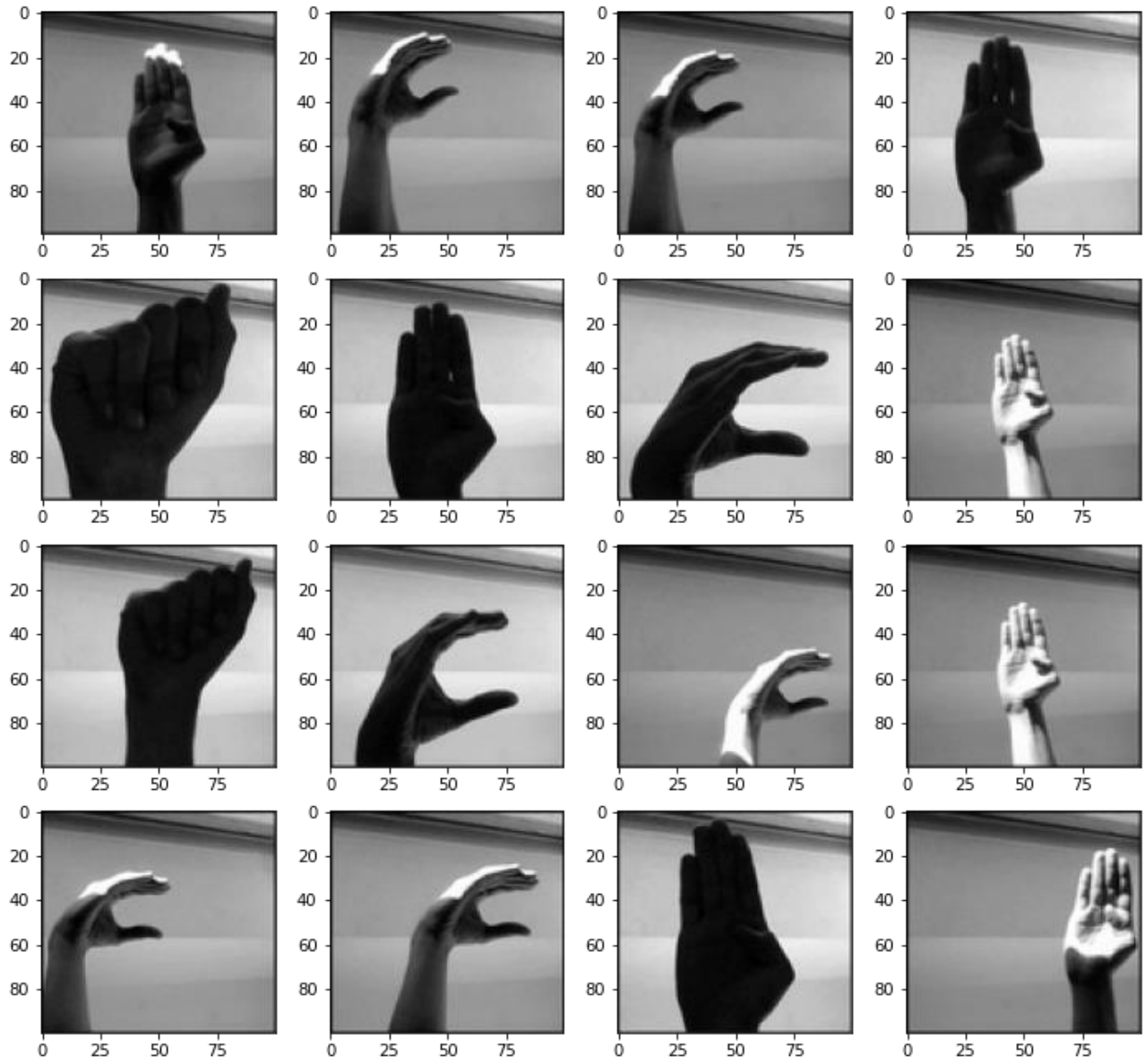
- ❑ Examine and Visualize the dataset.
- ❑ Building and Training the model.
- ❑ Visualize the mistakes

DATASET:

The dataset consists of 3000 images of hand gestures for three different alphabets. There are 1000 images for each hand gesture. The images are in jpg format. Each image is of size 100 x 100.

EXAMINE AND VISUALIZE THE DATA:

- ➢ The images present in jpg format need to be retrieved from the folder and need to be converted as a numpy array.
- ➢ The images were retrieved from the folder using opencv in a grayscale format and were stored as numpy arrays.
- ➢ The images of the hand gestures for the three alphabets “A”, “B”, “C” were plotted using matplotlib.

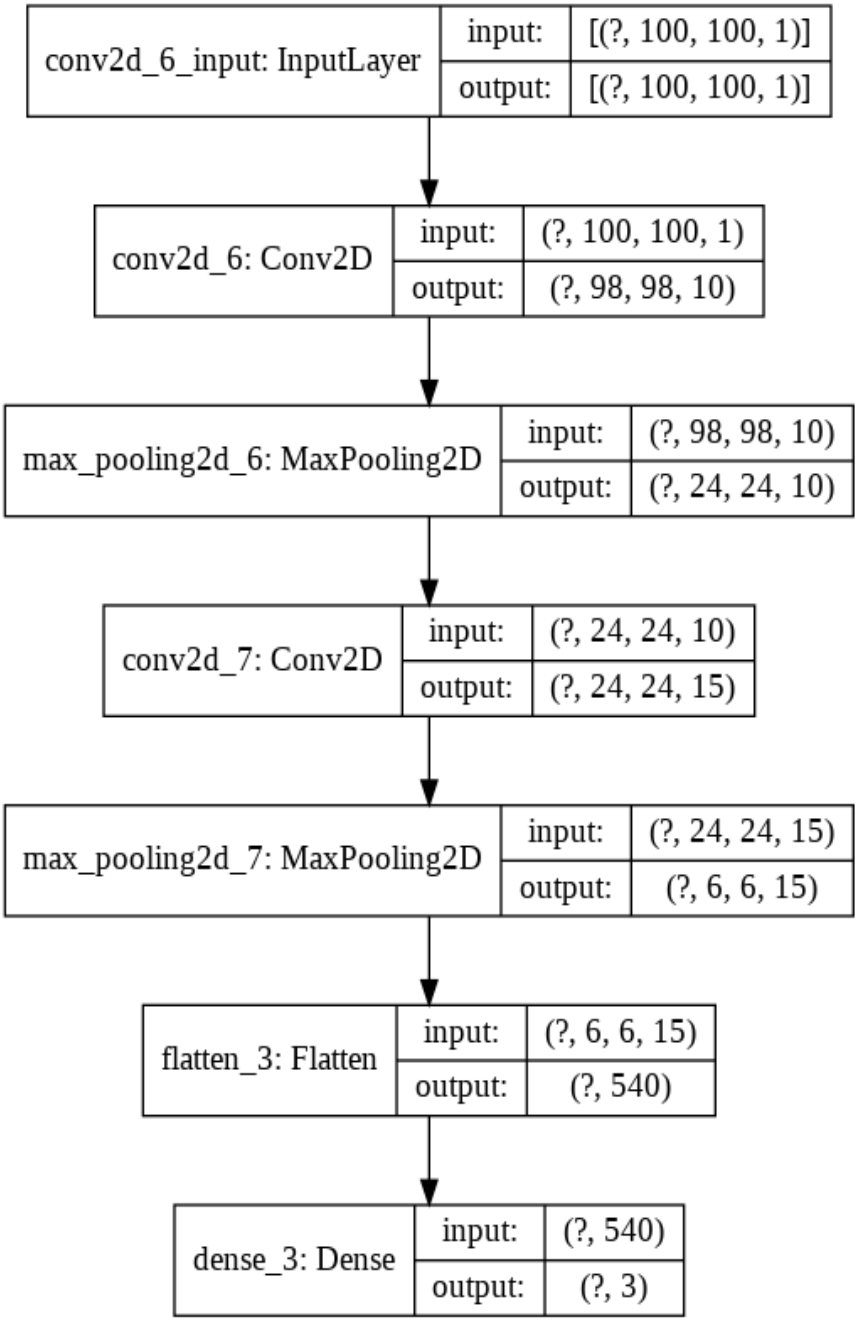


- The categorical labels are encoded using one hot encoding method.
- The images and labels are then shuffled before passing it for the training process. The images and labels can be shuffled using the shuffle function from utils class in sklearn library.
- The images and labels are then divided into train and test set so that the model can be trained on the train set and then can be evaluated on the test set.
- The shuffled images can be visualized by using the same method that we used for visualizing the three alphabets.

BUILD AND TRAIN THE MODEL:

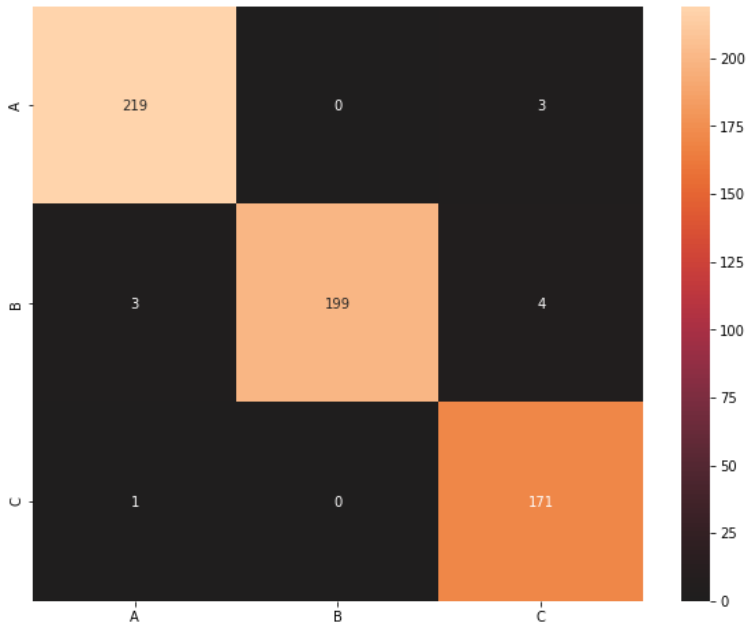
- The model consists of 7 layers.
- The first layer consists of a Convolutional 2D layer with 10 filters and kernel size of 3 followed by a Max Pooling layer of windows size 4 x 4.
- The third layer consists of a Convolutional 2D layer with 15 filters and kernel size of 5 followed by a Max Pooling layer of windows size 4 x 4.
- The next layer consists of a Flatten layer to collapse the spatial dimensions of the input into the channel dimension.
- The last layer is a dense layer of three neurons with softmax activation function.

- The model was trained on 2400 images consisting images for all the three alphabets. Model was trained for 15 epochs with rmsprop as optimizer and categorical_crossentropy as loss function.
- The Convolution Neural Network architecture can be seen below.


conv2d_6_input: InputLayer
input: [(?, 100, 100, 1)]
output: [(?, 100, 100, 1)]
conv2d_6: Conv2D
input: (?, 100, 100, 1)
output: (?, 98, 98, 10)
max_pooling2d_6: MaxPooling2D
input: (?, 98, 98, 10)
output: (?, 24, 24, 10)
conv2d_7: Conv2D
input: (?, 24, 24, 10)
output: (?, 24, 24, 15)
max_pooling2d_7: MaxPooling2D
input: (?, 24, 24, 15)
output: (?, 6, 6, 15)
flatten_3: Flatten
input: (?, 6, 6, 15)
output: (?, 540)
dense_3: Dense
input: (?, 540)
output: (?, 3)

VISUALIZE THE MISTAKES:

- The trained model should be evaluated on the test data set in order to measure the performance of our model.
- To check how many images were accurately classified by the model, we can plot a confusion matrix and look at how many images were properly classified and how many were misclassified by the model.



- Out of 600 test images the model classified 589 images correctly. i.e. the model classifies the images 98.6% accurately on an unseen data.