(Batch 2019-2023 Fifth Semester)

Machine Learning Course Project (MLCP) - 2021

Team Number													1	1
1	Name :	S	O	N	A		S		S	H	E	T	T	Y
2	Name :	V	I	S	H	W	A	S		R				
		B	A	N	A	G	A	R						
3	Name :	S	P	O	O	R	T	I		A	N	G	A	D	I
4	Name :	S	U	R	A	J		B	A	D	D	I			

USN:											
0	1	F	E	1	9	B	E	C	2	5	7
USN:											
0	1	F	E	1	9	B	E	C	2	5	8
USN:											
0	1	F	E	1	9	B	E	C	2	5	9
USN:											
0	1	F	E	1	9	B	E	C	2	6	8

Module: 1

Project Title: **SIGN BOARD RECOGNITION FOR WARNING, STOP AND YIELD CLASSES**

Problem statement: To implement sign board recognition for warning, stop and yield classes using deep learning, machine learning and self-supervised learning algorithms.

Summary:

Dataset details:

Number of data samples collected: 6050

Number of classes (if data is for classification): 3

Any other specific details about the datasets: All the data samples collected are in the ratio of 4:3. “Rule of thirds” (an image is broken down into thirds, so that we have 9 parts in total) is followed throughout the entire process.

Describe your dataset in a paragraph:

The dataset collected is captured uniformly in 4:3 ratio. We have avoided zooming in and reduced the noise as much as possible. “Rule of thirds” (an image is broken down into thirds, so that we have 9 parts in total) is followed throughout the entire process. The background maintained is either green or white so as to help with easier classification. The dataset captured for each class on an average is 2000. The clarity and focus on the subject is also an important attribute of our dataset collection.

Type of problem: Classification

Type of Machine Learning algorithm/ approach: Deep learning and Machine learning

Number of classes with appropriate labels: 3 (‘STOP’, ‘WARNING’, ‘YIELD’)

Number of features with appropriate labels: 3 (Stop board samples, Warning board samples and Yield board samples)

Model used for classification: Convolutional Neural Network (One of the types of Deep Learning)

(Batch 2019-2023 Fifth Semester)

Machine Learning Course Project (MLCP) - 2021

Results and Discussion:

Input Data: 6050 Images (STOP – 1824 images, WARNING – 2985, YIELD – 1241)

First, we uploaded the data into Google Drive. Then we mounted the drive onto Google Colab and accessed the images. Then we executed the code.

Expected Output:

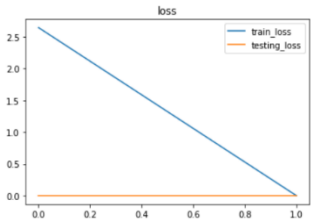
We have used Keras, TensorFlow, Numpy and Matplotlib libraries for better and easier implementation. We have used a pre-trained model for imagenet dataset. Instead of imagenet dataset, we have used the dataset collected by us. The algorithm should be able to classify the number of classes. The width and height of the image is specified to be 224. Batch size (every time it iterates, the number of images taken at a time) is taken as 64. The dataset needs to be trained, cross-validated and tested. First, the dataset is trained and tested in the ratio of 8:2. The colour mode is RGB. The base model used is VGG19 architecture. There are 3 layers in the neural network i.e, input, hidden and output layer. The type of deep learning used is CNN. The number of convolution layers is 16. Pooling is done 5 times. There are 2 dense layers. 'Softmax' (flatten layer) is used. The number of hidden layers is 24. Loss is categorical crossentropy. The metric used is categorical accuracy. Optimiser is adam. The GPU allocated is Tesla K80. The number of epochs is 3. Test loss is zero and test accuracy is 0.0 and 1.0 respectively.

Tools/Programming used: Google colab, Imagenet, Google Drive, VGG19 Model

Accuracy Achieved: 100%

Visualization of results:

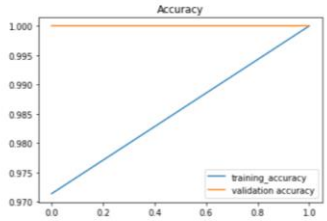
```
[ ] plt.plot(history.history['loss'],label = 'train_loss')
    plt.plot(history.history['val_loss'], label = 'testing_loss')
    plt.title('loss')
    plt.legend()
    plt.show()
```



(Batch 2019-2023 Fifth Semester)

Machine Learning Course Project (MLCP) - 2021

```
[ ] plt.plot(history.history['accuracy'], label='training_accuracy')
    plt.plot(history.history['val_accuracy'], label='validation accuracy')
    plt.title('Accuracy')
    plt.legend()
    plt.show()
```




```
[ ] from matplotlib.pyplot import imread
    from matplotlib.pyplot import imshow
    from keras.preprocessing.image import load_img
    from keras.preprocessing.image import img_to_array

    img_path = '/content/drive/MyDrive/IMG_20211112_111509_Burst20.jpg'
```

```
[ ] img = load_img(img_path, target_size=(224, 224))

    x = img_to_array(img)
    x = np.expand_dims(x, axis=0)

    print('Input image shape:', x.shape)

    my_image = imread(img_path)
    imshow(my_image)
```

```
Input image shape: (1, 224, 224, 3)
<matplotlib.image.AxesImage at 0x7fada457eed0>
```


```
[ ] print(model.predict(x))
```

```
[[1. 0. 0.]]
```

```
[ ] from matplotlib.pyplot import imread
    from matplotlib.pyplot import imshow
    from keras.preprocessing.image import load_img
    from keras.preprocessing.image import img_to_array

    img_path = '/content/drive/MyDrive/IMG_20211128_133428_005.jpg'

    img = load_img(img_path, target_size=(224, 224))

    x = img_to_array(img)
    x = np.expand_dims(x, axis=0)

    print('Input image shape:', x.shape)

    my_image = imread(img_path)
    imshow(my_image)
```

```
Input image shape: (1, 224, 224, 3)
<matplotlib.image.AxesImage at 0x7fada4511650>
```

(Batch 2019-2023 Fifth Semester)

Machine Learning Course Project (MLCP) - 2021

```
[ ] print(model.predict(x))

    [[0. 1. 0.]]
```

```
[ ] from matplotlib.pyplot import imread
    from matplotlib.pyplot import imshow
    from keras.preprocessing.image import load_img
    from keras.preprocessing.image import img_to_array

    img_path = '/content/drive/MyDrive/IMG_20211112_114803_Burst07.jpg'
```

```
[ ] img = load_img(img_path, target_size=(224, 224))

    x = img_to_array(img)
    x = np.expand_dims(x, axis=0)

    print('Input image shape:', x.shape)

    my_image = imread(img_path)
    imshow(my_image)

    Input image shape: (1, 224, 224, 3)
    <matplotlib.image.AxesImage at 0x7fada43af450>
```


```
[ ] print(model.predict(x))

    [[1.3643496e-35 6.1347647e-09 1.0000000e+00]]
```

Challenges faced to address the problem:

- RAM crashing in Google colab every time while training the dataset.
- Uploading the non-pre-processed data to drive and running the code.
- Taking nearly 1-2 hours just to run a piece of code.
- Choosing the appropriate architecture for our problem statement.

Online Link of the project: DL_ALGORITHM-Team11

(https://colab.research.google.com/drive/1DNV3AVDpLEwUXGtiHcgPhENV7-F0AApW?authuser=2)