Group4: Image Classification

Dataset: MIT Indoor Scenes

# About Dataset

Overall, data set contains 15620 images in the 67 folders, and each folder is having at least 100 images. Each folder represents the same class.

Name of the folders are considered as labels.

Folder names -

'airport\_inside', 'artstudio', 'auditorium', 'bakery', 'bar', 'bathroom', 'bedroom', 'bookstore', 'bowling', 'buffet', 'casino', 'children\_room', 'church\_inside', 'classroom', 'cloister', 'closet', 'clothingstore', 'computerroom', 'concert\_hall', 'corridor', 'deli', 'dentaloffice', 'dining\_room', 'elevator', 'fastfood\_restaurant', 'florist', 'gameroom', 'garage', 'greenhouse', 'grocerystore', 'gym', 'hairsalon', 'hospitalroom', 'inside\_bus', 'inside\_subway', 'jewelleryshop', 'kindergarden', 'kitchen', 'laboratorywet', 'laundromat', 'library', 'livingroom', 'lobby', 'locker\_room', 'mall', 'meeting\_room', 'movietheater', 'museum', 'nursery', 'office', 'operating\_room', 'pantry', 'poolinside', 'prisoncell', 'restaurant', 'restaurant\_kitchen', 'shoeshop', 'stairscase', 'studiomusic', 'subway', 'toystore', 'trainstation', 'tv\_studio', 'videostore', 'waitingroom', 'warehouse', 'winecellar’

Type of Images are jpg and count of images in the folder are shown below diagram.

Chart

Description automatically generated

# Loading & Preprocessing

Dataset was loaded using the flow\_from\_directory(directory) function which is in ImageDataGenerator from tf.keras.preprocessing.image API.

ImageDataGenerator - Generate batches of tensor image data with real-time data augmentation.

* While loading, images are resized to 224,224 i.e., width and height respectively
* All images are considered with original colour ‘RGB’
* As processing all the pixel values are normalized between 0 and 1 using rescale attribute in ImageDataGenerator

Text

Description automatically generated

# Architecture

Diagram, schematic

Description automatically generated

Model:

Text

Description automatically generated

# Challenges Faced

1. One of the initial challenges faced were scanning the structures of the files while importing the dataset while using Google Colab.



1. Another challenge faced was in the middle of the model training, encountered the following error. This is because of the few images in different subfolders.

InvalidArgumentError Traceback (most recent call last)

in 1 epochs=10 ----> 2 history = model.fit( 3 train\_ds, 4 validation\_data=valid\_ds, 5 epochs=epochs

1 frames

[/usr/local/lib/python3.8/dist-packages/tensorflow/python/eager/execute.py](https://colab.research.google.com/drive/1QMMkcdvI9Mvg9qOymlmKZ4DJmAYEWIHA) in quick\_execute(op\_name, num\_outputs, inputs, attrs, ctx, name) 52 try: 53 ctx.ensure\_initialized() ---> 54 tensors = pywrap\_tfe.TFE\_Py\_Execute(ctx.\_handle, device\_name, op\_name, 55 inputs, attrs, num\_outputs) 56 except core.\_NotOkStatusException as e:

InvalidArgumentError: Graph execution error:

2 root error(s) found. (0) INVALID\_ARGUMENT: Input size should match (header\_size + row\_size \* abs\_height) but they differ by 2 [[{{node decode\_image/DecodeImage}}]] [[IteratorGetNext]] [[IteratorGetNext/\_2]] (1) INVALID\_ARGUMENT: Input size should match (header\_size + row\_size \* abs\_height) but they differ by 2 [[{{node decode\_image/DecodeImage}}]] [[IteratorGetNext]] 0 successful operations. 0 derived errors ignored. [Op:\_\_inference\_train\_function\_2121] '''

* However, this was not the case if we use the ImageDataGenerator

Text

Description automatically generated

1. Model training has been difficult for us since we started with collab and later shifter to jupyter since there was memory outage due to the more number of model parameters and epochs.

* We had trained the model using fit\_generator() with 20 epochs and the accuracy we got for the validation set was 9.13% and the loss we got on the validation set was 2.98.

Text

Description automatically generated

Chart, line chart

Description automatically generated

# Predictions

* We had created a separate dataset for predicting the classes with 30 images.

Graphical user interface

Description automatically generated with medium confidence

* The predictions we got using the model were 1 correct prediction form 20 images.

Graphical user interface, text

Description automatically generated

# Hyperparameter Tuning

* We Hypertuned the model to get the best parameters for the model. The neurons varies from 2048 to 5210 and the learning rate varies from 1e-2 to 1e-3.

Text

Description automatically generated with medium confidence

* The best parameters we achieved after Hypertuning the model were:

Graphical user interface, text, application

Description automatically generated

And the maximum accuracy we achieved on the validation set was 9.3%.

* After training the model for 50 epochs, the best accuracy we achieved on the validation set using the best parameters were:

A picture containing text

Description automatically generated