## **Project Report**

****

**Submitted by:**

Shazaib Sarwar **ID:** F2018065021

M. Faizan Ameen **ID:** F2018065022

Ahmad Raza **ID:** F2018065117

**Submitted to:**

Dr. Shiraz Naseer

**Subject:**

Deep Learning and Neural Networks

**Section:** W2

**Session:** 2018-2022

**University of Management and Technology**

C-II Johar Town Lahore

**Multiclass Classification and Prediction**

**Introduction**

Image classification is a fascinating deep learning project. Specifically, image classification comes under the computer vision project category. In this project, we will build a convolution neural network in Keras with python on a [multiclassimagedatasetairplanecar] dataset. First, we will explore our dataset, and then we will train our neural network using python and Keras. What is Image Classification The classification problem being to categorize all the pixels of a digital image into one of the defined classes. Image classification is the most critical use case in digital image analysis. Image classification is an application of both supervised classification and unsupervised classification. In supervised classification, we select samples for each target class. We train our neural network on these target class samples and then classify new samples. In unsupervised classification, we group the sample images into clusters of images having similar properties. Then, we classify each cluster into our intended classes.

**Literature Review**

Image Classification – Deep Learning Project in Python with Keras. They discussed the image classification paradigm for digital image analysis in this keras deep learning project. They also talk about supervised and uncontrolled image categorization. The CIFAR-10 dataset and its classes are then explained. Finally, we looked at how to use the CIFAR-10 dataset to build a convolutional neural network for image categorization. In this we found accuracy of 68%.

https://www.kaggle.com/freddymeriwether/airplanes-cars-ships-vgg16

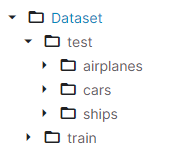
Airplanes, Cars & Ships VGG16: Similarly, some peoples solve this problem by using transfer learning. They use weights of some other model. vgg = VGG16(weights='ImageNet', include top=False,input\_shape=(224,224,3)).Downloadingdatafromhttps://storage.googleapis.com/tensorflow/keras-applications/vgg16/vgg16\_weights\_tf\_dim\_ordering\_tf\_kernels\_notop.h5 58892288/58889256 He uses loss function loss='categorical\_crossentropy', and optimizer as optimizer= SGD, in this he used Total parameters: 14,789,955 Trainable parameters: 75,267. After reaching Epoch 18/25 model reached at maximum accuracy and shows message Epoch 00018: validation accuracy did not improve from 0.99653. Basically it achieved maximum accuracy which is 99%.

https://www.kaggle.com/litaldavar/airplanes-cars-ships-classifier

Airplanes cars & ships classifier Similarly, some peoples solve this problem by using transfer learning. They use weights of some other model. vgg = VGG16(weights='ImageNet', include top=False, input\_shape = (224,224,3)). He uses loss function loss='categorical\_crossentropy', and optimizer as optimizer= SGD, in this he used Total parameters: 14,789,955 Trainable parameters: 75,267. After reaching Epoch 18/25 model reached at maximum accuracy and shows message Epoch 00018: validation accuracy did not improve from 0.99653. Basically it achieved maximum accuracy which is 99%. He also used weights of resnet =ResNet50(weights = 'ImageNet', include top=False, input\_shape = (224,224,3))

**Dataset Details:**

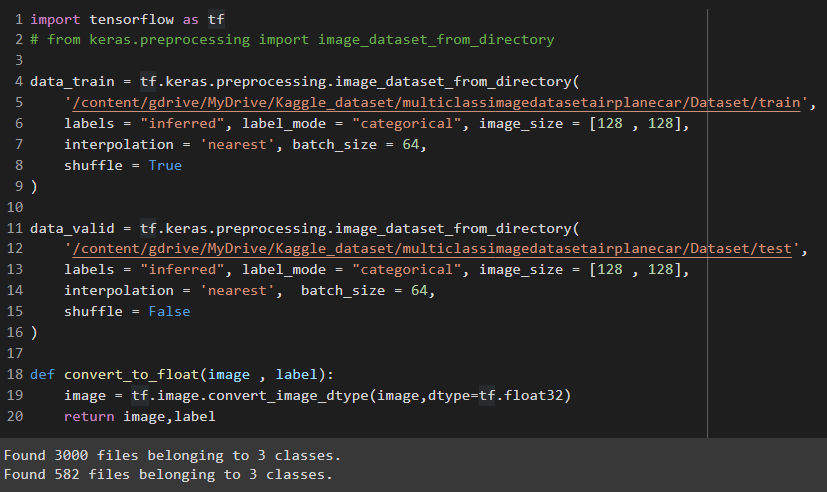
Link of our Dataset is = <https://www.kaggle.com/abtabm/multiclassimagedatasetairplanecar>

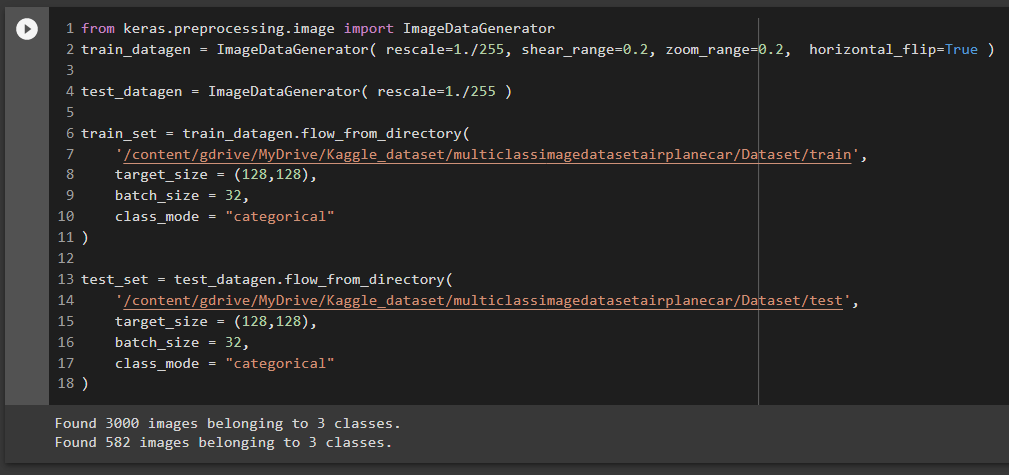
[multiclassimagedatasetairplanecar] is made up of three different image classes. This is essentially a well-known computer vision dataset. This dataset has been used in a variety of deep learning object recognition studies. This dataset has 3,000 photos separated into three target classes, each comprising images of the shape 1000\*1000.

* Airplane
* Car
* Ships

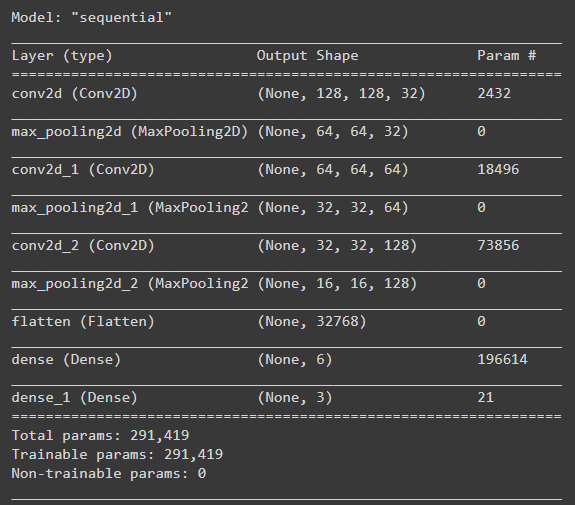
**Algorithm**

Pre-processing is a term used to describe procedures that work with images at the most basic level of abstraction, where the input and output are both intensity images. The intensity image is commonly represented by a matrix of image function values, and these iconic images are of the same type as the original data captured by the sensor (brightness’s). Although geometric transformations of images (e.g. rotation, scaling, translation) are classified among pre-processing methods here because similar techniques are used, the goal of pre-processing is to improve the image data by suppressing unwanted distortions or enhancing some image features important for further processing. In this project we Scaled our dataset images to by 1. /255 and resize them to 128\*128. Because for processing of 1000\*1000 will took lot of time to train. That’s why we resize our Dataset. In the we first separating our dataset in in Training dataset and Validation Dataset. And resizing them. After that, similarly I divided Dataset into Training Dataset and Testing Dataset.

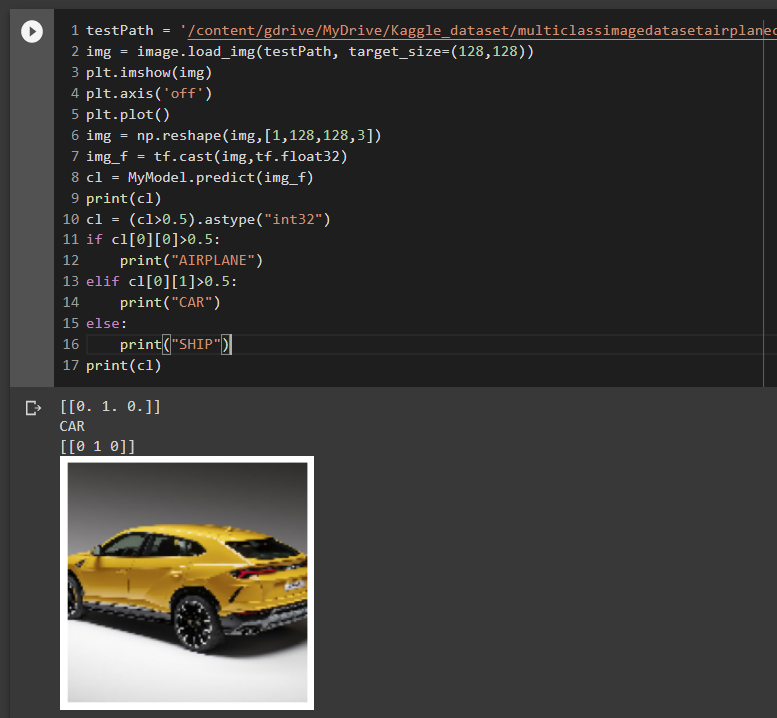


Similarly, I divided Dataset into Training Dataset and Testing Dataset.

Building ConvNet (CNN) Model

While building Convolutional Neural Network we used 3 Layers, on their feature map we used Maxpooling Layer to extract Task Specific Feature Extraction. Why we use Maxpooling layers because The fundamental benefit of pooling is that it aids in the extraction of sharp and smooth features. It's also done to cut down on computations and variation. Max-pooling aids in the extraction of higher features such as edges, points, and so on. And we Packed our Model to Sequential () container. At First Layer filters=32, kernel\_size=5, input\_shape= [128,128,3], activation="relu”, padding="SAME" At Second Layer filters=64, kernel\_size=3, activation="relu”, padding="SAME" and at Third Layer filters=128, kernel\_size=3, activation="relu”, padding="SAME" The Model Summary can be seen through the given image. In this we have Trainable parameters: 291,419. In this we use optimizer Adam and Loss function is categorical\_crossentropy.

**Results**

Model is trained to 30 epochs that yields accuracy: 0.9849 which is Essentially 98% and its validation accuracy is - accuracy: 0.9244 which is Essentially 92%. After that we saved our model in h5 file for letter use.

We can see model predicting

Right in one hot encoding.

