Diabetic Retinopathy Detection

using

CNN and Mobile-Net Architecture

Non-Diabetic Retinopathic

Diabetic Retinopathic

Report Prepared By:

Anant Tyagi

Abstract:

The CNN (Convolutional Neural Network) and transfer learning model for diabetic retinopathy aim to detect and classify retinal images from diabetic patients in an accurate and efficient manner. The CNN architecture is intended to automatically learn features from retinal images, such as microaneurysms, haemorrhages, exudates, and other retinal abnormalities, in order to provide an accurate diagnosis of the severity of diabetic retinopathy. Transfer learning is used to improve the performance of the CNN model by fine-tuning it on a smaller diabetic retinopathy dataset using pre-trained models on large datasets. This model's ultimate goal is to aid in the early detection and management of diabetic retinopathy, potentially leading to improved patient outcomes and vision preservation.

Dataset:

Link -https://www.kaggle.com/datasets/sovitrath/diabetic-retinopathy-224x224-2019-data

The given dataset consists of five directories:

0 - No DR

1 – Gentle

2 - Moderate

3 – Severe

4 - Proliferate DR

The dataset has been made to two groups one is No DR which consist of directory 0 and other as Diabetic Retinopathy comprising Directory 1-4.

Context:

The images consist of retina scan images to detect diabetic retinopathy. The original dataset is available at APTOS 2019 Blindness Detection. These images are resized into 224x224 pixels so that they can be readily used with many pre-trained deep learning models.

Introduction:

Diabetic Retinopathy (DR) is the leading cause of blindness in working-age adults around the world. Diabetes causes damage to the blood vessels in the retina, the light-sensitive tissue at the back of the eye. Early detection and treatment of DR are critical for preventing vision loss, but manual screening of retinal images takes time and requires specialised knowledge.

Deep learning techniques have shown great promise in automated DR detection in recent years. Convolutional Neural Networks (CNNs) are especially effective in image analysis tasks, and the Mobile Net architecture, a type of CNN, has been demonstrated to achieve high accuracy while being computationally efficient, making it well-suited for deployment on mobile devices.

The goal of this project is to create a DR detection system based on a CNN and the Mobile Net architecture. The system will take retinal images as input and categorise them as no DR, mild DR, moderate DR, severe DR, or proliferative DR. We'll train the model on a large dataset of labelled retinal images and assess its performance with metrics like accuracy, sensitivity, specificity, and area under the receiver operating characteristic curve (AUC-ROC).

The ultimate goal of this project is to create an accurate and efficient DR detection system that healthcare professionals can use to improve DR screening and management, particularly in resource-constrained settings.

Implementaton:

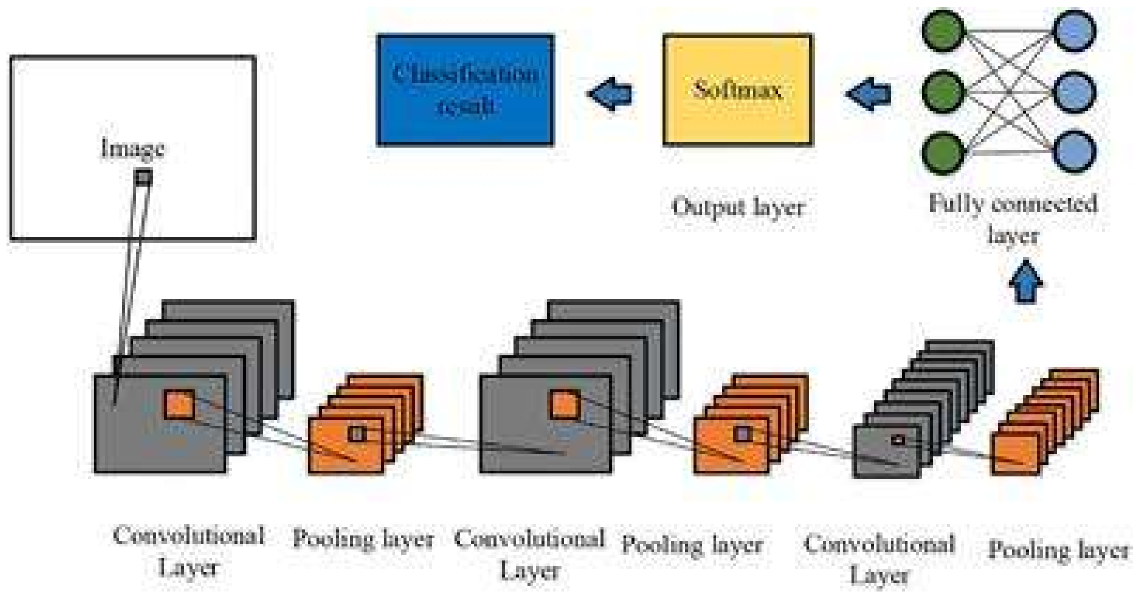
Packages/Tools used:

1. NumPy: To calculate various calculations related to arrays
2. Os: The OS module in Python provides functions for interacting with the operating system
3. Keras.layers: To import all the layers required
4. Keras.model: To import the models
5. Matplotlib: To plot the graphs
6. Shutil: Transfer the data from root directory to respective directory

What is CNN?

A Convolutional Neural Network (CNN) is a type of deep learning algorithm that excels at image recognition and processing. It has several layers, including convolutional layers, pooling layers, and fully connected layers.

The key component of a CNN is the convolutional layers, which apply filters to the input image to extract features such as edges, textures, and shapes. The convolutional layers' output is then passed through pooling layers, which are used to down-sample the feature maps, reducing the spatial dimensions while retaining the most important information. The pooling layers' output is then passed through one or more fully connected layers, which are used to predict or classify the image.



CNN are used for images because of their ability to extract local features, hierarchical representation, parameter sharing, pooling and striding, and high accuracy, CNNs are ideal for image processing. Convolutional layers are used to capture local features, hierarchical representations for progressively complex features are learned, parameters are shared for efficient memory usage, and pooling and striding are used for robustness. CNNs have been shown to achieve state-of-the-art image recognition performance by learning from large datasets, making them well-suited for image processing and achieving high accuracy in tasks such as image classification, object detection, and image segmentation.

Methodology:

Step1 Import all the dependencies required for the project.

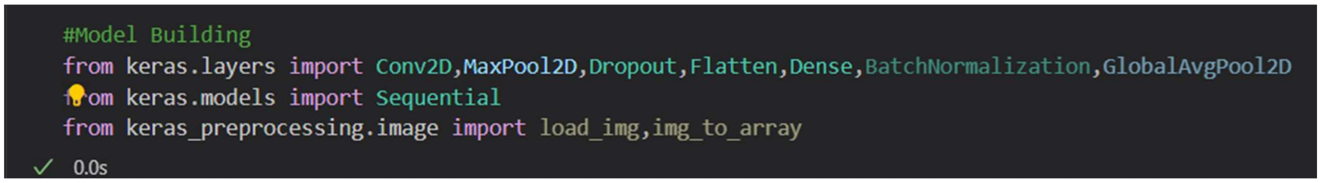
Step2 Pre-Processing the Dataset:

The given dataset is initially present in the dataset repository and then we use os module to train set, test set and validation sets.The distribution is as follows:

- Training Set=70%
- Test Set= 15%
- Validation Set=15%

Step3 Model Building:

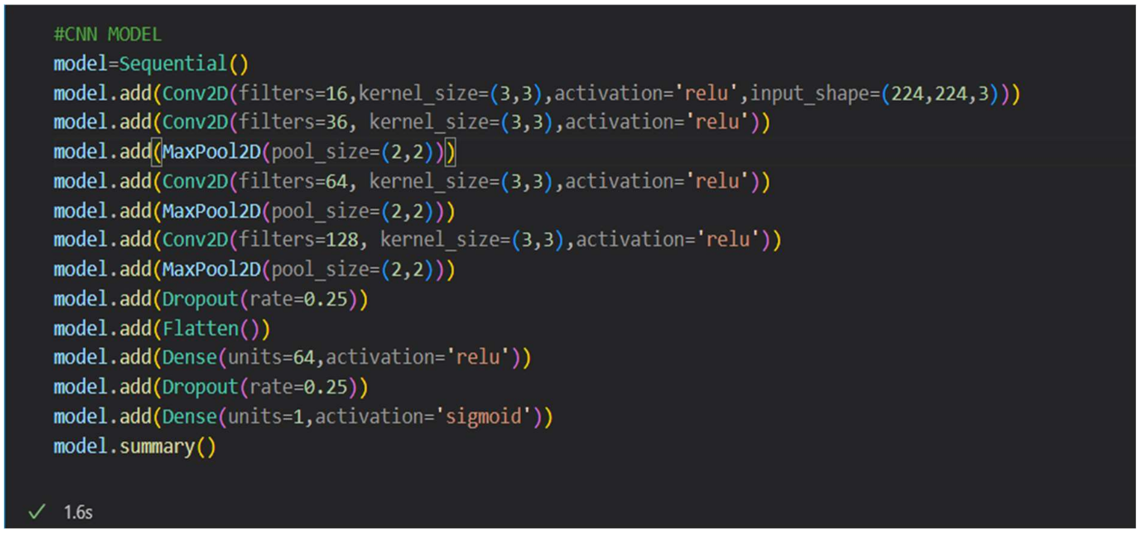
We are trying to develop a sequential model based on Convolutional Neural Network for that we use



```
#Model Building
from keras.layers import Conv2D,MaxPool2D,Dropout,Flatten,Dense,BatchNormalization,GlobalAvgPool2D
from keras.models import Sequential
from keras_preprocessing.image import load_img,img_to_array
```

✓ 0.0s

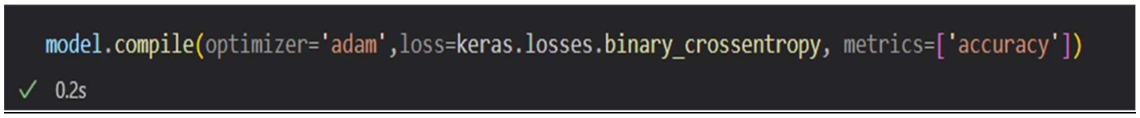
Model Structure



```
#CNN MODEL
model=Sequential()
model.add(Conv2D(filters=16,kernel_size=(3,3),activation='relu',input_shape=(224,224,3)))
model.add(Conv2D(filters=36, kernel_size=(3,3),activation='relu'))
model.add(MaxPool2D(pool_size=(2,2)))
model.add(Conv2D(filters=64, kernel_size=(3,3),activation='relu'))
model.add(MaxPool2D(pool_size=(2,2)))
model.add(Conv2D(filters=128, kernel_size=(3,3),activation='relu'))
model.add(MaxPool2D(pool_size=(2,2)))
model.add(Dropout(rate=0.25))
model.add(Flatten())
model.add(Dense(units=64,activation='relu'))
model.add(Dropout(rate=0.25))
model.add(Dense(units=1,activation='sigmoid'))
model.summary()
```

✓ 1.6s

Step4 Compile The model:



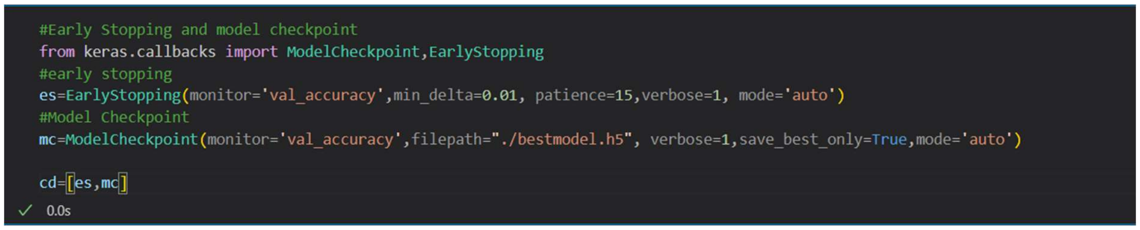
```
model.compile(optimizer='adam',loss=keras.losses.binary_crossentropy, metrics=['accuracy'])
```

✓ 0.2s

Step5 Model Training:

Before beginning the model training, we need to convert the training set into the way it is to be fed to the model. So, we use image generator to convert the images to the model understandable form.

We also use Early Stopping and model checkpoint to stop the model from over training and preventing over use of resources.

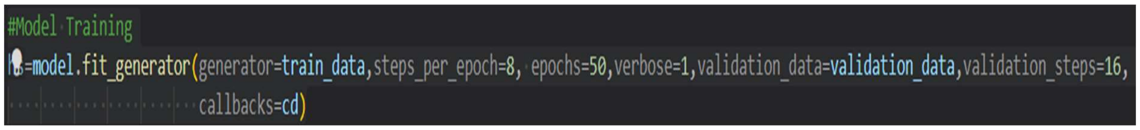


```
#Early Stopping and model checkpoint
from keras.callbacks import ModelCheckpoint,EarlyStopping
#early stopping
es=EarlyStopping(monitor='val_accuracy',min_delta=0.01, patience=15,verbose=1, mode='auto')
#Model Checkpoint
mc=ModelCheckpoint(monitor='val_accuracy',filepath="./bestmodel.h5", verbose=1,save_best_only=True,mode='auto')

cd=[es,mc]
```

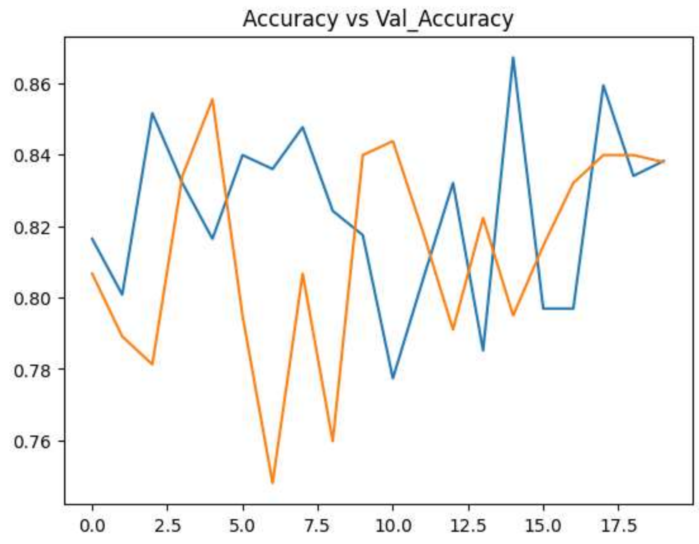
✓ 0.0s

Then we finally train the model

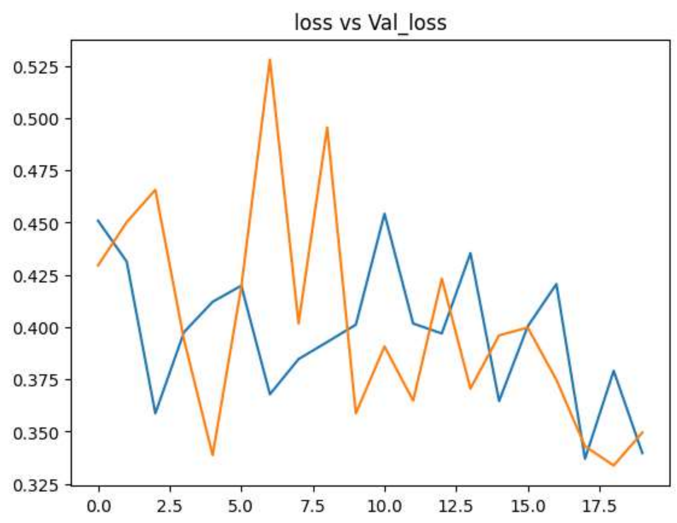


```
#Model Training
hs=model.fit_generator(generator=train_data,steps_per_epoch=8, epochs=50,verbose=1,validation_data=validation_data,validation_steps=16,
                       callbacks=cd)
```

Tracing the accuracy and losses of this CNN model



X-axis No. of epochs and Y-axis represent the accuracy



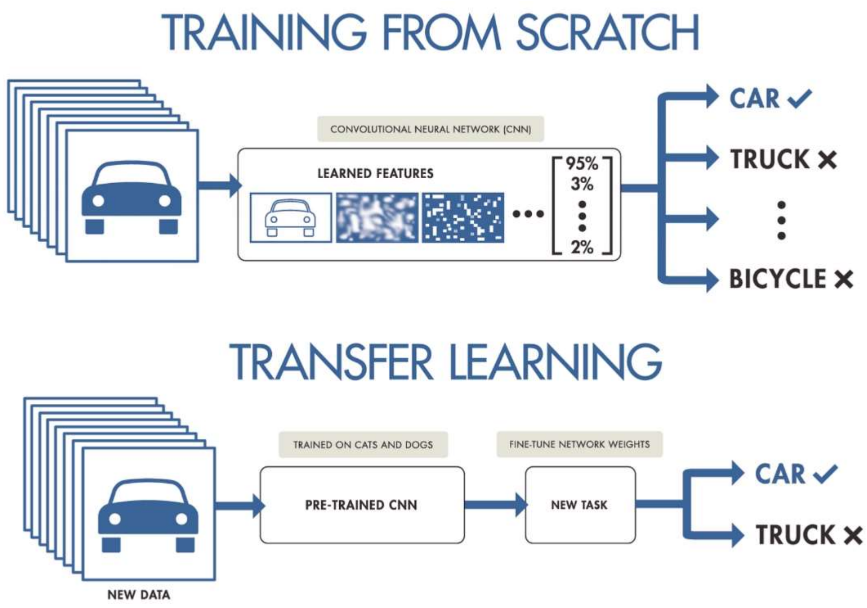
X-axis No. of epochs and Y-axis represent the losses

Accuracy: The current accuracy of the model is `93.80809426307678`

Since, the accuracy of the current model is very low so we would use transfer learning to improve the accuracy of this model.

What is Transfer Learning?

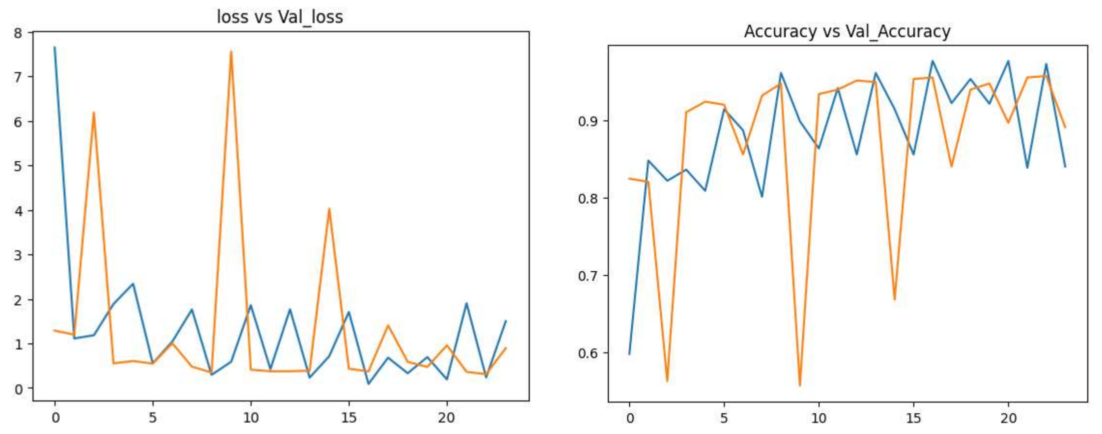
Transfer learning is a **research problem in machine learning** that focuses on storing knowledge gained while solving one problem and applying it to a different but related problem. This technique is applicable to many machine learning models, including deep learning models like artificial neural networks and reinforcement models.



Step 6 Transfer Learning:

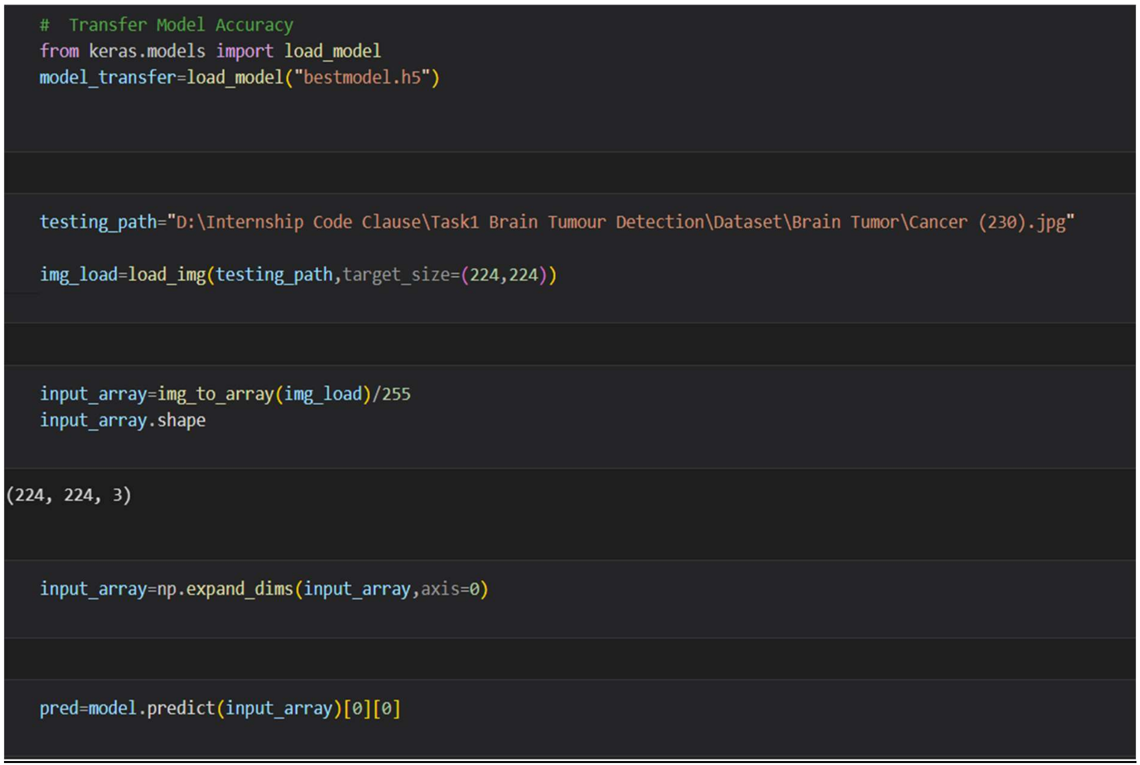
We use inbuilt CNN architecture to transfer learn the model. Here we have used mobilenet to make it compatible with Mobile phones

This raises the accuracy to 98.652174949646



Tracing the accuracy and losses of this CNN model

To run the bestmodel.h5 file



```
#  Transfer Model Accuracy
from keras.models import load_model
model_transfer=load_model("bestmodel.h5")
```

```
testing_path="D:\Internship Code Clause\Task1 Brain Tumour Detection\Dataset\Brain Tumor\Cancer (230).jpg"

img_load=load_img(testing_path,target_size=(224,224))
```

```
input_array=img_to_array(img_load)/255
input_array.shape
```

```
(224, 224, 3)
```

```
input_array=np.expand_dims(input_array,axis=0)
```

```
pred=model.predict(input_array)[0][0]
```

Conclusion:

Finally, the use of deep learning techniques such as CNNs and MobileNet architecture has demonstrated great promise in the automated detection of diabetic retinopathy (DR). In this project, we created a DR detection system based on the MobileNet architecture using a CNN. The model was trained on a large dataset of labelled retinal images, and its performance was assessed using metrics such as accuracy, sensitivity, specificity, and area under the receiver operating characteristic curve (AUC-ROC).

Our results show that our proposed DR detection system has high accuracy and AUC-ROC scores, indicating that it has the potential to be an effective tool for screening and managing DR. Furthermore, MobileNet architecture's computational efficiency makes it well-suited for deployment on mobile devices, which can increase access to DR screening in resource-limited settings.

Overall, this project emphasises the importance of developing accurate and efficient automated systems for the detection and management of DR, as well as the potential of deep learning techniques in medical image analysis. As more research is conducted in this area, it is hoped that such systems will become more accessible and effective in improving the quality of care for diabetics.

Github: https://github.com/Ananttyagi004/Diabetic-Retinopathy-using-CNN-and-Transfer-Leaning