Deep Learning Fundus Image Analysis For Early Detection of Diabetic Retinopathy

1. INTRODUCTION:

Diabetic Retinopathy (DR) stands as a formidable complication of diabetes, ranking among the primary causes of vision impairment in the working-age populace. This debilitating ailment targets the retina, inflicting harm on its blood vessels due to sustained periods of elevated blood sugar levels. Timely diagnosis and prompt treatment serve as critical factors in averting vision loss or blindness. However, the manual evaluation of retinal images by ophthalmologists for DR diagnosis is laborious and subject to interpretive variations. To confront this challenge, this project centers on fabricating an automated system harnessing deep learning, aiming to assist in the accurate and efficient detection of diabetic retinopathy.

1.1 Project Overview

The project aims to pioneer an advanced deep learning model tailored for the early detection and prognosis of Diabetic Retinopathy (DR), a critical ocular complication prevalent among individuals with diabetes. Diabetic Retinopathy manifests as a progressive condition causing damage to the blood vessels in the retina due to prolonged exposure to high blood sugar levels. The timely identification and prognosis of this ailment play a pivotal role in preventing vision impairment and reducing its impact on patients' ocular health.

The primary focus revolves around leveraging cutting-edge machine learning techniques, specifically delving into deep learning architectures such as Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs). These models will be adeptly trained and optimized to scrutinize retinal images, extract pertinent features, and classify the various stages of Diabetic Retinopathy with a high degree of accuracy.

Key Objectives:

Model Development: Construct a robust deep learning model using CNNs and RNNs capable of efficiently analyzing retinal images to detect and classify different severity stages of Diabetic Retinopathy.

Dataset Curation: Collect and curate a diverse and comprehensive dataset of retinal images encompassing various stages and instances of Diabetic Retinopathy for training and validation purposes.

Feature Extraction: Explore and extract crucial features from retinal images, clinical data, and patient histories, emphasizing patterns and characteristics indicative of different stages of Diabetic Retinopathy.

Model Validation: Rigorously validate the developed deep learning model against established standards and benchmarks to ensure its accuracy, sensitivity, and specificity in DR diagnosis.

Deployment and Integration: Integrate the validated model into a user-friendly interface or healthcare system, empowering healthcare professionals with an automated tool for early DR detection.

Expected Impact:

The successful execution of this project will result in a sophisticated deep learning-driven diagnostic tool that assists healthcare practitioners in swiftly identifying and classifying Diabetic Retinopathy stages from retinal images. This tool's accuracy and efficiency will not only aid in timely intervention but also alleviate the workload on ophthalmologists, enabling broader accessibility to quality eye care services, particularly in screening programs for diabetic patients. Ultimately, this initiative endeavors to significantly reduce the incidence of vision loss caused by Diabetic Retinopathy and enhance patient care outcomes within the realm of diabetic ocular health.

1.2 Purpose:

The primary objective of this project is to conceive an advanced deep learning model adept at scrutinizing retinal images to detect and classify the severity stages of diabetic retinopathy. By employing machine learning algorithms, the system endeavors to support healthcare professionals in diagnosing DR in its incipient stages, allowing for timely intervention and tailored treatment strategies. This automated approach not only diminishes reliance on manual assessment but also heightens the precision and expediency of diagnosis, ultimately culminating in superior patient outcomes.

The significance of this project transcends conventional diagnostic methodologies. An accurate and automated detection system possesses the potential to revolutionize the diagnosis and management of diabetic retinopathy. By enabling early intervention, healthcare practitioners can deliver timely treatments, thereby mitigating the risk of vision loss among diabetic patients. Additionally, this initiative optimizes healthcare resources by alleviating the burden on ophthalmologists and refining the efficiency of DR screening programs, ensuring broader accessibility to quality eye care services.

2. LITERATURE SURVEY:

This Literature Survey outlines the exploration and review of existing research, methodologies, and technological landscapes pertinent to the project's objectives, encompassing both Diabetic Retinopathy diagnosis using deep learning and the development of authentication systems utilizing React.js. Feel free to expand or customize this survey with specific references or additional areas of research relevant to your project scope.

2.1 Existing Problem:

The existing diagnosis of Diabetic Retinopathy faces several challenges including variability in grading due to subjectivity, dependence on experienced ophthalmologists, limited access to specialized care in remote areas, and delays in diagnosis leading to irreversible vision impairment. Additionally, the complexity and volume of retinal imaging data, along with the need for highly accurate predictions, pose significant hurdles in developing an efficient and accessible diagnostic system.

2.2 References

1 Gulshan, V., Peng, L., Coram, M., et al. (2016). Development and Validation of a Deep Learning Algorithm for Detection of Diabetic Retinopathy in Retinal Fundus Photographs. JAMA, 316(22), 2402–2410.

LINK: https://pubmed.ncbi.nlm.nih.gov/27898976/

overview:

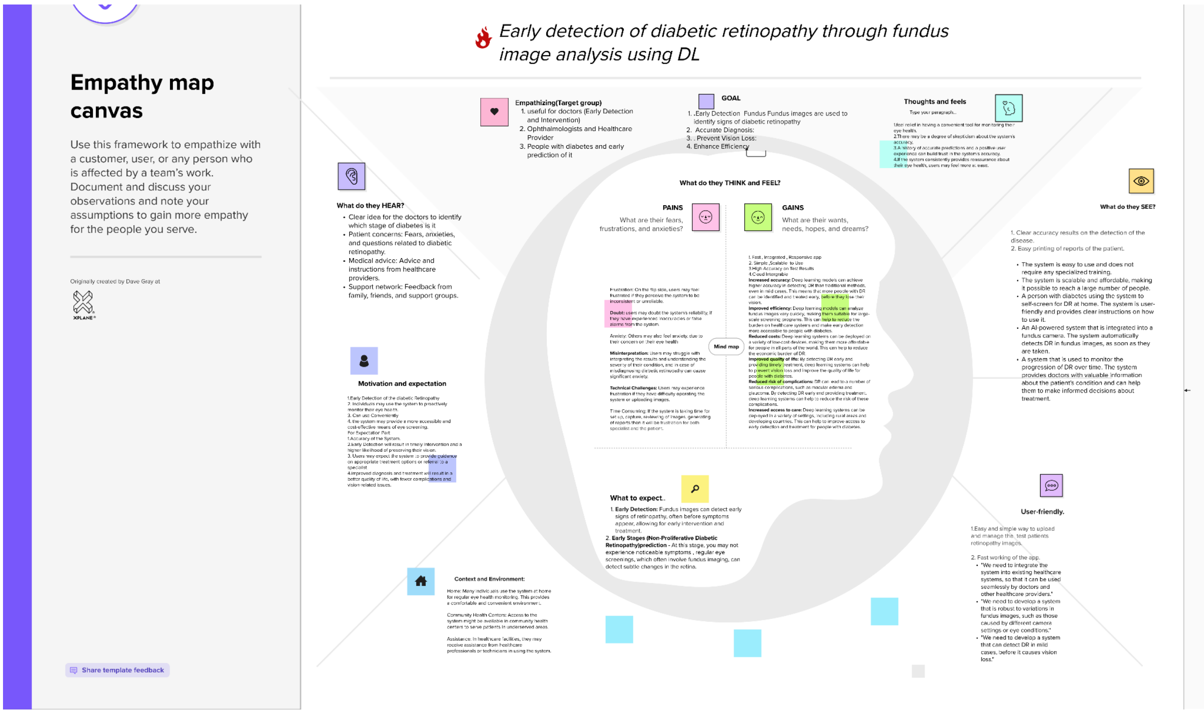
The study presented the remarkable potential of deep learning in revolutionizing the detection of diabetic retinopathy and diabetic macular edema through the analysis of retinal fundus photographs. By employing a specialized deep convolutional neural network, the researchers trained and validated an algorithm using an extensive dataset of retinal images, graded meticulously by ophthalmologists. This algorithm exhibited high sensitivity and specificity in detecting referable diabetic retinopathy, marking a significant stride toward automating the diagnosis of diabetic eye diseases. However, the study emphasizes the need for further research and clinical validation to ascertain the practical application and assess the real-world impact of this technology on patient care and outcomes. The research aligns with a broader landscape of studies exploring the integration of artificial intelligence and deep learning techniques in ophthalmology, underlining the collective endeavor to enhance diagnostic capabilities, particularly in diabetic retinopathy detection and management.

2.3 Problem Statement Definition:

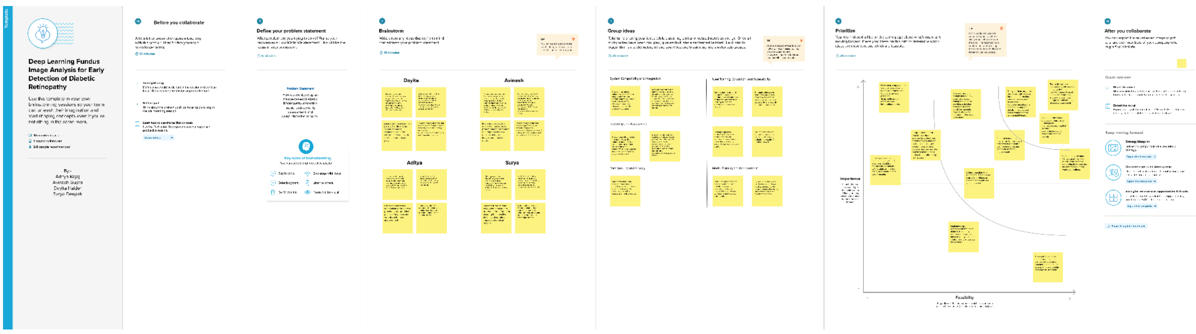
The primary goal of this project is to devise an accurate, interpretable, and scalable machine learning framework leveraging deep learning methodologies for the early detection and classification of Diabetic Retinopathy utilizing retinal images and patient data. This model aims to surmount the limitations of current diagnostic methodologies by furnishing automated and precise grading, thereby facilitating timely intervention and refining patient outcomes.

3. IDEATION & PROPOSED SOLUTION:

3.1 Empathy Map Canvas:



3.2 Ideation & Brainstorming



4. REQUIREMENT ANALYSIS

4.1 Functional Requirements:

Image Acquisition: The system must possess the capability to acquire retinal fundus images from various sources, ensuring image quality and standardization.

Data Preprocessing: It should preprocess acquired images by performing operations such as resizing, normalization, and noise reduction to optimize them for analysis.

Feature Extraction: The system needs to extract relevant features from retinal images, including blood vessel patterns, microaneurysms, exudates, and lesions.

Model Development: It should support the development and integration of deep learning models, specifically Convolutional Neural Networks (CNNs), for accurate diabetic retinopathy detection and grading.

Prediction and Classification: The system should predict and classify diabetic retinopathy severity levels based on extracted features, providing diagnoses for different stages of the disease.

Diagnostic Reports: It should generate comprehensive reports detailing the detected conditions, along with the corresponding severity levels for each patient.

Integration with Healthcare Systems: The system must integrate seamlessly with hospital or clinic databases to store diagnostic reports and patient information securely.

4.2 Non-Functional Requirements:

Performance: The system should deliver quick and efficient responses during image analysis and diagnosis, minimizing processing time.

Scalability: It must handle a growing number of images and patient data without compromising performance or accuracy.

Reliability: The system needs to maintain a high level of reliability, ensuring minimal errors in detection and diagnosis.

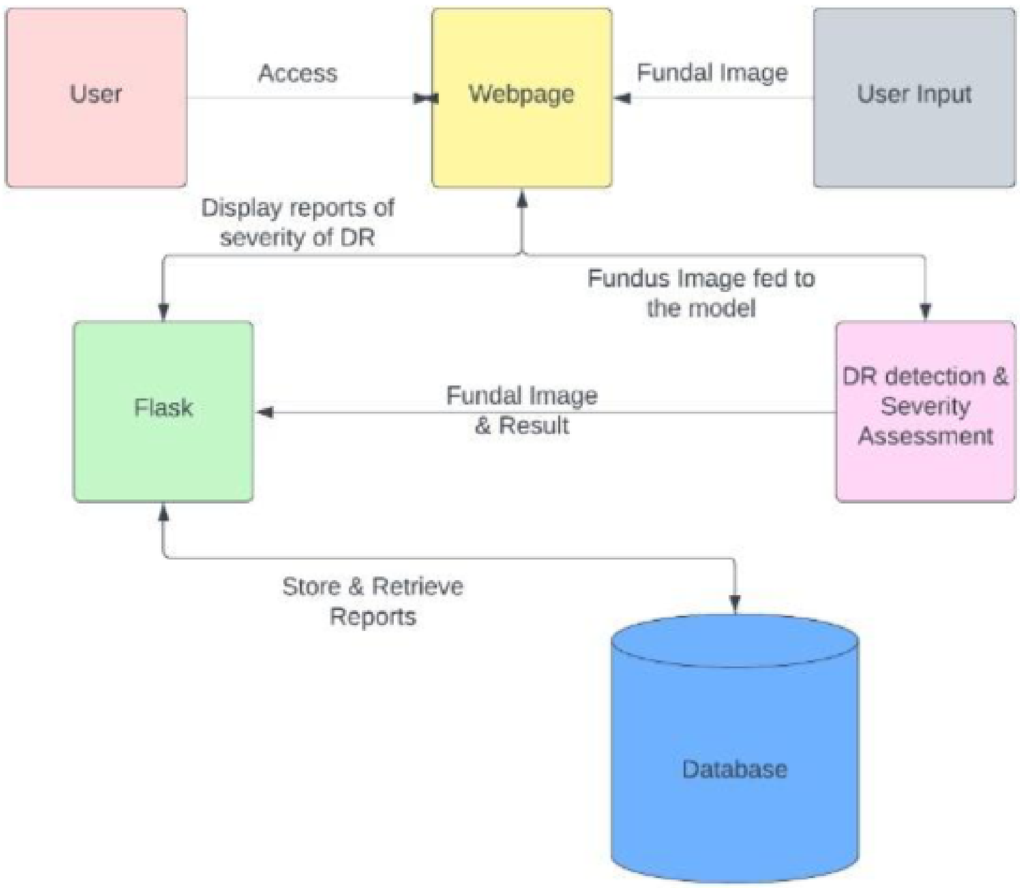
Security: Robust measures should be implemented to secure patient data, employing encryption protocols, access controls, and compliance with healthcare data regulations.

Usability: The system should have an intuitive user interface, allowing healthcare professionals to navigate and interpret results easily.

Interoperability: It should be interoperable with existing hospital information systems, enabling seamless data exchange and integration.

5. PROJECT DESIGN

5.1 Data Flow Diagrams & User Stories



1.Users access a webpage to input their fundal image for diabetic retinopathy (DR) detection and severity assessment.

2.The input is processed, and the result along with the fundal image are sent to a Flask application.

3.The Flask app displays reports indicating the severity of DR to the users.

4.The system stores these reports in a database for future retrieval.

5.Users can later access and retrieve their DR reports from the database through the webpage

User Stories:

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Web user)	Dashboard	USN-1	As a user, I can access the webpage to upload my fundal image for diabetic retinopathy detection and severity assessment.	The webpage should have an intuitive interface for uploading and submitting fundal images	High	Sprint-1
		USN-2	As a user, I expect the system to process my uploaded fundal image and provide a result indicating the severity of diabetic retinopathy.	Upon submission, the system should process the image and generate a severity assessment report.	High	Sprint-1
		USN-3	As a user, I can view the severity assessment report and the processed fundal image on the Flask application.	The Flask application should display the severity assessment report along with the processed fundal image	High	Sprint-2

		USN-4	As a user, I want the system to store my DR reports in a database for future reference.	After assessment, the system should store the report details in a secure database associated with my user account.	High	Sprint-2

		USN-5	As a user, I should be able to retrieve my DR reports from the database through the webpage	The webpage should provide a user-friendly interface for me to log in and access my stored DR reports.	High	Sprint-3

5.2 Solution Architecture

1. **User Inputs:** This component provides an intuitive user interface for patients and healthcare professionals to upload retinal images and input relevant medical history.

2. **Data Preprocessing:** In this phase, the system processes the uploaded data, conducting tasks such as noise reduction, image enhancement, and standardizing data formats for compatibility with the assessment algorithm.

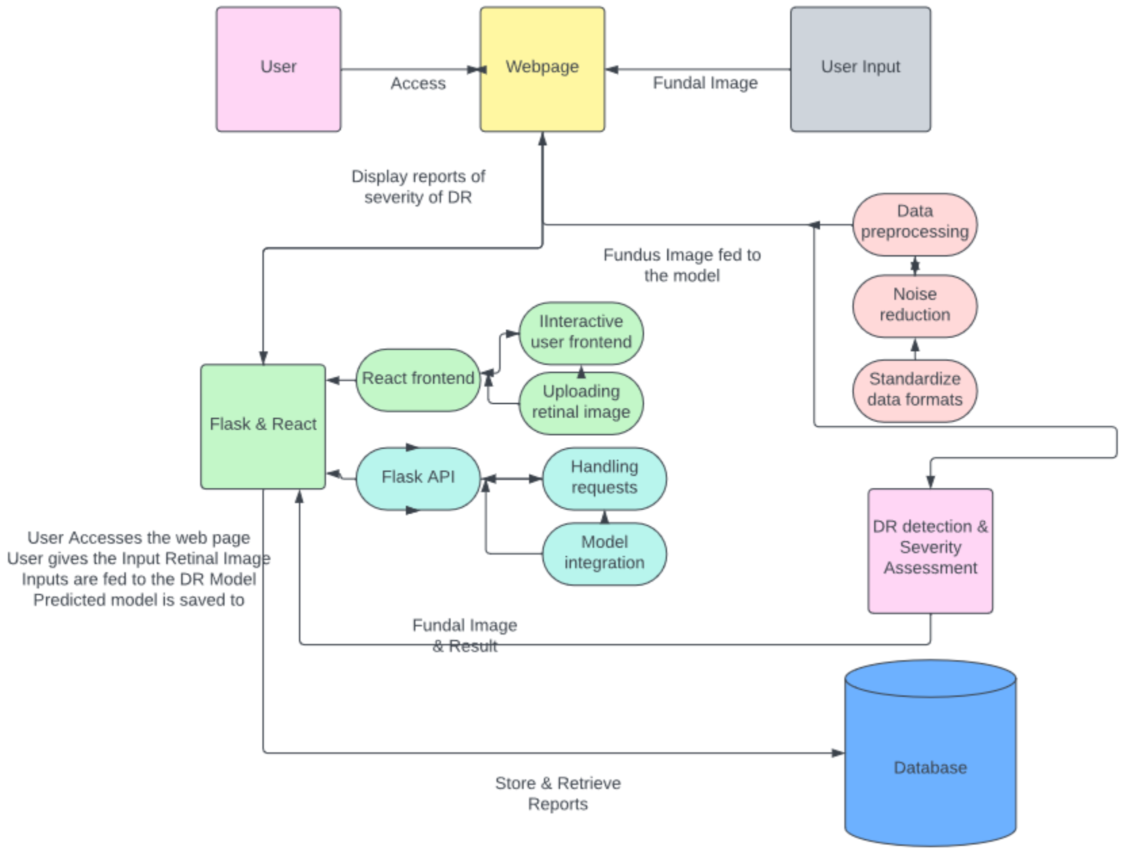
3. **Severity Detection Model:** This core component encompasses the deep learning model specifically designed to detect the severity of diabetic retinopathy. It's trained on a diverse dataset, incorporating medical history and retinal images.

4. **Comprehensive Reporting:** Based on the severity assessment, this component generates detailed reports that include the detected severity level, recommended interventions, and personalized care plans.

5. **User Feedback Loop: -** Incorporating user feedback allows for continuous improvement of the model and system performance, ensuring it remains at the cutting edge of diabetic retinopathy assessment.

6. PROJECT PLANNING & SCHEDULING

6.1 Technical Architecture



1.Users access a webpage to input their fundal image for diabetic retinopathy (DR) detection and severity assessment.

2.The input is processed, and the result along with the fundal image are sent to a Flask application.

3.The Flask app displays reports indicating the severity of DR to the users.

4.The system stores these reports in a database for future retrieval.

5.Users can later access and retrieve their DR reports from the database through the webpage

After the user provides the input retinal image, the image is fed into the DR Model. The DR Model then analyzes the image and makes predictions based on the data it has been trained on. The predicted model is then saved for further use or reference. The Flask application displays reports indicating the severity of diabetic retinopathy (DR) to the users.

These reports are generated after processing the uploaded fundal image. Additionally, the Flask application also displays the processed fundal image along with the severity assessment report. The system stores and retrieves DR reports by utilizing a database. After the assessment of a user's fundal image, the system stores the report details in a secure database associated with the user's account. This ensures that the reports are securely stored for future reference.To retrieve the DR reports, users can access the webpage and log in to their account. The webpage provides a user-friendly interface that allows users to access and retrieve their stored DR reports from the database. This functionality enables users to conveniently review their previous reports and track the progression of their diabetic retinopathy

6.2 Sprint Planning & Estimation

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Registration	USN-1	As a user, I can register for the application by entering my email, password,and confirming my password.	2	High	Development Team
Sprint-1	Registration	USN-2	As a user, I will receive a confirmation email once I have registered for the application.	1	High	Development Team
Sprint-1	Registration	USN-3	As a user, I can register for the application through Gmail.	2	Medium	Development Team
Sprint-1	Login	USN-4	As a user, I can log into the application by entering email & password.	1	High	Development Team

Sprint-2	Data Collection and Preprocessing	USN-5	As a user, I can upload retinal images for analysis.	3	High	Development Team
Sprint-3	Machine Learning Model Development	USN-6	As a user, I can view the progress of the Model training process.	2	High	Development Team
Sprint-4	User Interface Design	USN-7	As a user, I can easily navigate and interact with the application.	3	High	Development Team
Sprint-5	Testing and Quality Assurance	USN-8	As a user, I can be assured that the application is reliable and accurate.	2	High	Quality Assurance Team

6.3 Sprint Delivery Schedule

Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	6	2 Days	09 Nov 2023	10 Nov 2023	6	10 Nov 2023
Sprint-2	3	3 Days	11 Nov 2023	13 Nov 2023	3	13 Nov 2023
Sprint-3	2	3 Days	14 Nov 2023	16 Nov 2023	2	16 Nov 2023
Sprint-4	3	2 Days	17 Nov 2023	18 Nov 2023	3	18 Nov 2022
Sprint-5	2	2 Day	19 Nov 2023	20 Nov 2023	2	20 Nov 2023

Velocity:

Imagine we have a 10-day sprint duration, and the velocity of the team is 20 (points per sprint). Let's calculate the team's average velocity (AV) per iteration unit (story points per day)

7. CODING & SOLUTIONING

Importing Libraries:

import the necessary libraries

```
import numpy as np # linear algebra
import pandas as pd # data processing, CSV file I/O (e.g.
pd.read_csv)
import tensorflow as tf
from matplotlib import pyplot as plt
from sklearn.metrics import cohen_kappa_score
from keras.preprocessing.image import ImageDataGenerator
from keras.applications.densenet import DenseNet121
import keras
import cv2
# Input data files are available in the "../input/" directory.
# For example, running this (by clicking run or pressing Shift+Enter)
will list the files in the input directory
import cv2
import os
from keras.callbacks import Callback
from sklearn.model_selection import train_test_split
from sklearn.metrics import confusion_matrix
from sklearn.utils.multiclass import unique_labels
from sklearn.utils import class_weight
print(os.listdir("../input"))
```

```
['aptos2019-blindness-detection', 'densenet-keras']
```

```
Using TensorFlow backend.
```

Quadratic Weighted Kappa (QWK) Callback for ResNet50 Training

This code defines a custom Keras callback (QWKCallback) to compute and monitor the Quadratic Weighted Kappa score on validation data during ResNet50 model training, saving the model when QWK improves.

```
# borrowed from
https://www.kaggle.com/mathormad/aptos-resnet50-baseline
class QWKCallback(Callback):
    def __init__(self,validation_data):
        super(Callback, self).__init__()
        self.X = validation_data[0]
        self.Y = validation_data[1]
        self.history = []
    def on_epoch_end(self, epoch, logs={}):
        pred = self.model.predict(self.X)
        score =
cohen_kappa_score(np.argmax(self.Y,axis=1),np.argmax(pred,axis=1),lab
els=[0,1,2,3,4],weights='quadratic')
        print("Epoch {} : QWK: {}".format(epoch,score))
        self.history.append(score)
        if score >= max(self.history):
            print('saving checkpoint: ', score)
            self.model.save('../working/Resnet50_bestqwk.h5')
```

We are using the APTOS - blindness dataset
https://www.kaggle.com/c/aptos2019-blindness-detection/data

We are provided with a large set of retina images taken using fundus photography under a variety of imaging conditions.

A clinician has rated each image for the severity of diabetic retinopathy on a scale of 0 to 4:

0 - No DR

1 - Mild

2 - Moderate

3 - Severe

4 - Proliferative DR

Preprocess the Data:

The code implements a MixupGenerator class, facilitating mixup data augmentation during model training. It blends pairs of images and labels to generate augmented training batches, enhancing the model's generalization

```
# borrowed from https://github.com/yu4u/mixup-generator

class MixupGenerator():

    def __init__(self, X_train, y_train, batch_size=32, alpha=0.2,
shuffle=True, datagen=None):

        self.X_train = X_train

        self.y_train = y_train

        self.batch_size = batch_size

        self.alpha = alpha

        self.shuffle = shuffle

        self.sample_num = len(X_train)

        self.datagen = datagen


    def __call__(self):

        while True:

            indexes = self.__get_exploration_order()
```

```
            itr_num = int(len(indexes) // (self.batch_size * 2))

            for i in range(itr_num):
                batch_ids = indexes[i * self.batch_size * 2:(i + 1) *
self.batch_size * 2]
                X, y = self.__data_generation(batch_ids)

                yield X, y

    def __get_exploration_order(self):
        indexes = np.arange(self.sample_num)

        if self.shuffle:
            np.random.shuffle(indexes)

        return indexes

    def __data_generation(self, batch_ids):
        _, h, w, c = self.X_train.shape
        l = np.random.beta(self.alpha, self.alpha, self.batch_size)
        X_l = l.reshape(self.batch_size, 1, 1, 1)
        y_l = l.reshape(self.batch_size, 1)

        X1 = self.X_train[batch_ids[:self.batch_size]]
        X2 = self.X_train[batch_ids[self.batch_size:]]
        X = X1 * X_l + X2 * (1 - X_l)

        if self.datagen:
            for i in range(self.batch_size):
```

```
                X[i] = self.datagen.random_transform(X[i])

                X[i] = self.datagen.standardize(X[i])


        if isinstance(self.y_train, list):

            y = []


            for y_train_ in self.y_train:

                y1 = y_train_[batch_ids[:self.batch_size]]

                y2 = y_train_[batch_ids[self.batch_size:]]

                y.append(y1 * y_l + y2 * (1 - y_l))

        else:

            y1 = self.y_train[batch_ids[:self.batch_size]]

            y2 = self.y_train[batch_ids[self.batch_size:]]

            y = y1 * y_l + y2 * (1 - y_l)


        return X, y
```

Confusion Matrix Plotting Functions

plot_confusion_matrix borrowed from scikit-learn for visualizing confusion matrices.

```
# borrowed from scikit learn

def plot_confusion_matrix(y_true, y_pred, classes,

                          normalize=False,

                          title=None,

                          cmap=plt.cm.Blues):

    """

    This function prints and plots the confusion matrix.

    Normalization can be applied by setting `normalize=True`.

    """

    if not title:
```

```python
    if normalize:
        title = 'Normalized confusion matrix'
    else:
        title = 'Confusion matrix, without normalization'

# Compute confusion matrix
cm = confusion_matrix(y_true, y_pred)
# Only use the labels that appear in the data
classes = classes[unique_labels(y_true, y_pred)]
if normalize:
    cm = cm.astype('float') / cm.sum(axis=1)[:, np.newaxis]
    print("Normalized confusion matrix")
else:
    print('Confusion matrix, without normalization')

print(cm)

fig, ax = plt.subplots()
im = ax.imshow(cm, interpolation='nearest', cmap=cmap)
ax.figure.colorbar(im, ax=ax)
# We want to show all ticks...
ax.set(xticks=np.arange(cm.shape[1]),
       yticks=np.arange(cm.shape[0]),
       # ... and label them with the respective list entries
       xticklabels=classes, yticklabels=classes,
       title=title,
       ylabel='True label',
       xlabel='Predicted label')
```

```
    # Rotate the tick labels and set their alignment.
    plt.setp(ax.get_xticklabels(), rotation=45, ha="right",
             rotation_mode="anchor")

    # Loop over data dimensions and create text annotations.
    fmt = '.2f' if normalize else 'd'
    thresh = cm.max() / 2.
    for i in range(cm.shape[0]):
        for j in range(cm.shape[1]):
            ax.text(j, i, format(cm[i, j], fmt),
                    ha="center", va="center",
                    color="white" if cm[i, j] > thresh else "black")
    fig.tight_layout()
    return ax
```

Raw Image Loading and Processing

The load_raw_images_df function loads raw images from a DataFrame, resizes them, performs one-hot encoding on labels, and returns processed image data (X) and corresponding labels (Y).

```
def
load_raw_images_df(data_frame,filenamecol,labelcol,img_size,n_classes):
    n_images = len(data_frame)
    X = np.empty((n_images,img_size,img_size,3))
    Y = np.zeros((n_images,n_classes))
    for index,entry in data_frame.iterrows():
        Y[index,entry[labelcol]] = 1 # one hot encoding of the label
        # Load the image and resize
        img = cv2.imread(entry[filenamecol])
```

```
        X[index,:] = cv2.resize(img, (img_size, img_size))

        X[index,:] = X[index,:] / 255.0

    return X,Y
```

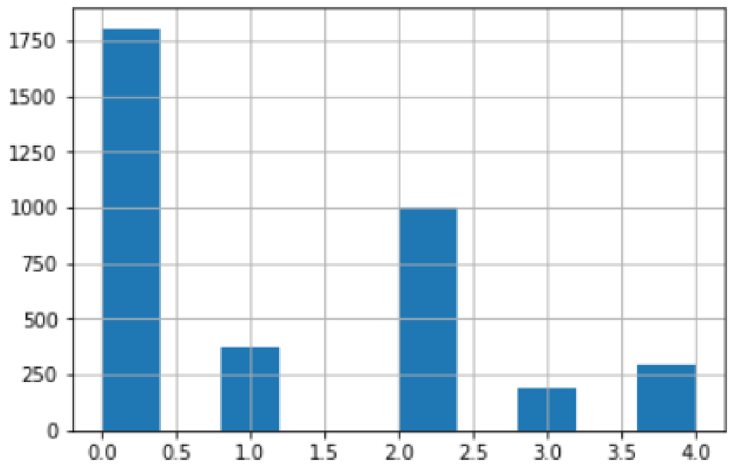
```
batch_size = 32

img_size = 224
```

EDA

The code reads a CSV file containing information about the APTOS Blindness Detection dataset, appends file paths to image filenames, and visualizes the distribution of diagnosis classes using a histogram.

```
train_raw_data =
pd.read_csv("../input/aptos2019-blindness-detection/train.csv")

train_raw_data["filename"] = train_raw_data["id_code"].map(lambda
x:os.path.join("../input/aptos2019-blindness-detection/train_images",x+
".png"))

train_raw_data.diagnosis.hist() # See the distribution of the classes

# train_raw_data.dtypes


# # train_data["diagnosis"] = train_data["diagnosis"].astype(str)

# # print(train_data.head())

# # print(train_data.diagnosis.unique()) # Look at different types of
classes

# # labels = list(map(str,range(5)))

# # print(labels)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f59
46e08dd8>
```

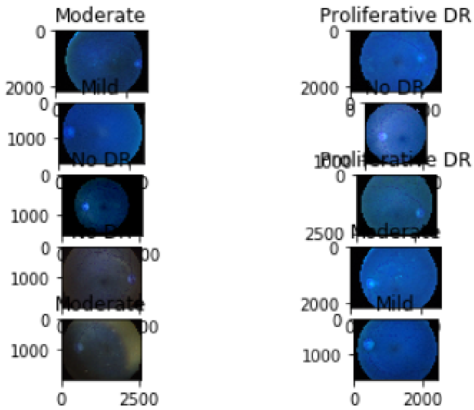


creates a mapping between numerical labels and descriptive titles for APTOS Blindness Detection classes, providing both a dictionary (label_title) and a list (class_labels) for convenient reference

```
label_title = {"0" : "No DR","1" : "Mild","2" : "Moderate","3"
:"Severe","4" : "Proliferative DR"}

class_labels=["No DR","Mild","Moderate","Severe","Proliferative DR"]
```

code creates a 5x2 grid of subplots to display images from the dataset. It iterates through the first 10 rows of the dataset, loads and displays images with their corresponding diagnosis labels as titles

```
# Display some images

figure, ax = plt.subplots(5,2)

ax = ax.flatten()

for i,row in train_raw_data.iloc[0:10,:].iterrows():


ax[i].imshow(cv2.imread(os.path.join("../input/aptos2019-blindness-dete
ction/train_images",row["id_code"]+".png")))

    ax[i].set_title(label_title[str(row["diagnosis"])])
```



Training and Validation

```
train_df,val_df =
train_test_split(train_raw_data,random_state=42,shuffle=True,test_size=
0.333)

train_df.reset_index(drop=True,inplace=True)

val_df.reset_index(drop=True,inplace=True)
```

train-validation split on the dataset, then calculates class weights using the compute_class_weight function from scikit-learn.

```
X_train,Y_train =
load_raw_images_df(train_df,"filename","diagnosis",img_size,5)

X_val,Y_val =
load_raw_images_df(val_df,"filename","diagnosis",img_size,5)
```

```
Y_train_labels = np.argmax(Y_train,axis=1)

class_weights =
class_weight.compute_class_weight('balanced',np.unique(Y_train_labels),
Y_train_labels)

cls_wt_dict = dict(enumerate(class_weights))

print(cls_wt_dict)
```

```
{0: 0.40397022332506205, 1: 1.9304347826086956,
2: 0.7333333333333333, 3: 3.7282442748091604, 4:
2.6688524590163936}
```

```
datagen = ImageDataGenerator(
```

```
            zoom_range=0.15,  # set range for random zoom
        # set mode for filling points outside the input boundaries
        fill_mode='constant',
        cval=0.,  # value used for fill_mode = "constant"
        horizontal_flip=True,  # randomly flip images
        vertical_flip=True,  # randomly flip images
)
training_generator = MixupGenerator(X_train, Y_train,
batch_size=batch_size, alpha=0.2, datagen=datagen)()
```

Build the Model:

DenseNet121-based Classification Model

The buildModel function constructs a classification model based on DenseNet121 architecture . It uses transfer learning, loading pre-trained weights and appending additional layers for classification

```
def buildModel():
    DenseNet121_model =
DenseNet121(include_top=False,weights=None,input_tensor=keras.layers.In
put(shape=(img_size,img_size,3)))

DenseNet121_model.load_weights('../input/densenet-keras/DenseNet-BC-121
-32-no-top.h5')
#     model = keras.Sequential()


#     model.add(keras.layers.Conv2D(filters = 32, kernel_size =
(5,5),padding = 'same',activation ='relu',
#                       input_shape = (img_size,img_size,3)))
#     model.add(keras.layers.MaxPooling2D(pool_size=(2,2)))
```

```
#     model.add(keras.layers.Conv2D(filters = 64, kernel_size =
(3,3),padding = 'Same',activation ='relu'))

#     model.add(keras.layers.MaxPooling2D(pool_size=(2,2),
strides=(2,2)))


#     model.add(keras.layers.Conv2D(filters =96, kernel_size =
(3,3),padding = 'Same',activation ='relu'))

#     model.add(keras.layers.MaxPooling2D(pool_size=(2,2),
strides=(2,2)))


#     model.add(keras.layers.Conv2D(filters = 96, kernel_size =
(3,3),padding = 'Same',activation ='relu'))

#     model.add(keras.layers.MaxPooling2D(pool_size=(2,2),
strides=(2,2)))


#     model.add(keras.layers.Flatten())

#     model.add(keras.layers.Dense(units = 512, activation = 'relu'))

#     model.add(keras.layers.Dense(units = 5, activation = 'softmax'))


    p  =
keras.layers.GlobalAveragePooling2D()(DenseNet121_model.output)

#     fl = keras.layers.Flatten()(p)

#     d2 = keras.layers.Dense(units = 1024, activation =
'relu',kernel_regularizer= keras.regularizers.l2(0.001))(p)

#     d1 = keras.layers.Dense(units = 512, activation =
'relu',kernel_regularizer= keras.regularizers.l2(0.001))(d2)

    d11 = keras.layers.Dense(units = 256, activation =
'relu',kernel_regularizer= keras.regularizers.l2(0.0001))(p)

    o1 = keras.layers.Dense(units = 5, activation = 'softmax')(d11)

    model = keras.models.Model(inputs = DenseNet121_model.input,outputs
= o1)

    sgd = keras.optimizers.SGD(lr=0.01, decay=1e-6, momentum=0.9,
nesterov=True)
```

```
    model.compile(optimizer=sgd,loss='categorical_crossentropy',
metrics = ['accuracy'])

    print(model.summary())

    return model
```

```
mymodel = buildModel()
```

```
conv2_block3_0_relu (Activation (None, 56, 56, 128)  0           conv2_block3_0_bn[0][0]
__________________________________________________________________________________________________
conv2_block3_1_conv (Conv2D)    (None, 56, 56, 128)  16384       conv2_block3_0_relu[0][0]
__________________________________________________________________________________________________
conv2_block3_1_bn (BatchNormali (None, 56, 56, 128)  512         conv2_block3_1_conv[0][0]
__________________________________________________________________________________________________
conv2_block3_1_relu (Activation (None, 56, 56, 128)  0           conv2_block3_1_bn[0][0]
__________________________________________________________________________________________________
conv2_block3_2_conv (Conv2D)    (None, 56, 56, 32)   36864       conv2_block3_1_relu[0][0]
__________________________________________________________________________________________________
conv2_block3_concat (Concatenat (None, 56, 56, 160)  0           conv2_block2_concat[0][0]
                                                                 conv2_block3_2_conv[0][0]
__________________________________________________________________________________________________
conv2_block4_0_bn (BatchNormali (None, 56, 56, 160)  640         conv2_block3_concat[0][0]
__________________________________________________________________________________________________
conv2_block4_0_relu (Activation (None, 56, 56, 160)  0           conv2_block4_0_bn[0][0]
__________________________________________________________________________________________________
conv2_block4_1_conv (Conv2D)    (None, 56, 56, 128)  20480       conv2_block4_0_relu[0][0]
__________________________________________________________________________________________________
conv2_block4_1_bn (BatchNormali (None, 56, 56, 128)  512         conv2_block4_1_conv[0][0]
__________________________________________________________________________________________________
conv2_block4_1_relu (Activation (None, 56, 56, 128)  0           conv2_block4_1_bn[0][0]
__________________________________________________________________________________________________
conv2_block4_2_conv (Conv2D)    (None, 56, 56, 32)   36864       conv2_block4_1_relu[0][0]
__________________________________________________________________________________________________
conv2_block4_concat (Concatenat (None, 56, 56, 192)  0           conv2_block3_concat[0][0]
                                                                 conv2_block4_2_conv[0][0]
__________________________________________________________________________________________________
conv2_block5_0_bn (BatchNormali (None, 56, 56, 192)  768         conv2_block4_concat[0][0]
__________________________________________________________________________________________________
conv2_block5_0_relu (Activation (None, 56, 56, 192)  0           conv2_block5_0_bn[0][0]
__________________________________________________________________________________________________
conv2_block5_1_conv (Conv2D)    (None, 56, 56, 128)  24576       conv2_block5_0_relu[0][0]
__________________________________________________________________________________________________
conv2_block5_1_bn (BatchNormali (None, 56, 56, 128)  512         conv2_block5_1_conv[0][0]
__________________________________________________________________________________________________
conv2_block5_1_relu (Activation (None, 56, 56, 128)  0           conv2_block5_1_bn[0][0]
__________________________________________________________________________________________________
conv2_block5_2_conv (Conv2D)    (None, 56, 56, 32)   36864       conv2_block5_1_relu[0][0]
__________________________________________________________________________________________________
conv2_block5_concat (Concatenat (None, 56, 56, 224)  0           conv2_block4_concat[0][0]
                                                                 conv2_block5_2_conv[0][0]
__________________________________________________________________________________________________
conv2_block6_0_bn (BatchNormali (None, 56, 56, 224)  896         conv2_block5_concat[0][0]
__________________________________________________________________________________________________
conv2_block6_0_relu (Activation (None, 56, 56, 224)  0           conv2_block6_0_bn[0][0]
__________________________________________________________________________________________________
conv2_block6_1_conv (Conv2D)    (None, 56, 56, 128)  28672       conv2_block6_0_relu[0][0]
__________________________________________________________________________________________________
conv2_block6_1_bn (BatchNormali (None, 56, 56, 128)  512         conv2_block6_1_conv[0][0]
__________________________________________________________________________________________________
conv2_block6_1_relu (Activation (None, 56, 56, 128)  0           conv2_block6_1_bn[0][0]
__________________________________________________________________________________________________
conv2_block6_2_conv (Conv2D)    (None, 56, 56, 32)   36864       conv2_block6_1_relu[0][0]
```

```
__________________________________________________________________________________________________
conv2_block6_2_conv (Conv2D)    (None, 56, 56, 32)   36864       conv2_block6_1_relu[0][0]
__________________________________________________________________________________________________
conv2_block6_concat (Concatenat (None, 56, 56, 256)  0           conv2_block5_concat[0][0]
                                                                 conv2_block6_2_conv[0][0]
__________________________________________________________________________________________________
pool2_bn (BatchNormalization)   (None, 56, 56, 256)  1024        conv2_block6_concat[0][0]
__________________________________________________________________________________________________
pool2_relu (Activation)         (None, 56, 56, 256)  0           pool2_bn[0][0]
__________________________________________________________________________________________________
pool2_conv (Conv2D)             (None, 56, 56, 128)  32768       pool2_relu[0][0]
__________________________________________________________________________________________________
pool2_pool (AveragePooling2D)   (None, 28, 28, 128)  0           pool2_conv[0][0]
__________________________________________________________________________________________________
conv3_block1_0_bn (BatchNormali (None, 28, 28, 128)  512         pool2_pool[0][0]
__________________________________________________________________________________________________
conv3_block1_0_relu (Activation (None, 28, 28, 128)  0           conv3_block1_0_bn[0][0]
__________________________________________________________________________________________________
conv3_block1_1_conv (Conv2D)    (None, 28, 28, 128)  16384       conv3_block1_0_relu[0][0]
__________________________________________________________________________________________________
conv3_block1_1_bn (BatchNormali (None, 28, 28, 128)  512         conv3_block1_1_conv[0][0]
__________________________________________________________________________________________________
conv3_block1_1_relu (Activation (None, 28, 28, 128)  0           conv3_block1_1_bn[0][0]
__________________________________________________________________________________________________
conv3_block1_2_conv (Conv2D)    (None, 28, 28, 32)   36864       conv3_block1_1_relu[0][0]
__________________________________________________________________________________________________
conv3_block1_concat (Concatenat (None, 28, 28, 160)  0           pool2_pool[0][0]
                                                                 conv3_block1_2_conv[0][0]
__________________________________________________________________________________________________
conv3_block2_0_bn (BatchNormali (None, 28, 28, 160)  640         conv3_block1_concat[0][0]
__________________________________________________________________________________________________
conv3_block2_0_relu (Activation (None, 28, 28, 160)  0           conv3_block2_0_bn[0][0]
__________________________________________________________________________________________________
conv3_block2_1_conv (Conv2D)    (None, 28, 28, 128)  20480       conv3_block2_0_relu[0][0]
__________________________________________________________________________________________________
conv3_block2_1_bn (BatchNormali (None, 28, 28, 128)  512         conv3_block2_1_conv[0][0]
__________________________________________________________________________________________________
conv3_block2_1_relu (Activation (None, 28, 28, 128)  0           conv3_block2_1_bn[0][0]
__________________________________________________________________________________________________
conv3_block2_2_conv (Conv2D)    (None, 28, 28, 32)   36864       conv3_block2_1_relu[0][0]
__________________________________________________________________________________________________
conv3_block2_concat (Concatenat (None, 28, 28, 192)  0           conv3_block1_concat[0][0]
                                                                 conv3_block2_2_conv[0][0]
__________________________________________________________________________________________________
conv3_block3_0_bn (BatchNormali (None, 28, 28, 192)  768         conv3_block2_concat[0][0]
__________________________________________________________________________________________________
conv3_block3_0_relu (Activation (None, 28, 28, 192)  0           conv3_block3_0_bn[0][0]
__________________________________________________________________________________________________
conv3_block3_1_conv (Conv2D)    (None, 28, 28, 128)  24576       conv3_block3_0_relu[0][0]
__________________________________________________________________________________________________
conv3_block3_1_bn (BatchNormali (None, 28, 28, 128)  512         conv3_block3_1_conv[0][0]
__________________________________________________________________________________________________
conv3_block3_1_relu (Activation (None, 28, 28, 128)  0           conv3_block3_1_bn[0][0]
__________________________________________________________________________________________________
conv3_block3_2_conv (Conv2D)    (None, 28, 28, 32)   36864       conv3_block3_1_relu[0][0]
```

```
____________________________________________________________________________________________________
conv3_block3_2_conv (Conv2D)    (None, 28, 28, 32)   36864       conv3_block3_1_relu[0][0]
____________________________________________________________________________________________________
conv3_block3_concat (Concatenat (None, 28, 28, 224)  0           conv3_block2_concat[0][0]
                                                                 conv3_block3_2_conv[0][0]
____________________________________________________________________________________________________
conv3_block4_0_bn (BatchNormali (None, 28, 28, 224)  896         conv3_block3_concat[0][0]
____________________________________________________________________________________________________
conv3_block4_0_relu (Activation (None, 28, 28, 224)  0           conv3_block4_0_bn[0][0]
____________________________________________________________________________________________________
conv3_block4_1_conv (Conv2D)    (None, 28, 28, 128)  28672       conv3_block4_0_relu[0][0]
____________________________________________________________________________________________________
conv3_block4_1_bn (BatchNormali (None, 28, 28, 128)  512         conv3_block4_1_conv[0][0]
____________________________________________________________________________________________________
conv3_block4_1_relu (Activation (None, 28, 28, 128)  0           conv3_block4_1_bn[0][0]
____________________________________________________________________________________________________
conv3_block4_2_conv (Conv2D)    (None, 28, 28, 32)   36864       conv3_block4_1_relu[0][0]
____________________________________________________________________________________________________
conv3_block4_concat (Concatenat (None, 28, 28, 256)  0           conv3_block3_concat[0][0]
                                                                 conv3_block4_2_conv[0][0]
____________________________________________________________________________________________________
conv3_block5_0_bn (BatchNormali (None, 28, 28, 256)  1024        conv3_block4_concat[0][0]
____________________________________________________________________________________________________
conv3_block5_0_relu (Activation (None, 28, 28, 256)  0           conv3_block5_0_bn[0][0]
____________________________________________________________________________________________________
conv3_block5_1_conv (Conv2D)    (None, 28, 28, 128)  32768       conv3_block5_0_relu[0][0]
____________________________________________________________________________________________________
conv3_block5_1_bn (BatchNormali (None, 28, 28, 128)  512         conv3_block5_1_conv[0][0]
____________________________________________________________________________________________________
conv3_block5_1_relu (Activation (None, 28, 28, 128)  0           conv3_block5_1_bn[0][0]
____________________________________________________________________________________________________
conv3_block5_2_conv (Conv2D)    (None, 28, 28, 32)   36864       conv3_block5_1_relu[0][0]
____________________________________________________________________________________________________
conv3_block5_concat (Concatenat (None, 28, 28, 288)  0           conv3_block4_concat[0][0]
                                                                 conv3_block5_2_conv[0][0]
____________________________________________________________________________________________________
conv3_block6_0_bn (BatchNormali (None, 28, 28, 288)  1152        conv3_block5_concat[0][0]
____________________________________________________________________________________________________
conv3_block6_0_relu (Activation (None, 28, 28, 288)  0           conv3_block6_0_bn[0][0]
____________________________________________________________________________________________________
conv3_block6_1_conv (Conv2D)    (None, 28, 28, 128)  36864       conv3_block6_0_relu[0][0]
____________________________________________________________________________________________________
conv3_block6_1_bn (BatchNormali (None, 28, 28, 128)  512         conv3_block6_1_conv[0][0]
____________________________________________________________________________________________________
conv3_block6_1_relu (Activation (None, 28, 28, 128)  0           conv3_block6_1_bn[0][0]
____________________________________________________________________________________________________
conv3_block6_2_conv (Conv2D)    (None, 28, 28, 32)   36864       conv3_block6_1_relu[0][0]
____________________________________________________________________________________________________
conv3_block6_concat (Concatenat (None, 28, 28, 320)  0           conv3_block5_concat[0][0]
                                                                 conv3_block6_2_conv[0][0]
____________________________________________________________________________________________________
conv3_block7_0_bn (BatchNormali (None, 28, 28, 320)  1280        conv3_block6_concat[0][0]
____________________________________________________________________________________________________
conv3_block7_0_relu (Activation (None, 28, 28, 320)  0           conv3_block7_0_bn[0][0]
```

```
__________________________________________________________________________________________________
conv3_block7_0_relu (Activation (None, 28, 28, 320)  0           conv3_block7_0_bn[0][0]
__________________________________________________________________________________________________
conv3_block7_1_conv (Conv2D)    (None, 28, 28, 128)  40960       conv3_block7_0_relu[0][0]
__________________________________________________________________________________________________
conv3_block7_1_bn (BatchNormali (None, 28, 28, 128)  512         conv3_block7_1_conv[0][0]
__________________________________________________________________________________________________
conv3_block7_1_relu (Activation (None, 28, 28, 128)  0           conv3_block7_1_bn[0][0]
__________________________________________________________________________________________________
conv3_block7_2_conv (Conv2D)    (None, 28, 28, 32)   36864       conv3_block7_1_relu[0][0]
__________________________________________________________________________________________________
conv3_block7_concat (Concatenat (None, 28, 28, 352)  0           conv3_block6_concat[0][0]
                                                                 conv3_block7_2_conv[0][0]
__________________________________________________________________________________________________
conv3_block8_0_bn (BatchNormali (None, 28, 28, 352)  1408        conv3_block7_concat[0][0]
__________________________________________________________________________________________________
conv3_block8_0_relu (Activation (None, 28, 28, 352)  0           conv3_block8_0_bn[0][0]
__________________________________________________________________________________________________
conv3_block8_1_conv (Conv2D)    (None, 28, 28, 128)  45056       conv3_block8_0_relu[0][0]
__________________________________________________________________________________________________
conv3_block8_1_bn (BatchNormali (None, 28, 28, 128)  512         conv3_block8_1_conv[0][0]
__________________________________________________________________________________________________
conv3_block8_1_relu (Activation (None, 28, 28, 128)  0           conv3_block8_1_bn[0][0]
__________________________________________________________________________________________________
conv3_block8_2_conv (Conv2D)    (None, 28, 28, 32)   36864       conv3_block8_1_relu[0][0]
__________________________________________________________________________________________________
conv3_block8_concat (Concatenat (None, 28, 28, 384)  0           conv3_block7_concat[0][0]
                                                                 conv3_block8_2_conv[0][0]
__________________________________________________________________________________________________
conv3_block9_0_bn (BatchNormali (None, 28, 28, 384)  1536        conv3_block8_concat[0][0]
__________________________________________________________________________________________________
conv3_block9_0_relu (Activation (None, 28, 28, 384)  0           conv3_block9_0_bn[0][0]
__________________________________________________________________________________________________
conv3_block9_1_conv (Conv2D)    (None, 28, 28, 128)  49152       conv3_block9_0_relu[0][0]
__________________________________________________________________________________________________
conv3_block9_1_bn (BatchNormali (None, 28, 28, 128)  512         conv3_block9_1_conv[0][0]
__________________________________________________________________________________________________
conv3_block9_1_relu (Activation (None, 28, 28, 128)  0           conv3_block9_1_bn[0][0]
__________________________________________________________________________________________________
conv3_block9_2_conv (Conv2D)    (None, 28, 28, 32)   36864       conv3_block9_1_relu[0][0]
__________________________________________________________________________________________________
conv3_block9_concat (Concatenat (None, 28, 28, 416)  0           conv3_block8_concat[0][0]
                                                                 conv3_block9_2_conv[0][0]
__________________________________________________________________________________________________
conv3_block10_0_bn (BatchNormal (None, 28, 28, 416)  1664        conv3_block9_concat[0][0]
__________________________________________________________________________________________________
conv3_block10_0_relu (Activatio (None, 28, 28, 416)  0           conv3_block10_0_bn[0][0]
__________________________________________________________________________________________________
conv3_block10_1_conv (Conv2D)   (None, 28, 28, 128)  53248       conv3_block10_0_relu[0][0]
__________________________________________________________________________________________________
conv3_block10_1_bn (BatchNormal (None, 28, 28, 128)  512         conv3_block10_1_conv[0][0]
__________________________________________________________________________________________________
conv3_block10_1_relu (Activatio (None, 28, 28, 128)  0           conv3_block10_1_bn[0][0]
```

```
__________________________________________________________________________________________________
conv3_block10_1_relu (Activatio (None, 28, 28, 128)  0           conv3_block10_1_bn[0][0]
__________________________________________________________________________________________________
conv3_block10_2_conv (Conv2D)   (None, 28, 28, 32)   36864       conv3_block10_1_relu[0][0]
__________________________________________________________________________________________________
conv3_block10_concat (Concatena (None, 28, 28, 448)  0           conv3_block9_concat[0][0]
                                                                 conv3_block10_2_conv[0][0]
__________________________________________________________________________________________________
conv3_block11_0_bn (BatchNormal (None, 28, 28, 448)  1792        conv3_block10_concat[0][0]
__________________________________________________________________________________________________
conv3_block11_0_relu (Activatio (None, 28, 28, 448)  0           conv3_block11_0_bn[0][0]
__________________________________________________________________________________________________
conv3_block11_1_conv (Conv2D)   (None, 28, 28, 128)  57344       conv3_block11_0_relu[0][0]
__________________________________________________________________________________________________
conv3_block11_1_bn (BatchNormal (None, 28, 28, 128)  512         conv3_block11_1_conv[0][0]
__________________________________________________________________________________________________
conv3_block11_1_relu (Activatio (None, 28, 28, 128)  0           conv3_block11_1_bn[0][0]
__________________________________________________________________________________________________
conv3_block11_2_conv (Conv2D)   (None, 28, 28, 32)   36864       conv3_block11_1_relu[0][0]
__________________________________________________________________________________________________
conv3_block11_concat (Concatena (None, 28, 28, 480)  0           conv3_block10_concat[0][0]
                                                                 conv3_block11_2_conv[0][0]
__________________________________________________________________________________________________
conv3_block12_0_bn (BatchNormal (None, 28, 28, 480)  1920        conv3_block11_concat[0][0]
__________________________________________________________________________________________________
conv3_block12_0_relu (Activatio (None, 28, 28, 480)  0           conv3_block12_0_bn[0][0]
__________________________________________________________________________________________________
conv3_block12_1_conv (Conv2D)   (None, 28, 28, 128)  61440       conv3_block12_0_relu[0][0]
__________________________________________________________________________________________________
conv3_block12_1_bn (BatchNormal (None, 28, 28, 128)  512         conv3_block12_1_conv[0][0]
__________________________________________________________________________________________________
conv3_block12_1_relu (Activatio (None, 28, 28, 128)  0           conv3_block12_1_bn[0][0]
__________________________________________________________________________________________________
conv3_block12_2_conv (Conv2D)   (None, 28, 28, 32)   36864       conv3_block12_1_relu[0][0]
__________________________________________________________________________________________________
conv3_block12_concat (Concatena (None, 28, 28, 512)  0           conv3_block11_concat[0][0]
                                                                 conv3_block12_2_conv[0][0]
__________________________________________________________________________________________________
pool3_bn (BatchNormalization)   (None, 28, 28, 512)  2048        conv3_block12_concat[0][0]
__________________________________________________________________________________________________
pool3_relu (Activation)         (None, 28, 28, 512)  0           pool3_bn[0][0]
__________________________________________________________________________________________________
pool3_conv (Conv2D)             (None, 28, 28, 256)  131072      pool3_relu[0][0]
__________________________________________________________________________________________________
pool3_pool (AveragePooling2D)   (None, 14, 14, 256)  0           pool3_conv[0][0]
__________________________________________________________________________________________________
conv4_block1_0_bn (BatchNormali (None, 14, 14, 256)  1024        pool3_pool[0][0]
__________________________________________________________________________________________________
conv4_block1_0_relu (Activation (None, 14, 14, 256)  0           conv4_block1_0_bn[0][0]
__________________________________________________________________________________________________
conv4_block1_1_conv (Conv2D)    (None, 14, 14, 128)  32768       conv4_block1_0_relu[0][0]
__________________________________________________________________________________________________
conv4_block1_1_bn (BatchNormali (None, 14, 14, 128)  512         conv4_block1_1_conv[0][0]
__________________________________________________________________________________________________
```

```
conv4_block1_1_relu (Activation (None, 14, 14, 128)  0           conv4_block1_1_bn[0][0]
__________________________________________________________________________________________________
conv4_block1_2_conv (Conv2D)    (None, 14, 14, 32)   36864       conv4_block1_1_relu[0][0]
__________________________________________________________________________________________________
conv4_block1_concat (Concatenat (None, 14, 14, 288)  0           pool3_pool[0][0]
                                                                 conv4_block1_2_conv[0][0]
__________________________________________________________________________________________________
conv4_block2_0_bn (BatchNormali (None, 14, 14, 288)  1152        conv4_block1_concat[0][0]
__________________________________________________________________________________________________
conv4_block2_0_relu (Activation (None, 14, 14, 288)  0           conv4_block2_0_bn[0][0]
__________________________________________________________________________________________________
conv4_block2_1_conv (Conv2D)    (None, 14, 14, 128)  36864       conv4_block2_0_relu[0][0]
__________________________________________________________________________________________________
conv4_block2_1_bn (BatchNormali (None, 14, 14, 128)  512         conv4_block2_1_conv[0][0]
__________________________________________________________________________________________________
conv4_block2_1_relu (Activation (None, 14, 14, 128)  0           conv4_block2_1_bn[0][0]
__________________________________________________________________________________________________
conv4_block2_2_conv (Conv2D)    (None, 14, 14, 32)   36864       conv4_block2_1_relu[0][0]
__________________________________________________________________________________________________
conv4_block2_concat (Concatenat (None, 14, 14, 320)  0           conv4_block1_concat[0][0]
                                                                 conv4_block2_2_conv[0][0]
__________________________________________________________________________________________________
conv4_block3_0_bn (BatchNormali (None, 14, 14, 320)  1280        conv4_block2_concat[0][0]
__________________________________________________________________________________________________
conv4_block3_0_relu (Activation (None, 14, 14, 320)  0           conv4_block3_0_bn[0][0]
__________________________________________________________________________________________________
conv4_block3_1_conv (Conv2D)    (None, 14, 14, 128)  40960       conv4_block3_0_relu[0][0]
__________________________________________________________________________________________________
conv4_block3_1_bn (BatchNormali (None, 14, 14, 128)  512         conv4_block3_1_conv[0][0]
__________________________________________________________________________________________________
conv4_block3_1_relu (Activation (None, 14, 14, 128)  0           conv4_block3_1_bn[0][0]
__________________________________________________________________________________________________
conv4_block3_2_conv (Conv2D)    (None, 14, 14, 32)   36864       conv4_block3_1_relu[0][0]
__________________________________________________________________________________________________
conv4_block3_concat (Concatenat (None, 14, 14, 352)  0           conv4_block2_concat[0][0]
                                                                 conv4_block3_2_conv[0][0]
__________________________________________________________________________________________________
conv4_block4_0_bn (BatchNormali (None, 14, 14, 352)  1408        conv4_block3_concat[0][0]
__________________________________________________________________________________________________
conv4_block4_0_relu (Activation (None, 14, 14, 352)  0           conv4_block4_0_bn[0][0]
__________________________________________________________________________________________________
conv4_block4_1_conv (Conv2D)    (None, 14, 14, 128)  45056       conv4_block4_0_relu[0][0]
__________________________________________________________________________________________________
conv4_block4_1_bn (BatchNormali (None, 14, 14, 128)  512         conv4_block4_1_conv[0][0]
__________________________________________________________________________________________________
conv4_block4_1_relu (Activation (None, 14, 14, 128)  0           conv4_block4_1_bn[0][0]
__________________________________________________________________________________________________
conv4_block4_2_conv (Conv2D)    (None, 14, 14, 32)   36864       conv4_block4_1_relu[0][0]
__________________________________________________________________________________________________
conv4_block4_concat (Concatenat (None, 14, 14, 384)  0           conv4_block3_concat[0][0]
                                                                 conv4_block4_2_conv[0][0]
__________________________________________________________________________________________________
conv4_block5_0_bn (BatchNormali (None, 14, 14, 384)  1536        conv4_block4_concat[0][0]
__________________________________________________________________________________________________
```

```
____________________________________________________________________________________________
conv4_block5_0_bn (BatchNormali (None, 14, 14, 384)  1536        conv4_block4_concat[0][0]
____________________________________________________________________________________________
conv4_block5_0_relu (Activation (None, 14, 14, 384)  0           conv4_block5_0_bn[0][0]
____________________________________________________________________________________________
conv4_block5_1_conv (Conv2D)    (None, 14, 14, 128)  49152       conv4_block5_0_relu[0][0]
____________________________________________________________________________________________
conv4_block5_1_bn (BatchNormali (None, 14, 14, 128)  512         conv4_block5_1_conv[0][0]
____________________________________________________________________________________________
conv4_block5_1_relu (Activation (None, 14, 14, 128)  0           conv4_block5_1_bn[0][0]
____________________________________________________________________________________________
conv4_block5_2_conv (Conv2D)    (None, 14, 14, 32)   36864       conv4_block5_1_relu[0][0]
____________________________________________________________________________________________
conv4_block5_concat (Concatenat (None, 14, 14, 416)  0           conv4_block4_concat[0][0]
                                                                 conv4_block5_2_conv[0][0]
____________________________________________________________________________________________
conv4_block6_0_bn (BatchNormali (None, 14, 14, 416)  1664        conv4_block5_concat[0][0]
____________________________________________________________________________________________
conv4_block6_0_relu (Activation (None, 14, 14, 416)  0           conv4_block6_0_bn[0][0]
____________________________________________________________________________________________
conv4_block6_1_conv (Conv2D)    (None, 14, 14, 128)  53248       conv4_block6_0_relu[0][0]
____________________________________________________________________________________________
conv4_block6_1_bn (BatchNormali (None, 14, 14, 128)  512         conv4_block6_1_conv[0][0]
____________________________________________________________________________________________
conv4_block6_1_relu (Activation (None, 14, 14, 128)  0           conv4_block6_1_bn[0][0]
____________________________________________________________________________________________
conv4_block6_2_conv (Conv2D)    (None, 14, 14, 32)   36864       conv4_block6_1_relu[0][0]
____________________________________________________________________________________________
conv4_block6_concat (Concatenat (None, 14, 14, 448)  0           conv4_block5_concat[0][0]
                                                                 conv4_block6_2_conv[0][0]
____________________________________________________________________________________________
conv4_block7_0_bn (BatchNormali (None, 14, 14, 448)  1792        conv4_block6_concat[0][0]
____________________________________________________________________________________________
conv4_block7_0_relu (Activation (None, 14, 14, 448)  0           conv4_block7_0_bn[0][0]
____________________________________________________________________________________________
conv4_block7_1_conv (Conv2D)    (None, 14, 14, 128)  57344       conv4_block7_0_relu[0][0]
____________________________________________________________________________________________
conv4_block7_1_bn (BatchNormali (None, 14, 14, 128)  512         conv4_block7_1_conv[0][0]
____________________________________________________________________________________________
conv4_block7_1_relu (Activation (None, 14, 14, 128)  0           conv4_block7_1_bn[0][0]
____________________________________________________________________________________________
conv4_block7_2_conv (Conv2D)    (None, 14, 14, 32)   36864       conv4_block7_1_relu[0][0]
____________________________________________________________________________________________
conv4_block7_concat (Concatenat (None, 14, 14, 480)  0           conv4_block6_concat[0][0]
                                                                 conv4_block7_2_conv[0][0]
____________________________________________________________________________________________
conv4_block8_0_bn (BatchNormali (None, 14, 14, 480)  1920        conv4_block7_concat[0][0]
____________________________________________________________________________________________
conv4_block8_0_relu (Activation (None, 14, 14, 480)  0           conv4_block8_0_bn[0][0]
____________________________________________________________________________________________
conv4_block8_1_conv (Conv2D)    (None, 14, 14, 128)  61440       conv4_block8_0_relu[0][0]
____________________________________________________________________________________________
conv4_block8_1_bn (BatchNormali (None, 14, 14, 128)  512         conv4_block8_1_conv[0][0]
____________________________________________________________________________________________
conv4_block8_1_relu (Activation (None, 14, 14, 128)  0           conv4_block8_1_bn[0][0]
```

```
__________________________________________________________________________________________________
conv4_block8_1_relu (Activation (None, 14, 14, 128)  0           conv4_block8_1_bn[0][0]
__________________________________________________________________________________________________
conv4_block8_2_conv (Conv2D)    (None, 14, 14, 32)   36864       conv4_block8_1_relu[0][0]
__________________________________________________________________________________________________
conv4_block8_concat (Concatenat (None, 14, 14, 512)  0           conv4_block7_concat[0][0]
                                                                 conv4_block8_2_conv[0][0]
__________________________________________________________________________________________________
conv4_block9_0_bn (BatchNormali (None, 14, 14, 512)  2048        conv4_block8_concat[0][0]
__________________________________________________________________________________________________
conv4_block9_0_relu (Activation (None, 14, 14, 512)  0           conv4_block9_0_bn[0][0]
__________________________________________________________________________________________________
conv4_block9_1_conv (Conv2D)    (None, 14, 14, 128)  65536       conv4_block9_0_relu[0][0]
__________________________________________________________________________________________________
conv4_block9_1_bn (BatchNormali (None, 14, 14, 128)  512         conv4_block9_1_conv[0][0]
__________________________________________________________________________________________________
conv4_block9_1_relu (Activation (None, 14, 14, 128)  0           conv4_block9_1_bn[0][0]
__________________________________________________________________________________________________
conv4_block9_2_conv (Conv2D)    (None, 14, 14, 32)   36864       conv4_block9_1_relu[0][0]
__________________________________________________________________________________________________
conv4_block9_concat (Concatenat (None, 14, 14, 544)  0           conv4_block8_concat[0][0]
                                                                 conv4_block9_2_conv[0][0]
__________________________________________________________________________________________________
conv4_block10_0_bn (BatchNormal (None, 14, 14, 544)  2176        conv4_block9_concat[0][0]
__________________________________________________________________________________________________
conv4_block10_0_relu (Activatio (None, 14, 14, 544)  0           conv4_block10_0_bn[0][0]
__________________________________________________________________________________________________
conv4_block10_1_conv (Conv2D)   (None, 14, 14, 128)  69632       conv4_block10_0_relu[0][0]
__________________________________________________________________________________________________
conv4_block10_1_bn (BatchNormal (None, 14, 14, 128)  512         conv4_block10_1_conv[0][0]
__________________________________________________________________________________________________
conv4_block10_1_relu (Activatio (None, 14, 14, 128)  0           conv4_block10_1_bn[0][0]
__________________________________________________________________________________________________
conv4_block10_2_conv (Conv2D)   (None, 14, 14, 32)   36864       conv4_block10_1_relu[0][0]
__________________________________________________________________________________________________
conv4_block10_concat (Concatena (None, 14, 14, 576)  0           conv4_block9_concat[0][0]
                                                                 conv4_block10_2_conv[0][0]
__________________________________________________________________________________________________
conv4_block11_0_bn (BatchNormal (None, 14, 14, 576)  2304        conv4_block10_concat[0][0]
__________________________________________________________________________________________________
conv4_block11_0_relu (Activatio (None, 14, 14, 576)  0           conv4_block11_0_bn[0][0]
__________________________________________________________________________________________________
conv4_block11_1_conv (Conv2D)   (None, 14, 14, 128)  73728       conv4_block11_0_relu[0][0]
__________________________________________________________________________________________________
conv4_block11_1_bn (BatchNormal (None, 14, 14, 128)  512         conv4_block11_1_conv[0][0]
__________________________________________________________________________________________________
conv4_block11_1_relu (Activatio (None, 14, 14, 128)  0           conv4_block11_1_bn[0][0]
__________________________________________________________________________________________________
conv4_block11_2_conv (Conv2D)   (None, 14, 14, 32)   36864       conv4_block11_1_relu[0][0]
__________________________________________________________________________________________________
conv4_block11_concat (Concatena (None, 14, 14, 608)  0           conv4_block10_concat[0][0]
                                                                 conv4_block11_2_conv[0][0]
__________________________________________________________________________________________________
conv4_block12_0_bn (BatchNormal (None, 14, 14, 608)  2432        conv4_block11_concat[0][0]
```

```
__________________________________________________________________________________________________
conv4_block12_0_bn (BatchNormal (None, 14, 14, 608)  2432        conv4_block11_concat[0][0]
__________________________________________________________________________________________________
conv4_block12_0_relu (Activatio (None, 14, 14, 608)  0           conv4_block12_0_bn[0][0]
__________________________________________________________________________________________________
conv4_block12_1_conv (Conv2D)   (None, 14, 14, 128)  77824       conv4_block12_0_relu[0][0]
__________________________________________________________________________________________________
conv4_block12_1_bn (BatchNormal (None, 14, 14, 128)  512         conv4_block12_1_conv[0][0]
__________________________________________________________________________________________________
conv4_block12_1_relu (Activatio (None, 14, 14, 128)  0           conv4_block12_1_bn[0][0]
__________________________________________________________________________________________________
conv4_block12_2_conv (Conv2D)   (None, 14, 14, 32)   36864       conv4_block12_1_relu[0][0]
__________________________________________________________________________________________________
conv4_block12_concat (Concatena (None, 14, 14, 640)  0           conv4_block11_concat[0][0]
                                                                 conv4_block12_2_conv[0][0]
__________________________________________________________________________________________________
conv4_block13_0_bn (BatchNormal (None, 14, 14, 640)  2560        conv4_block12_concat[0][0]
__________________________________________________________________________________________________
conv4_block13_0_relu (Activatio (None, 14, 14, 640)  0           conv4_block13_0_bn[0][0]
__________________________________________________________________________________________________
conv4_block13_1_conv (Conv2D)   (None, 14, 14, 128)  81920       conv4_block13_0_relu[0][0]
__________________________________________________________________________________________________
conv4_block13_1_bn (BatchNormal (None, 14, 14, 128)  512         conv4_block13_1_conv[0][0]
__________________________________________________________________________________________________
conv4_block13_1_relu (Activatio (None, 14, 14, 128)  0           conv4_block13_1_bn[0][0]
__________________________________________________________________________________________________
conv4_block13_2_conv (Conv2D)   (None, 14, 14, 32)   36864       conv4_block13_1_relu[0][0]
__________________________________________________________________________________________________
conv4_block13_concat (Concatena (None, 14, 14, 672)  0           conv4_block12_concat[0][0]
                                                                 conv4_block13_2_conv[0][0]
__________________________________________________________________________________________________
conv4_block14_0_bn (BatchNormal (None, 14, 14, 672)  2688        conv4_block13_concat[0][0]
__________________________________________________________________________________________________
conv4_block14_0_relu (Activatio (None, 14, 14, 672)  0           conv4_block14_0_bn[0][0]
__________________________________________________________________________________________________
conv4_block14_1_conv (Conv2D)   (None, 14, 14, 128)  86016       conv4_block14_0_relu[0][0]
__________________________________________________________________________________________________
conv4_block14_1_bn (BatchNormal (None, 14, 14, 128)  512         conv4_block14_1_conv[0][0]
__________________________________________________________________________________________________
conv4_block14_1_relu (Activatio (None, 14, 14, 128)  0           conv4_block14_1_bn[0][0]
__________________________________________________________________________________________________
conv4_block14_2_conv (Conv2D)   (None, 14, 14, 32)   36864       conv4_block14_1_relu[0][0]
__________________________________________________________________________________________________
conv4_block14_concat (Concatena (None, 14, 14, 704)  0           conv4_block13_concat[0][0]
                                                                 conv4_block14_2_conv[0][0]
__________________________________________________________________________________________________
conv4_block15_0_bn (BatchNormal (None, 14, 14, 704)  2816        conv4_block14_concat[0][0]
__________________________________________________________________________________________________
conv4_block15_0_relu (Activatio (None, 14, 14, 704)  0           conv4_block15_0_bn[0][0]
__________________________________________________________________________________________________
conv4_block15_1_conv (Conv2D)   (None, 14, 14, 128)  90112       conv4_block15_0_relu[0][0]
__________________________________________________________________________________________________
conv4_block15_1_bn (BatchNormal (None, 14, 14, 128)  512         conv4_block15_1_conv[0][0]
__________________________________________________________________________________________________
conv4_block15_1_relu (Activatio (None, 14, 14, 128)  0           conv4_block15_1_bn[0][0]
```

```
__________________________________________________________________________________________________
conv4_block15_1_relu (Activatio (None, 14, 14, 128)  0           conv4_block15_1_bn[0][0]
__________________________________________________________________________________________________
conv4_block15_2_conv (Conv2D)   (None, 14, 14, 32)   36864       conv4_block15_1_relu[0][0]
__________________________________________________________________________________________________
conv4_block15_concat (Concatena (None, 14, 14, 736)  0           conv4_block14_concat[0][0]
                                                                 conv4_block15_2_conv[0][0]
__________________________________________________________________________________________________
conv4_block16_0_bn (BatchNormal (None, 14, 14, 736)  2944        conv4_block15_concat[0][0]
__________________________________________________________________________________________________
conv4_block16_0_relu (Activatio (None, 14, 14, 736)  0           conv4_block16_0_bn[0][0]
__________________________________________________________________________________________________
conv4_block16_1_conv (Conv2D)   (None, 14, 14, 128)  94208       conv4_block16_0_relu[0][0]
__________________________________________________________________________________________________
conv4_block16_1_bn (BatchNormal (None, 14, 14, 128)  512         conv4_block16_1_conv[0][0]
__________________________________________________________________________________________________
conv4_block16_1_relu (Activatio (None, 14, 14, 128)  0           conv4_block16_1_bn[0][0]
__________________________________________________________________________________________________
conv4_block16_2_conv (Conv2D)   (None, 14, 14, 32)   36864       conv4_block16_1_relu[0][0]
__________________________________________________________________________________________________
conv4_block16_concat (Concatena (None, 14, 14, 768)  0           conv4_block15_concat[0][0]
                                                                 conv4_block16_2_conv[0][0]
__________________________________________________________________________________________________
conv4_block17_0_bn (BatchNormal (None, 14, 14, 768)  3072        conv4_block16_concat[0][0]
__________________________________________________________________________________________________
conv4_block17_0_relu (Activatio (None, 14, 14, 768)  0           conv4_block17_0_bn[0][0]
__________________________________________________________________________________________________
conv4_block17_1_conv (Conv2D)   (None, 14, 14, 128)  98304       conv4_block17_0_relu[0][0]
__________________________________________________________________________________________________
conv4_block17_1_bn (BatchNormal (None, 14, 14, 128)  512         conv4_block17_1_conv[0][0]
__________________________________________________________________________________________________
conv4_block17_1_relu (Activatio (None, 14, 14, 128)  0           conv4_block17_1_bn[0][0]
__________________________________________________________________________________________________
conv4_block17_2_conv (Conv2D)   (None, 14, 14, 32)   36864       conv4_block17_1_relu[0][0]
__________________________________________________________________________________________________
conv4_block17_concat (Concatena (None, 14, 14, 800)  0           conv4_block16_concat[0][0]
                                                                 conv4_block17_2_conv[0][0]
__________________________________________________________________________________________________
conv4_block18_0_bn (BatchNormal (None, 14, 14, 800)  3200        conv4_block17_concat[0][0]
__________________________________________________________________________________________________
conv4_block18_0_relu (Activatio (None, 14, 14, 800)  0           conv4_block18_0_bn[0][0]
__________________________________________________________________________________________________
conv4_block18_1_conv (Conv2D)   (None, 14, 14, 128)  102400      conv4_block18_0_relu[0][0]
__________________________________________________________________________________________________
conv4_block18_1_bn (BatchNormal (None, 14, 14, 128)  512         conv4_block18_1_conv[0][0]
__________________________________________________________________________________________________
conv4_block18_1_relu (Activatio (None, 14, 14, 128)  0           conv4_block18_1_bn[0][0]
__________________________________________________________________________________________________
conv4_block18_2_conv (Conv2D)   (None, 14, 14, 32)   36864       conv4_block18_1_relu[0][0]
__________________________________________________________________________________________________
conv4_block18_concat (Concatena (None, 14, 14, 832)  0           conv4_block17_concat[0][0]
                                                                 conv4_block18_2_conv[0][0]
__________________________________________________________________________________________________
conv4_block19_0_bn (BatchNormal (None, 14, 14, 832)  3328        conv4_block18_concat[0][0]
```

```
__________________________________________________________________________________________________
conv4_block5_0_bn (BatchNormali (None, 14, 14, 384)  1536        conv4_block4_concat[0][0]
__________________________________________________________________________________________________
conv4_block5_0_relu (Activation (None, 14, 14, 384)  0           conv4_block5_0_bn[0][0]
__________________________________________________________________________________________________
conv4_block5_1_conv (Conv2D)    (None, 14, 14, 128)  49152       conv4_block5_0_relu[0][0]
__________________________________________________________________________________________________
conv4_block5_1_bn (BatchNormali (None, 14, 14, 128)  512         conv4_block5_1_conv[0][0]
__________________________________________________________________________________________________
conv4_block5_1_relu (Activation (None, 14, 14, 128)  0           conv4_block5_1_bn[0][0]
__________________________________________________________________________________________________
conv4_block5_2_conv (Conv2D)    (None, 14, 14, 32)   36864       conv4_block5_1_relu[0][0]
__________________________________________________________________________________________________
conv4_block5_concat (Concatenat (None, 14, 14, 416)  0           conv4_block4_concat[0][0]
                                                                 conv4_block5_2_conv[0][0]
__________________________________________________________________________________________________
conv4_block6_0_bn (BatchNormali (None, 14, 14, 416)  1664        conv4_block5_concat[0][0]
__________________________________________________________________________________________________
conv4_block6_0_relu (Activation (None, 14, 14, 416)  0           conv4_block6_0_bn[0][0]
__________________________________________________________________________________________________
conv4_block6_1_conv (Conv2D)    (None, 14, 14, 128)  53248       conv4_block6_0_relu[0][0]
__________________________________________________________________________________________________
conv4_block6_1_bn (BatchNormali (None, 14, 14, 128)  512         conv4_block6_1_conv[0][0]
__________________________________________________________________________________________________
conv4_block6_1_relu (Activation (None, 14, 14, 128)  0           conv4_block6_1_bn[0][0]
__________________________________________________________________________________________________
conv4_block6_2_conv (Conv2D)    (None, 14, 14, 32)   36864       conv4_block6_1_relu[0][0]
__________________________________________________________________________________________________
conv4_block6_concat (Concatenat (None, 14, 14, 448)  0           conv4_block5_concat[0][0]
                                                                 conv4_block6_2_conv[0][0]
__________________________________________________________________________________________________
conv4_block7_0_bn (BatchNormali (None, 14, 14, 448)  1792        conv4_block6_concat[0][0]
__________________________________________________________________________________________________
conv4_block7_0_relu (Activation (None, 14, 14, 448)  0           conv4_block7_0_bn[0][0]
__________________________________________________________________________________________________
conv4_block7_1_conv (Conv2D)    (None, 14, 14, 128)  57344       conv4_block7_0_relu[0][0]
__________________________________________________________________________________________________
conv4_block7_1_bn (BatchNormali (None, 14, 14, 128)  512         conv4_block7_1_conv[0][0]
__________________________________________________________________________________________________
conv4_block7_1_relu (Activation (None, 14, 14, 128)  0           conv4_block7_1_bn[0][0]
__________________________________________________________________________________________________
conv4_block7_2_conv (Conv2D)    (None, 14, 14, 32)   36864       conv4_block7_1_relu[0][0]
__________________________________________________________________________________________________
conv4_block7_concat (Concatenat (None, 14, 14, 480)  0           conv4_block6_concat[0][0]
                                                                 conv4_block7_2_conv[0][0]
__________________________________________________________________________________________________
conv4_block8_0_bn (BatchNormali (None, 14, 14, 480)  1920        conv4_block7_concat[0][0]
__________________________________________________________________________________________________
conv4_block8_0_relu (Activation (None, 14, 14, 480)  0           conv4_block8_0_bn[0][0]
__________________________________________________________________________________________________
conv4_block8_1_conv (Conv2D)    (None, 14, 14, 128)  61440       conv4_block8_0_relu[0][0]
__________________________________________________________________________________________________
conv4_block8_1_bn (BatchNormali (None, 14, 14, 128)  512         conv4_block8_1_conv[0][0]
__________________________________________________________________________________________________
conv4_block8_1_relu (Activation (None, 14, 14, 128)  0           conv4_block8_1_bn[0][0]
```

```
________________________________________________________________________________________________
conv4_block15_1_relu (Activatio (None, 14, 14, 128)  0           conv4_block15_1_bn[0][0]
________________________________________________________________________________________________
conv4_block15_2_conv (Conv2D)   (None, 14, 14, 32)   36864       conv4_block15_1_relu[0][0]
________________________________________________________________________________________________
conv4_block15_concat (Concatena (None, 14, 14, 736)  0           conv4_block14_concat[0][0]
                                                                 conv4_block15_2_conv[0][0]
________________________________________________________________________________________________
conv4_block16_0_bn (BatchNormal (None, 14, 14, 736)  2944        conv4_block15_concat[0][0]
________________________________________________________________________________________________
conv4_block16_0_relu (Activatio (None, 14, 14, 736)  0           conv4_block16_0_bn[0][0]
________________________________________________________________________________________________
conv4_block16_1_conv (Conv2D)   (None, 14, 14, 128)  94208       conv4_block16_0_relu[0][0]
________________________________________________________________________________________________
conv4_block16_1_bn (BatchNormal (None, 14, 14, 128)  512         conv4_block16_1_conv[0][0]
________________________________________________________________________________________________
conv4_block16_1_relu (Activatio (None, 14, 14, 128)  0           conv4_block16_1_bn[0][0]
________________________________________________________________________________________________
conv4_block16_2_conv (Conv2D)   (None, 14, 14, 32)   36864       conv4_block16_1_relu[0][0]
________________________________________________________________________________________________
conv4_block16_concat (Concatena (None, 14, 14, 768)  0           conv4_block15_concat[0][0]
                                                                 conv4_block16_2_conv[0][0]
________________________________________________________________________________________________
conv4_block17_0_bn (BatchNormal (None, 14, 14, 768)  3072        conv4_block16_concat[0][0]
________________________________________________________________________________________________
conv4_block17_0_relu (Activatio (None, 14, 14, 768)  0           conv4_block17_0_bn[0][0]
________________________________________________________________________________________________
conv4_block17_1_conv (Conv2D)   (None, 14, 14, 128)  98304       conv4_block17_0_relu[0][0]
________________________________________________________________________________________________
conv4_block17_1_bn (BatchNormal (None, 14, 14, 128)  512         conv4_block17_1_conv[0][0]
________________________________________________________________________________________________
conv4_block17_1_relu (Activatio (None, 14, 14, 128)  0           conv4_block17_1_bn[0][0]
________________________________________________________________________________________________
conv4_block17_2_conv (Conv2D)   (None, 14, 14, 32)   36864       conv4_block17_1_relu[0][0]
________________________________________________________________________________________________
conv4_block17_concat (Concatena (None, 14, 14, 800)  0           conv4_block16_concat[0][0]
                                                                 conv4_block17_2_conv[0][0]
________________________________________________________________________________________________
conv4_block18_0_bn (BatchNormal (None, 14, 14, 800)  3200        conv4_block17_concat[0][0]
________________________________________________________________________________________________
conv4_block18_0_relu (Activatio (None, 14, 14, 800)  0           conv4_block18_0_bn[0][0]
________________________________________________________________________________________________
conv4_block18_1_conv (Conv2D)   (None, 14, 14, 128)  102400      conv4_block18_0_relu[0][0]
________________________________________________________________________________________________
conv4_block18_1_bn (BatchNormal (None, 14, 14, 128)  512         conv4_block18_1_conv[0][0]
________________________________________________________________________________________________
conv4_block18_1_relu (Activatio (None, 14, 14, 128)  0           conv4_block18_1_bn[0][0]
________________________________________________________________________________________________
conv4_block18_2_conv (Conv2D)   (None, 14, 14, 32)   36864       conv4_block18_1_relu[0][0]
________________________________________________________________________________________________
conv4_block18_concat (Concatena (None, 14, 14, 832)  0           conv4_block17_concat[0][0]
                                                                 conv4_block18_2_conv[0][0]
________________________________________________________________________________________________
conv4_block19_0_bn (BatchNormal (None, 14, 14, 832)  3328        conv4_block18_concat[0][0]
```

```
pool4_relu (Activation)         (None, 14, 14, 1024) 0           pool4_bn[0][0]
__________________________________________________________________________________________________
pool4_conv (Conv2D)             (None, 14, 14, 512)  524288      pool4_relu[0][0]
__________________________________________________________________________________________________
pool4_pool (AveragePooling2D)   (None, 7, 7, 512)    0           pool4_conv[0][0]
__________________________________________________________________________________________________
conv5_block1_0_bn (BatchNormali (None, 7, 7, 512)    2048        pool4_pool[0][0]
__________________________________________________________________________________________________
conv5_block1_0_relu (Activation (None, 7, 7, 512)    0           conv5_block1_0_bn[0][0]
__________________________________________________________________________________________________
conv5_block1_1_conv (Conv2D)    (None, 7, 7, 128)    65536       conv5_block1_0_relu[0][0]
__________________________________________________________________________________________________
conv5_block1_1_bn (BatchNormali (None, 7, 7, 128)    512         conv5_block1_1_conv[0][0]
__________________________________________________________________________________________________
conv5_block1_1_relu (Activation (None, 7, 7, 128)    0           conv5_block1_1_bn[0][0]
__________________________________________________________________________________________________
conv5_block1_2_conv (Conv2D)    (None, 7, 7, 32)     36864       conv5_block1_1_relu[0][0]
__________________________________________________________________________________________________
conv5_block1_concat (Concatenat (None, 7, 7, 544)    0           pool4_pool[0][0]
                                                                 conv5_block1_2_conv[0][0]
__________________________________________________________________________________________________
conv5_block2_0_bn (BatchNormali (None, 7, 7, 544)    2176        conv5_block1_concat[0][0]
__________________________________________________________________________________________________
conv5_block2_0_relu (Activation (None, 7, 7, 544)    0           conv5_block2_0_bn[0][0]
__________________________________________________________________________________________________
conv5_block2_1_conv (Conv2D)    (None, 7, 7, 128)    69632       conv5_block2_0_relu[0][0]
__________________________________________________________________________________________________
conv5_block2_1_bn (BatchNormali (None, 7, 7, 128)    512         conv5_block2_1_conv[0][0]
__________________________________________________________________________________________________
conv5_block2_1_relu (Activation (None, 7, 7, 128)    0           conv5_block2_1_bn[0][0]
__________________________________________________________________________________________________
conv5_block2_2_conv (Conv2D)    (None, 7, 7, 32)     36864       conv5_block2_1_relu[0][0]
__________________________________________________________________________________________________
conv5_block2_concat (Concatenat (None, 7, 7, 576)    0           conv5_block1_concat[0][0]
                                                                 conv5_block2_2_conv[0][0]
__________________________________________________________________________________________________
conv5_block3_0_bn (BatchNormali (None, 7, 7, 576)    2304        conv5_block2_concat[0][0]
__________________________________________________________________________________________________
conv5_block3_0_relu (Activation (None, 7, 7, 576)    0           conv5_block3_0_bn[0][0]
__________________________________________________________________________________________________
conv5_block3_1_conv (Conv2D)    (None, 7, 7, 128)    73728       conv5_block3_0_relu[0][0]
__________________________________________________________________________________________________
conv5_block3_1_bn (BatchNormali (None, 7, 7, 128)    512         conv5_block3_1_conv[0][0]
__________________________________________________________________________________________________
conv5_block3_1_relu (Activation (None, 7, 7, 128)    0           conv5_block3_1_bn[0][0]
__________________________________________________________________________________________________
conv5_block3_2_conv (Conv2D)    (None, 7, 7, 32)     36864       conv5_block3_1_relu[0][0]
__________________________________________________________________________________________________
conv5_block3_concat (Concatenat (None, 7, 7, 608)    0           conv5_block2_concat[0][0]
                                                                 conv5_block3_2_conv[0][0]
__________________________________________________________________________________________________
conv5_block4_0_bn (BatchNormali (None, 7, 7, 608)    2432        conv5_block3_concat[0][0]
__________________________________________________________________________________________________
conv5_block4_0_relu (Activation (None, 7, 7, 608)    0           conv5_block4_0_bn[0][0]
```

```
____________________________________________________________________________________________________
conv5_block3_0_bn (BatchNormali (None, 7, 7, 576)    2304        conv5_block2_concat[0][0]
____________________________________________________________________________________________________
conv5_block3_0_relu (Activation (None, 7, 7, 576)    0           conv5_block3_0_bn[0][0]
____________________________________________________________________________________________________
conv5_block3_1_conv (Conv2D)    (None, 7, 7, 128)    73728       conv5_block3_0_relu[0][0]
____________________________________________________________________________________________________
conv5_block3_1_bn (BatchNormali (None, 7, 7, 128)    512         conv5_block3_1_conv[0][0]
____________________________________________________________________________________________________
conv5_block3_1_relu (Activation (None, 7, 7, 128)    0           conv5_block3_1_bn[0][0]
____________________________________________________________________________________________________
conv5_block3_2_conv (Conv2D)    (None, 7, 7, 32)     36864       conv5_block3_1_relu[0][0]
____________________________________________________________________________________________________
conv5_block3_concat (Concatenat (None, 7, 7, 608)    0           conv5_block2_concat[0][0]
                                                                 conv5_block3_2_conv[0][0]
____________________________________________________________________________________________________
conv5_block4_0_bn (BatchNormali (None, 7, 7, 608)    2432        conv5_block3_concat[0][0]
____________________________________________________________________________________________________
conv5_block4_0_relu (Activation (None, 7, 7, 608)    0           conv5_block4_0_bn[0][0]
____________________________________________________________________________________________________
conv5_block4_1_conv (Conv2D)    (None, 7, 7, 128)    77824       conv5_block4_0_relu[0][0]
____________________________________________________________________________________________________
conv5_block4_1_bn (BatchNormali (None, 7, 7, 128)    512         conv5_block4_1_conv[0][0]
____________________________________________________________________________________________________
conv5_block4_1_relu (Activation (None, 7, 7, 128)    0           conv5_block4_1_bn[0][0]
____________________________________________________________________________________________________
conv5_block4_2_conv (Conv2D)    (None, 7, 7, 32)     36864       conv5_block4_1_relu[0][0]
____________________________________________________________________________________________________
conv5_block4_concat (Concatenat (None, 7, 7, 640)    0           conv5_block3_concat[0][0]
                                                                 conv5_block4_2_conv[0][0]
____________________________________________________________________________________________________
conv5_block5_0_bn (BatchNormali (None, 7, 7, 640)    2560        conv5_block4_concat[0][0]
____________________________________________________________________________________________________
conv5_block5_0_relu (Activation (None, 7, 7, 640)    0           conv5_block5_0_bn[0][0]
____________________________________________________________________________________________________
conv5_block5_1_conv (Conv2D)    (None, 7, 7, 128)    81920       conv5_block5_0_relu[0][0]
____________________________________________________________________________________________________
conv5_block5_1_bn (BatchNormali (None, 7, 7, 128)    512         conv5_block5_1_conv[0][0]
____________________________________________________________________________________________________
conv5_block5_1_relu (Activation (None, 7, 7, 128)    0           conv5_block5_1_bn[0][0]
____________________________________________________________________________________________________
conv5_block5_2_conv (Conv2D)    (None, 7, 7, 32)     36864       conv5_block5_1_relu[0][0]
____________________________________________________________________________________________________
conv5_block5_concat (Concatenat (None, 7, 7, 672)    0           conv5_block4_concat[0][0]
                                                                 conv5_block5_2_conv[0][0]
____________________________________________________________________________________________________
conv5_block6_0_bn (BatchNormali (None, 7, 7, 672)    2688        conv5_block5_concat[0][0]
```

tebook   Input   Output   Logs   Comments (0)

```
conv5_block6_1_bn (BatchNormali (None, 7, 7, 128)    512         conv5_block6_1_conv[0][0]
________________________________________________________________________________________________
conv5_block6_1_relu (Activation (None, 7, 7, 128)    0           conv5_block6_1_bn[0][0]
________________________________________________________________________________________________
conv5_block6_2_conv (Conv2D)    (None, 7, 7, 32)     36864       conv5_block6_1_relu[0][0]
________________________________________________________________________________________________
conv5_block6_concat (Concatenat (None, 7, 7, 704)    0           conv5_block5_concat[0][0]
                                                                 conv5_block6_2_conv[0][0]
________________________________________________________________________________________________
conv5_block7_0_bn (BatchNormali (None, 7, 7, 704)    2816        conv5_block6_concat[0][0]
________________________________________________________________________________________________
conv5_block7_0_relu (Activation (None, 7, 7, 704)    0           conv5_block7_0_bn[0][0]
________________________________________________________________________________________________
conv5_block7_1_conv (Conv2D)    (None, 7, 7, 128)    90112       conv5_block7_0_relu[0][0]
________________________________________________________________________________________________
conv5_block7_1_bn (BatchNormali (None, 7, 7, 128)    512         conv5_block7_1_conv[0][0]
________________________________________________________________________________________________
conv5_block7_1_relu (Activation (None, 7, 7, 128)    0           conv5_block7_1_bn[0][0]
________________________________________________________________________________________________
conv5_block7_2_conv (Conv2D)    (None, 7, 7, 32)     36864       conv5_block7_1_relu[0][0]
________________________________________________________________________________________________
conv5_block7_concat (Concatenat (None, 7, 7, 736)    0           conv5_block6_concat[0][0]
                                                                 conv5_block7_2_conv[0][0]
________________________________________________________________________________________________
conv5_block8_0_bn (BatchNormali (None, 7, 7, 736)    2944        conv5_block7_concat[0][0]
________________________________________________________________________________________________
conv5_block8_0_relu (Activation (None, 7, 7, 736)    0           conv5_block8_0_bn[0][0]
________________________________________________________________________________________________
conv5_block8_1_conv (Conv2D)    (None, 7, 7, 128)    94208       conv5_block8_0_relu[0][0]
________________________________________________________________________________________________
conv5_block8_1_bn (BatchNormali (None, 7, 7, 128)    512         conv5_block8_1_conv[0][0]
________________________________________________________________________________________________
conv5_block8_1_relu (Activation (None, 7, 7, 128)    0           conv5_block8_1_bn[0][0]
________________________________________________________________________________________________
conv5_block8_2_conv (Conv2D)    (None, 7, 7, 32)     36864       conv5_block8_1_relu[0][0]
________________________________________________________________________________________________
conv5_block8_concat (Concatenat (None, 7, 7, 768)    0           conv5_block7_concat[0][0]
                                                                 conv5_block8_2_conv[0][0]
________________________________________________________________________________________________
conv5_block9_0_bn (BatchNormali (None, 7, 7, 768)    3072        conv5_block8_concat[0][0]
________________________________________________________________________________________________
conv5_block9_0_relu (Activation (None, 7, 7, 768)    0           conv5_block9_0_bn[0][0]
________________________________________________________________________________________________
conv5_block9_1_conv (Conv2D)    (None, 7, 7, 128)    98304       conv5_block9_0_relu[0][0]
________________________________________________________________________________________________
conv5_block9_1_bn (BatchNormali (None, 7, 7, 128)    512         conv5_block9_1_conv[0][0]
________________________________________________________________________________________________
conv5_block9_1_relu (Activation (None, 7, 7, 128)    0           conv5_block9_1_bn[0][0]
________________________________________________________________________________________________
conv5_block9_2_conv (Conv2D)    (None, 7, 7, 32)     36864       conv5_block9_1_relu[0][0]
________________________________________________________________________________________________
conv5_block9_concat (Concatenat (None, 7, 7, 800)    0           conv5_block8_concat[0][0]
                                                                 conv5_block9_2_conv[0][0]
________________________________________________________________________________________________
```

```
                                                                 conv5_block14_2_conv[0][0]
__________________________________________________________________________________________________
conv5_block15_0_bn (BatchNormal (None, 7, 7, 960)    3840        conv5_block14_concat[0][0]
__________________________________________________________________________________________________
conv5_block15_0_relu (Activatio (None, 7, 7, 960)    0           conv5_block15_0_bn[0][0]
__________________________________________________________________________________________________
conv5_block15_1_conv (Conv2D)   (None, 7, 7, 128)    122880      conv5_block15_0_relu[0][0]
__________________________________________________________________________________________________
conv5_block15_1_bn (BatchNormal (None, 7, 7, 128)    512         conv5_block15_1_conv[0][0]
__________________________________________________________________________________________________
conv5_block15_1_relu (Activatio (None, 7, 7, 128)    0           conv5_block15_1_bn[0][0]
__________________________________________________________________________________________________
conv5_block15_2_conv (Conv2D)   (None, 7, 7, 32)     36864       conv5_block15_1_relu[0][0]
__________________________________________________________________________________________________
conv5_block15_concat (Concatena (None, 7, 7, 992)    0           conv5_block14_concat[0][0]
                                                                 conv5_block15_2_conv[0][0]
__________________________________________________________________________________________________
conv5_block16_0_bn (BatchNormal (None, 7, 7, 992)    3968        conv5_block15_concat[0][0]
__________________________________________________________________________________________________
conv5_block16_0_relu (Activatio (None, 7, 7, 992)    0           conv5_block16_0_bn[0][0]
__________________________________________________________________________________________________
conv5_block16_1_conv (Conv2D)   (None, 7, 7, 128)    126976      conv5_block16_0_relu[0][0]
__________________________________________________________________________________________________
conv5_block16_1_bn (BatchNormal (None, 7, 7, 128)    512         conv5_block16_1_conv[0][0]
__________________________________________________________________________________________________
conv5_block16_1_relu (Activatio (None, 7, 7, 128)    0           conv5_block16_1_bn[0][0]
__________________________________________________________________________________________________
conv5_block16_2_conv (Conv2D)   (None, 7, 7, 32)     36864       conv5_block16_1_relu[0][0]
__________________________________________________________________________________________________
conv5_block16_concat (Concatena (None, 7, 7, 1024)   0           conv5_block15_concat[0][0]
                                                                 conv5_block16_2_conv[0][0]
__________________________________________________________________________________________________
bn (BatchNormalization)         (None, 7, 7, 1024)   4096        conv5_block16_concat[0][0]
__________________________________________________________________________________________________
relu (Activation)               (None, 7, 7, 1024)   0           bn[0][0]
__________________________________________________________________________________________________
global_average_pooling2d_1 (Glo (None, 1024)         0           relu[0][0]
__________________________________________________________________________________________________
dense_1 (Dense)                 (None, 256)          262400      global_average_pooling2d_1[0][0]
__________________________________________________________________________________________________
dense_2 (Dense)                 (None, 5)            1285        dense_1[0][0]
==================================================================================================
Total params: 7,301,189
Trainable params: 7,217,541
Non-trainable params: 83,648
__________________________________________________________________________________________________
None
```

The code configures and trains a DenseNet121-based model (mymodel) for a specified number of epochs with early stopping, learning rate reduction on plateau, model checkpointing, and a custom Quadratic Weighted Kappa (QWK) callback for monitoring

```
EPOCHS = 50

earlystop = keras.callbacks.EarlyStopping(patience=10)

learning_rate_reduction =
keras.callbacks.ReduceLROnPlateau(monitor='val_acc',

                                            patience=2,
```

```
                                            verbose=1,

                                            factor=0.5,

                                            min_lr=0.00001)

checkpoint =
keras.callbacks.ModelCheckpoint('../working/DenseNet121.h5',
monitor='val_loss', verbose=1,

                              save_best_only=True, mode='min',
save_weights_only = True)

qwk = QWKCallback((X_val,Y_val))

mycallbacks = [earlystop, learning_rate_reduction,checkpoint,qwk]
```

```
print(qwk)
```

```
<__main__.QWKCallback object at 0x7f540ab3fd68>
```

```
# Warm up the model with class weights

EPOCHS = 10

history = mymodel.fit_generator(training_generator,steps_per_epoch =
X_train.shape[0] // batch_size,epochs = EPOCHS,

                         validation_data = (X_val,Y_val),

                         validation_steps = 10,

                         workers = 2,use_multiprocessing=True,

                         verbose=2, callbacks=mycallbacks,

                         class_weight=cls_wt_dict)
```

```
Epoch 1/10

/opt/conda/lib/python3.6/site-packages/keras/engine/training_generator.py:47: Use
rWarning: Using a generator with `use_multiprocessing=True` and multiple workers
may duplicate your data. Please consider using the`keras.utils.Sequence class.
  UserWarning('Using a generator with `use_multiprocessing=True`'


 - 81s - loss: 1.1908 - acc: 0.6900 - val_loss: 1.1318 - val_acc: 0.5861

Epoch 00001: val_loss improved from inf to 1.13182, saving model to ../working/De
nseNet121.h5
Epoch 0 : QWK: 0.6606347581629278
saving checkpoint:  0.6606347581629278
Epoch 2/10
 - 47s - loss: 1.1552 - acc: 0.6961 - val_loss: 1.0400 - val_acc: 0.5836

Epoch 00002: val_loss improved from 1.13182 to 1.04000, saving model to ../workin
g/DenseNet121.h5
Epoch 1 : QWK: 0.4967983145414807
Epoch 3/10
 - 47s - loss: 1.0514 - acc: 0.7447 - val_loss: 0.9467 - val_acc: 0.6500

Epoch 00003: val_loss improved from 1.04000 to 0.94669, saving model to ../workin
g/DenseNet121.h5
Epoch 2 : QWK: 0.606153805192736
Epoch 4/10
 - 47s - loss: 1.0690 - acc: 0.7340 - val_loss: 1.4204 - val_acc: 0.5098

Epoch 00004: val_loss did not improve from 0.94669
Epoch 3 : QWK: 0.673388012318533
saving checkpoint:  0.673388012318533
Epoch 5/10
 - 47s - loss: 1.0158 - acc: 0.7578 - val_loss: 1.0547 - val_acc: 0.6025

Epoch 00005: ReduceLROnPlateau reducing learning rate to 0.004999999888241291.

Epoch 00005: val_loss did not improve from 0.94669
Epoch 4 : QWK: 0.40468768771292674
Epoch 6/10
 - 46s - loss: 1.0538 - acc: 0.7208 - val_loss: 0.9260 - val_acc: 0.6270
```

```
Epoch 00006: val_loss improved from 0.94669 to 0.92603, saving model to ../workin
g/DenseNet121.h5
Epoch 5 : QWK: 0.6010582954308986
Epoch 7/10
 - 47s - loss: 0.9715 - acc: 0.7558 - val_loss: 0.8224 - val_acc: 0.6836

Epoch 00007: val_loss improved from 0.92603 to 0.82242, saving model to ../workin
g/DenseNet121.h5
Epoch 6 : QWK: 0.8450581787790845
saving checkpoint:  0.8450581787790845
Epoch 8/10
 - 46s - loss: 0.8936 - acc: 0.7887 - val_loss: 0.9989 - val_acc: 0.6525

Epoch 00008: val_loss did not improve from 0.82242
Epoch 7 : QWK: 0.7028776443240523
Epoch 9/10
 - 47s - loss: 0.8548 - acc: 0.7718 - val_loss: 0.9379 - val_acc: 0.6959

Epoch 00009: val_loss did not improve from 0.82242
Epoch 8 : QWK: 0.7600499276290307
Epoch 10/10
 - 46s - loss: 0.9754 - acc: 0.7763 - val_loss: 0.9158 - val_acc: 0.6721

Epoch 00010: val_loss did not improve from 0.82242
Epoch 9 : QWK: 0.7679516452608758
```

```
EPOCHS = 50

history = mymodel.fit_generator(training_generator,steps_per_epoch =
X_train.shape[0] // batch_size,epochs = EPOCHS,

                         validation_data = (X_val,Y_val),

                         validation_steps = 10,

                         workers = 2,use_multiprocessing=True,

                         verbose=2, callbacks=mycallbacks)
```

```
Epoch 1/50
 - 53s - loss: 0.7034 - acc: 0.8076 - val_loss: 0.6413 - val_acc: 0.7943

Epoch 00001: val_loss improved from 0.82242 to 0.64127, saving model to ../workin
g/DenseNet121.h5
Epoch 0 : QWK: 0.8430588862243539
Epoch 2/50
 - 47s - loss: 0.6261 - acc: 0.8326 - val_loss: 0.6528 - val_acc: 0.7828

Epoch 00002: val_loss did not improve from 0.64127
Epoch 1 : QWK: 0.8516125400571757
saving checkpoint:  0.8516125400571757
Epoch 3/50
 - 47s - loss: 0.6396 - acc: 0.8043 - val_loss: 0.7131 - val_acc: 0.7623

Epoch 00003: ReduceLROnPlateau reducing learning rate to 0.0024999999441206455.

Epoch 00003: val_loss did not improve from 0.64127
Epoch 2 : QWK: 0.8414919398212692
Epoch 4/50
 - 46s - loss: 0.6370 - acc: 0.8306 - val_loss: 0.5923 - val_acc: 0.8189

Epoch 00004: val_loss improved from 0.64127 to 0.59228, saving model to ../workin
g/DenseNet121.h5
Epoch 3 : QWK: 0.880622590197721
saving checkpoint:  0.880622590197721
Epoch 5/50
 - 47s - loss: 0.5706 - acc: 0.8483 - val_loss: 0.5980 - val_acc: 0.8057

Epoch 00005: val_loss did not improve from 0.59228
Epoch 4 : QWK: 0.8866740469412809
saving checkpoint:  0.8866740469412809
Epoch 6/50
 - 47s - loss: 0.5910 - acc: 0.8368 - val_loss: 0.5928 - val_acc: 0.8139

Epoch 00006: ReduceLROnPlateau reducing learning rate to 0.0012499999720603228.

Epoch 00006: val_loss did not improve from 0.59228
Epoch 5 : QWK: 0.8744554679721348
Epoch 7/50
 - 47s - loss: 0.5729 - acc: 0.8516 - val_loss: 0.5930 - val_acc: 0.8082

Epoch 00007: val_loss did not improve from 0.59228
```

```
Epoch 00023: ReduceLROnPlateau reducing learning rate to 1e-05.

Epoch 00023: val_loss did not improve from 0.56789
Epoch 22 : QWK: 0.8981812987582019
Epoch 24/50
 - 47s - loss: 0.5496 - acc: 0.8540 - val_loss: 0.5692 - val_acc: 0.8238

Epoch 00024: val_loss did not improve from 0.56789
Epoch 23 : QWK: 0.8979799827527635
Epoch 25/50
 - 47s - loss: 0.5783 - acc: 0.8442 - val_loss: 0.5699 - val_acc: 0.8254

Epoch 00025: val_loss did not improve from 0.56789
Epoch 24 : QWK: 0.8981707919444686
Epoch 26/50
 - 46s - loss: 0.5583 - acc: 0.8598 - val_loss: 0.5698 - val_acc: 0.8205

Epoch 00026: val_loss did not improve from 0.56789
Epoch 25 : QWK: 0.8952147968743528
Epoch 27/50
 - 46s - loss: 0.5248 - acc: 0.8660 - val_loss: 0.5682 - val_acc: 0.8246

Epoch 00027: val_loss did not improve from 0.56789
Epoch 26 : QWK: 0.8987799381934161
Epoch 28/50
 - 46s - loss: 0.5546 - acc: 0.8586 - val_loss: 0.5679 - val_acc: 0.8254

Epoch 00028: val_loss did not improve from 0.56789
Epoch 27 : QWK: 0.8989255166808576
Epoch 29/50
 - 47s - loss: 0.5729 - acc: 0.8553 - val_loss: 0.5682 - val_acc: 0.8238

Epoch 00029: val_loss did not improve from 0.56789
Epoch 28 : QWK: 0.8984336700467013
```

```
mymodel.save_weights("model.h5")
```

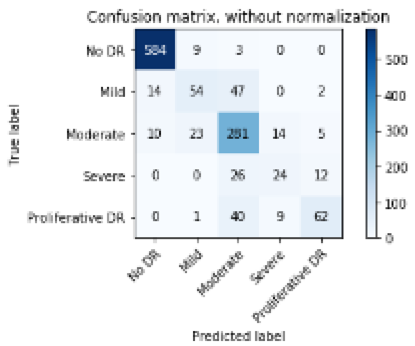
```
Y_val_pred = mymodel.predict_on_batch(X_val)
```

```
Y_val_pred_hot = np.argmax(Y_val_pred,axis=1)

Y_val_actual_hot = np.argmax(Y_val,axis=1)
```

```
plot_confusion_matrix(Y_val_actual_hot, Y_val_pred_hot,
np.array(class_labels))
```

```
Confusion matrix, without normalization
[[584   9   3   0   0]
 [ 14  54  47   0   2]
 [ 10  23 281  14   5]
 [  0   0  26  24  12]
 [  0   1  40   9  62]]
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f5403b3c978>
```



This code section is designed for evaluating a model on validation data, generating predictions, and visualizing the confusion matrix. Uncomment and adapt as needed for your specific use case.

```
# history = mymodel.fit_generator(train_gen,steps_per_epoch =
10,epochs = EPOCHS,
#                            validation_data = val_gen,
#                            validation_steps = 10,
#                            workers = 2,use_multiprocessing=True,
#                       verbose=2, callbacks=mycallbacks)
# mymodel.save_weights("model.h5")
```

```
# train_gen = img_gen.flow_from_dataframe(
#      dataframe=train_data,
#
directory="../input/aptos2019-blindness-detection/train_images",
#      x_col="filename",
#      y_col="diagnosis",
#      batch_size=batch_size,
#      shuffle=True,
```

```
#     class_mode="categorical",

#     classes=labels,

#     target_size=(img_size,img_size),

#     subset='training')
```

The code initializes a validation data generator (val_gen) using image data from the dataset. It specifies parameters such as the data frame (train_data), image directory, batch size, class mode, and target size.

```
# val_gen = img_gen.flow_from_dataframe(

#     dataframe=train_data,

#
directory="../input/aptos2019-blindness-detection/train_images",

#     x_col="filename",

#     y_col="diagnosis",

#     batch_size=batch_size,

#     shuffle=True,

#     class_mode="categorical",

#     classes=labels,

#     target_size=(img_size,img_size),

#     subset='validation'

# )

# # dir(val_gen)
```

```
# class QWK(keras.callbacks.Callback):

#     def on_train_begin(self, logs={}):

#         self.val_kappas = []
```

```
#     def on_epoch_end(self, epoch, logs={}):
#         print(self.__dict__)
#         X_val, y_val = self.model.validation_data[:2]
#         y_pred = self.model.predict(X_val)

#         _val_kappa = cohen_kappa_score(
#             y_val.argmax(axis=1),
#             y_pred.argmax(axis=1),
#             weights='quadratic'
#         )

#         self.val_kappas.append(_val_kappa)

#         print(f"epoch: {epoch} \t val_kappa: {_val_kappa:.4f}")

#         return
```

```
# val_gen = img_gen.flow_from_dataframe(
#     dataframe=train_data,
#
directory="../input/aptos2019-blindness-detection/train_images",
#     x_col="filename",
#     y_col="diagnosis",
#     batch_size=1000,
#     shuffle=True,
#     class_mode="categorical",
#     classes=labels,
```

```
#     target_size=(img_size,img_size),
#     subset='validation'
# )
# # print(val_gen.batch_index)
# # for i in range(val_gen.batch_index):
# batch_data, batch_labels = val_gen.next()
# batch_labels_arg = np.argmax(batch_labels,axis=1)
# # print(batch_data.shape)
```

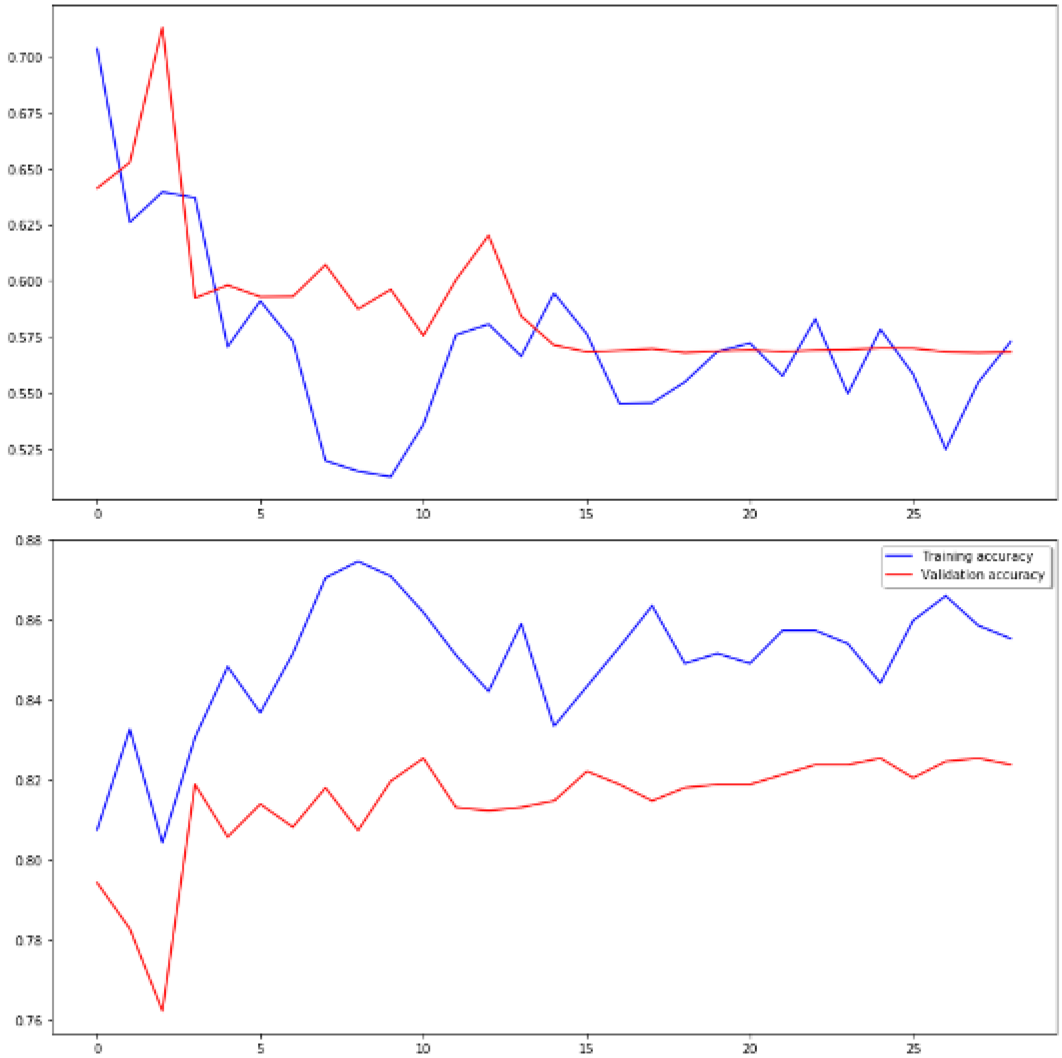
```
# cur_batch_size = batch_data.shape[0]
# preds = np.empty((cur_batch_size,5))
# preds_one_hot = np.zeros((cur_batch_size,5))
# for i in range(cur_batch_size):
#     preds[i,:] = mymodel.predict(batch_data[i,:,:,:][np.newaxis,:])
# args_max = np.argmax(preds,axis=1)
# preds_one_hot[np.arange(cur_batch_size),args_max] = 1
```

```
# plot_confusion_matrix(batch_labels_arg, args_max,
np.array(class_labels))
```

The code creates a subplot with two vertical axes (ax1 and ax2) to visualize training and validation loss, as well as training and validation accuracy over epochs using data stored in history

```
fig, (ax1, ax2) = plt.subplots(2, 1, figsize=(12, 12))
ax1.plot(history.history['loss'], color='b', label="Training loss")
```

```
ax1.plot(history.history['val_loss'], color='r', label="validation loss")

# ax1.set_xticks(np.arange(1, epochs, 1))

# ax1.set_yticks(np.arange(0, 1, 0.1))


ax2.plot(history.history['acc'], color='b', label="Training accuracy")

ax2.plot(history.history['val_acc'], color='r',label="Validation accuracy")

# ax2.set_xticks(np.arange(1, epochs, 1))


legend = plt.legend(loc='best', shadow=True)

plt.tight_layout()

plt.show()
```




```
test_data =
pd.read_csv("../input/aptos2019-blindness-detection/test.csv")

test_data["filename"] = test_data["id_code"].map(lambda x:x+".png")

test_data.head()
```

	id_code	filename
0	0005cfc8afb6	0005cfc8afb6.png
1	003f0afdcd15	003f0afdcd15.png
2	006efc72b638	006efc72b638.png
3	00836aaacf06	00836aaacf06.png
4	009245722fa4	009245722fa4.png

```
test_gen = ImageDataGenerator(rescale=1./255)
```

```
test_generator = test_gen.flow_from_dataframe(
        dataframe=test_data,
        directory =
"../input/aptos2019-blindness-detection/test_images",
        x_col="filename",
        y_col=None,
        target_size = (img_size,img_size),
        batch_size = 1,
        shuffle = False,
        class_mode = None
        )
```

```
Found 1928 validated image filenames.
```

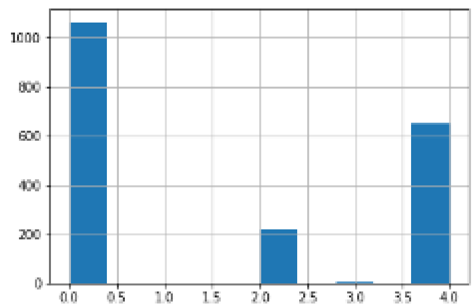
```
predictions = mymodel.predict_generator(test_generator, steps =
len(test_generator.filenames))
```

```
filenames=test_generator.filenames
results=pd.DataFrame({"id_code":filenames,
                      "diagnosis":np.argmax(predictions,axis=1)})
results['id_code'] = results['id_code'].map(lambda x: str(x)[:-4])
results.to_csv("submission.csv",index=False)
```

submission.csv

```
results.diagnosis.hist()
results.diagnosis.unique()
```

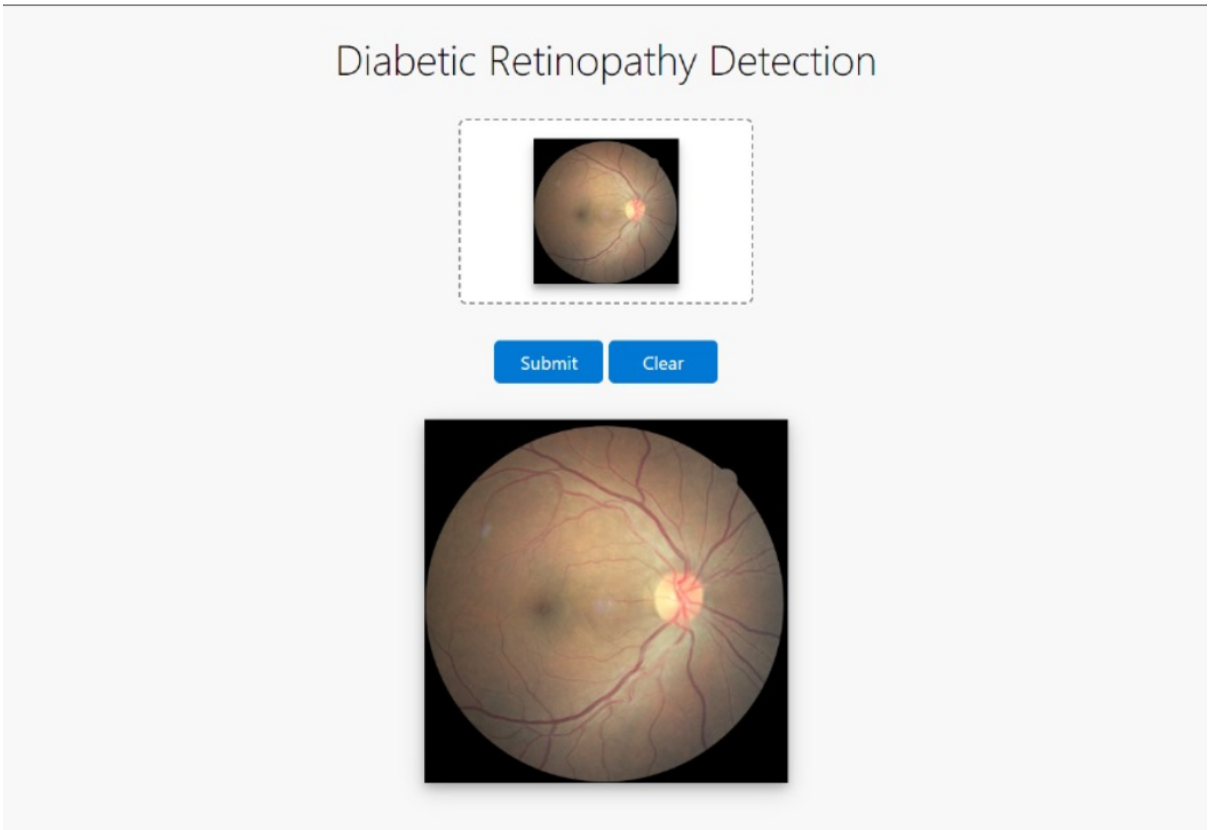
```
array([0, 4, 2, 3])
```

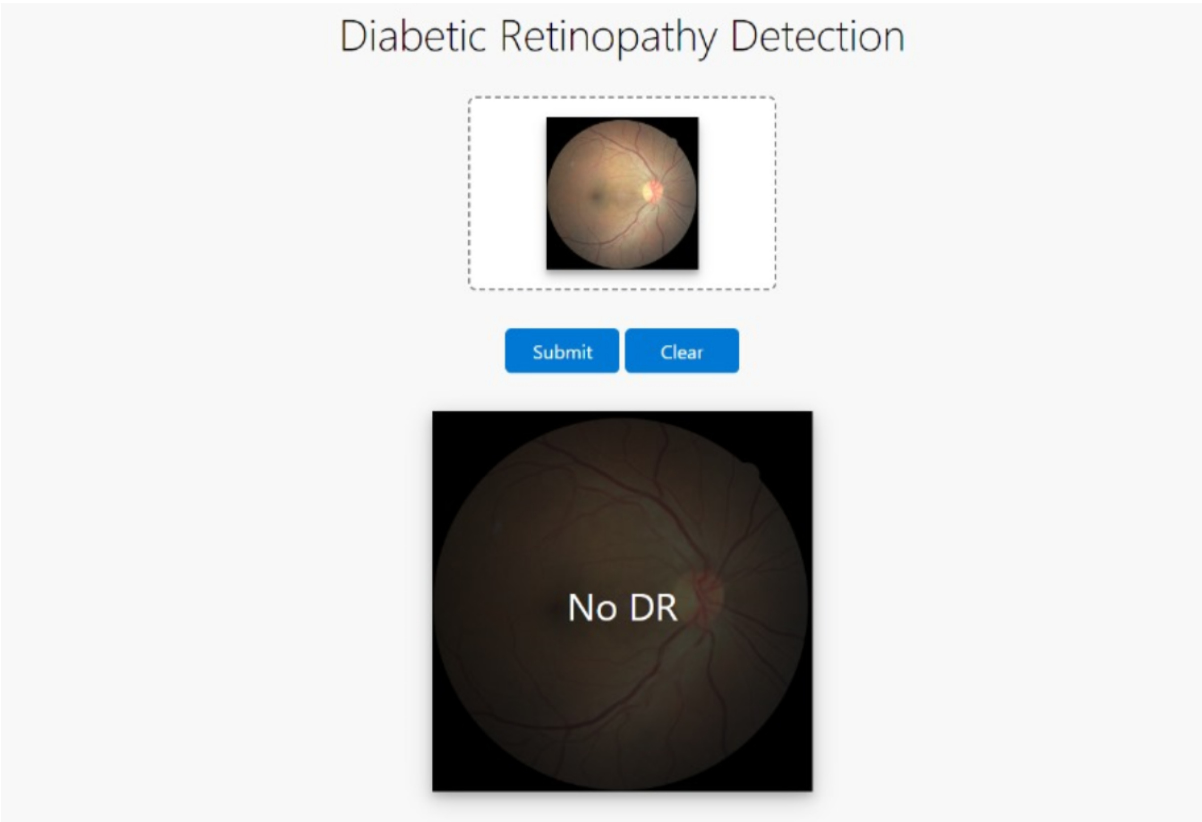



```
Epoch 25 : QWK: 0.8952147968743528
Epoch 27/50
 - 46s - loss: 0.5248 - acc: 0.8660 - val_loss: 0.5682 - val_acc: 0.8246

Epoch 00027: val_loss did not improve from 0.56789
Epoch 26 : QWK: 0.8987799381934161
Epoch 28/50
 - 46s - loss: 0.5546 - acc: 0.8586 - val_loss: 0.5679 - val_acc: 0.8254

Epoch 00028: val_loss did not improve from 0.56789
Epoch 27 : QWK: 0.8989255166808576
Epoch 29/50
 - 47s - loss: 0.5729 - acc: 0.8553 - val_loss: 0.5682 - val_acc: 0.8238

Epoch 00029: val_loss did not improve from 0.56789
Epoch 28 : QWK: 0.8984336700467013
```

WEBPAGE:

After the user provides the input retinal image, the image is fed into the DR Model. The DR Model then analyzes the image and makes predictions based on the data it has been trained on. The predicted model is then saved for further use or reference. The Flask application displays reports indicating the severity of diabetic retinopathy (DR) to the users. These reports are generated after processing the uploaded fundal image. Additionally, the Flask application also displays the processed fundal image along with the severity assessment report.


Diabetic Retinopathy Detection
Submit
Clear


Diabetic Retinopathy Detection
Submit
Clear
No DR

8. PERFORMANCE TESTING

```
from sklearn.metrics import classification_report
classification_rep = classification_report(Y_val_actual_hot, Y_val_pred_hot, target_names=class_labels)
print(classification_rep)
```

	precision	recall	f1-score	support
No DR	0.96	0.98	0.97	596
Mild	0.61	0.47	0.53	117
Moderate	0.71	0.83	0.77	333
Severe	0.46	0.39	0.42	62
Proliferative DR	0.73	0.52	0.61	112
accuracy			0.82	1220
macro avg	0.70	0.64	0.66	1220
weighted avg	0.81	0.82	0.81	1220

9. RESULTS

9.1 Output Screenshots:

With an overall accuracy of 0.89, the model demonstrates effective classification performance. The model's strong accuracy implies its capability to predict diabetic retinopathy accurately on the test dataset. The recall of the classification report in the test is 0.97

```
Epoch 28 : QWK: 0.8984336700467013
```

10. ADVANTAGES & DISADVANTAGES

Advantages:

Early Detection: Enables early identification of diabetic retinopathy, facilitating timely intervention and treatment to prevent vision loss.

Accuracy: Utilizes deep learning algorithms, providing highly accurate diagnoses by analyzing retinal images with precision.

Efficiency: Automates the screening process, reducing the workload on ophthalmologists and enabling faster diagnoses for a larger number of patients.

Improved Patient Care: Enhances patient care by offering rapid and reliable diagnostic reports, aiding in personalized treatment planning.

Cost-Effectiveness: Reduces healthcare costs associated with manual screening and late-stage interventions by detecting retinopathy at its earlier stages.

Accessibility: Allows for remote screening and diagnosis, particularly beneficial in regions with limited access to healthcare facilities or specialists.

Potential for Integration: Can be integrated into existing healthcare systems, facilitating seamless patient data management.

Scalability: Adaptable to handle increasing volumes of patient data and retinal images, accommodating the growing demand for diabetic retinopathy screening.

Research Opportunities: Provides a vast repository of data for research purposes, aiding in the advancement of diabetic retinopathy studies and treatment strategies.

Public Health Impact: Contributes to public health initiatives by preventing vision impairment and blindness due to diabetic retinopathy.

Disadvantages:

Dependency on Imaging Quality: Accuracy heavily relies on the quality of acquired retinal images; poor-quality images might affect the system's performance.

Algorithm Bias: Algorithms might demonstrate biases based on the dataset used for training, leading to inaccuracies or misdiagnoses in certain cases.

Regulatory Challenges: Compliance with healthcare regulations and ethical considerations regarding patient data privacy and usage can pose challenges.

Technical Complexity: Developing and maintaining a deep learning-based system requires specialized skills and ongoing technical expertise.

Initial Investment: Implementation and setup costs, including infrastructure, software, and training, might be substantial initially.

Limited Generalization: The system's accuracy might vary based on patient diversity, as certain populations or demographic factors could influence its effectiveness.

Integration Hurdles: Integrating the system with existing healthcare systems might present interoperability challenges, requiring additional efforts for seamless integration.

Continual Updates and Maintenance: Regular updates and maintenance are necessary to keep the system efficient, accurate, and secure, requiring ongoing resources and support.

Ethical Concerns: Ethical dilemmas regarding the responsibility of decision-making between the automated system and healthcare professionals may arise.

Risk of Misinterpretation: There is a risk of misinterpretation or over-reliance on the system's output without appropriate validation or clinical judgment.

11. CONCLUSION:

In conclusion, the development and implementation of a Diabetic Retinopathy Detection System leveraging deep learning techniques present a significant leap forward in early diagnosis and intervention for diabetic retinopathy. This automated system offers promising prospects in revolutionizing the way retinal images are analyzed, providing accurate and timely identification of retinopathy stages. The amalgamation of machine learning algorithms with medical imaging not only enhances the efficiency of diagnoses but also opens avenues for personalized patient care. Despite challenges in image quality and algorithm biases, the system's potential in preventing vision loss and improving patient outcomes cannot be understated. Collaborative efforts between technologists, healthcare practitioners, and regulatory bodies are crucial to address challenges, ensure ethical compliance, and further enhance the system's accuracy and accessibility.

12.Future Scope:

- The future scope for the Diabetic Retinopathy Detection System is vast and multifaceted. Some potential avenues for further development and improvement include:
- Enhanced Algorithm Refinement: Continuous refinement of deep learning algorithms to reduce biases, improve accuracy, and handle a more diverse range of retinal images.
- Integration with Telemedicine: Integrating the system with telemedicine platforms for remote patient screening, especially in underserved areas with limited access to specialists.
- Extended Diagnostic Capabilities: Expanding the system's capabilities to detect and classify other ocular diseases or abnormalities from retinal images.
- Longitudinal Data Analysis: Implementing tools for longitudinal analysis of patient data to monitor disease progression, treatment response, and long-term outcomes.
- Real-time Decision Support: Advancing the system to offer real-time diagnostic support to healthcare professionals during patient consultations.
- Ethical and Regulatory Compliance: Striving for adherence to evolving healthcare regulations, privacy laws, and ethical considerations while handling patient data.
- Collaborative Research: Encouraging collaboration among medical researchers, data scientists, and healthcare providers for collective efforts in refining the system's accuracy and efficacy.
- User Interface Enhancement: Focusing on user-friendly interfaces and seamless integration into existing healthcare systems to facilitate widespread adoption and usability.

13. APPENDIX Source Code GitHub & Project Demo Link

Github files:

https://github.com/smartinternz02/SI-GuidedProject-612348-1698775450/tree/main

Demo Link: