

# Retinal OCT Images (optical coherence tomography)

## 1. Business Problem

### 1.1 Description

Retinal optical coherence tomography (OCT) is an imaging technique used to capture high-resolution cross sections of the retinas of living patients. Approximately 30 million OCT scans are performed each year, and the analysis and interpretation of these images takes up a significant amount of time (Swanson and Fujimoto, 2017).

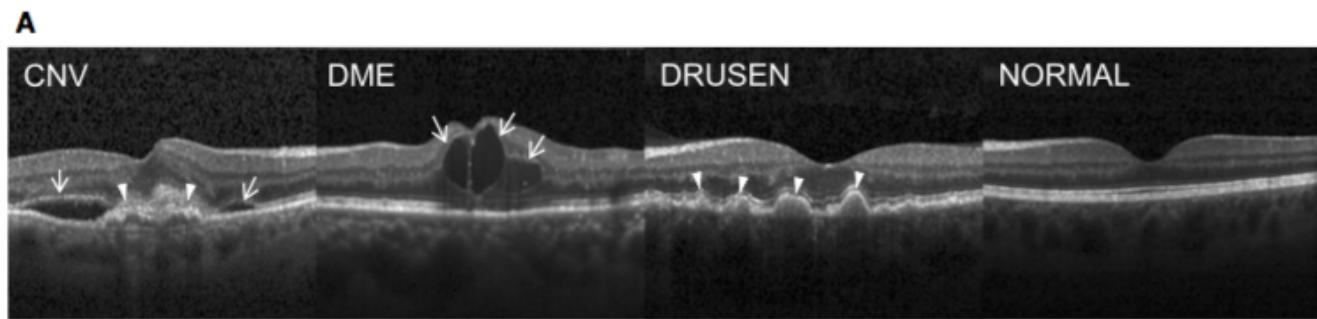


Figure 2. Representative Optical Coherence Tomography Images and the Workflow Diagram [Kermany et. al. 2018]

(A) (Far left) choroidal neovascularization (CNV) with neovascular membrane (white arrowheads) and associated subretinal fluid (arrows). (Middle left) Diabetic macular edema (DME) with retinal-thickening-associated intraretinal fluid (arrows). (Middle right) Multiple drusen (arrowheads) present in early AMD. (Far right) Normal retina with preserved foveal contour and absence of any retinal fluid/edema.

### 1.2. Data Description

The dataset is organized into 3 folders (train, test, val) and contains subfolders for each image category (NORMAL,CNV,DME,DRUSEN). There are 84,495 X-Ray images (JPEG) and 4 categories (NORMAL,CNV,DME,DRUSEN).

Images are labeled as (disease)-(randomized patient ID)-(image number by this patient) and split into 4 directories: CNV, DME, DRUSEN, and NORMAL.

Optical coherence tomography (OCT) images (Spectralis OCT, Heidelberg Engineering, Germany) were selected from retrospective cohorts of adult patients from the Shiley Eye Institute of the University of California San Diego, the California Retinal Research Foundation, Medical Center Ophthalmology Associates, the Shanghai First People's Hospital, and Beijing Tongren Eye Center between July 1, 2013 and March 1, 2017.

Before training, each image went through a tiered grading system consisting of multiple layers of trained graders of increasing expertise for verification and correction of image labels. Each image imported into the database started with a label matching the most recent diagnosis of the patient. The first tier of graders consisted of undergraduate and medical students who had taken and passed an OCT interpretation course review. This first tier of graders conducted initial quality control and excluded OCT images containing severe artifacts or significant image resolution reductions. The second tier of graders consisted of four ophthalmologists who independently graded each image that had passed the first tier. The presence or absence of choroidal neovascularization (active or in the form of subretinal fibrosis), macular edema, drusen, and other pathologies visible on the OCT scan were recorded. Finally, a third tier of two senior independent retinal specialists, each with over 20 years of clinical retina experience, verified the true labels for each image. The dataset selection and stratification process is displayed in a CONSORT-style diagram in Figure 2B. To account for human error in grading, a validation subset of 993 scans was graded separately by two ophthalmologist graders, with disagreement in clinical labels arbitrated by a senior retinal specialist. For additional information: see [http://www.cell.com/cell/fulltext/S0092-8674\(18\)30154-5](http://www.cell.com/cell/fulltext/S0092-8674(18)30154-5)

## Problem Statement

- Correctly classify Retinal OCT(optical coherence tomography) Images

## 1.2 Source/Useful links

### 1.2.1. Data Source

**Source :** <https://www.kaggle.com/paultimothymooney/kermany2018>  
<https://www.kaggle.com/paultimothymooney/kermany2018>

### 1.2.2. Reference

- [http://www.cell.com/cell/fulltext/S0092-8674\(18\)30154-5](http://www.cell.com/cell/fulltext/S0092-8674(18)30154-5) ([http://www.cell.com/cell/fulltext/S0092-8674\(18\)30154-5](http://www.cell.com/cell/fulltext/S0092-8674(18)30154-5))
- <https://www.kaggle.com/paultimothymooney/kermany2018> (<https://www.kaggle.com/paultimothymooney/kermany2018>)

## 1.3. Real World/Business Objectives and Constraints

- Accurately predict Retinal optical coherence tomography(OCT).
- No low latency constraint.
- Errors can be very costly.
- Probability of a data-point belonging to each class is needed.

## 2. Machine Learning Problem Formulation

### 2.1. Data

- Source : <https://www.kaggle.com/paultimothymooney/kermany2018> (<https://www.kaggle.com/paultimothymooney/kermany2018>)
- We have two data folders train and test.
- train folder contains 83484 images and test folder contains 1000 images

### 2.2. Mapping the real-world problem to an ML problem

#### 2.2.1. Type of machine learning problem

- There are four different classes CNV, DME, DRUSEN, NORMAL => Multi-class classification problem

#### 2.2.2. Performance Metric

Metric(s) :-

- Micro-avg f1-score
- Confusion Matrix

### 2.2.3. Machine Learning Objectives and Constraints

**Objective:** Predict the probability of each data-point belonging to each of the four classes.

**Constraints:**

- Interpretability
- Class probabilities are needed.
- Penalize the errors in class probabilities => Metric is Log-loss.
- No Latency constraints.

## 3. Exploratory Data Analysis

```
In [1]: import warnings
warnings.filterwarnings("ignore")
```

```
In [2]: import datetime
from datetime import date
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
import shutil
import pickle
# import imageio

%matplotlib inline
import matplotlib.pyplot as plt
import seaborn as sns
# plt.rcParams["figure.figsize"] = (12, 8) # (w, h)
# from tqdm import tqdm
import pickle
#!/find . -name ".DS_Store" -print -delete
```

```
In [3]: from __future__ import print_function
import keras
from keras.datasets import mnist
from keras.models import Sequential
from keras.layers import Dense, Dropout, Flatten
from keras.layers import Conv2D, MaxPooling2D
from tensorflow.python.keras.applications import ResNet50
from keras.utils import np_utils
from keras.initializers import RandomNormal
from keras.layers.normalization import BatchNormalization
from keras.callbacks import ModelCheckpoint, LearningRateScheduler, TensorBoard, EarlyStopping, ReduceLROnPlateau

import cv2
import tensorflow as tf

from keras.layers import *
# from keras.models import Sequential
from keras.applications.resnet50 import ResNet50
from keras.applications.vgg16 import VGG16
from keras.applications.vgg19 import VGG19
from keras.applications.inception_v3 import InceptionV3
from keras.applications.mobilenet import MobileNet
from keras.applications.mobilenet_v2 import MobileNetV2
from keras.applications.densenet import DenseNet121
from keras.models import Model
from keras.layers import Input
from keras.layers import Dense
import os

from keras import backend as K
```

Using TensorFlow backend.

In [4]: # Importing all the needed modules.

```
import os
from glob import glob
import matplotlib.pyplot as plt
import random
import cv2
import pandas as pd
import numpy as np
import matplotlib.gridspec as gridspec
import seaborn as sns
import zlib
import itertools
import sklearn
import itertools
import scipy
import skimage
from skimage.transform import resize
import csv
# from tqdm import tqdm
import warnings
warnings.filterwarnings("ignore")
from sklearn import model_selection
from sklearn.model_selection import train_test_split, KFold, cross_val_score,
StratifiedKFold, GridSearchCV
from sklearn.utils import class_weight
from sklearn.metrics import confusion_matrix, make_scorer, accuracy_score, classification_report
import keras
from keras.layers import Dense, Dropout, Activation, Flatten, Conv2D, MaxPooling2D, Lambda, MaxPool2D, BatchNormalization
from keras.utils import np_utils
from keras.utils.np_utils import to_categorical
from keras.preprocessing.image import ImageDataGenerator
from keras import models, layers, optimizers
from sklearn.model_selection import train_test_split
from sklearn.metrics import confusion_matrix, accuracy_score
from sklearn.utils import class_weight
from keras.optimizers import SGD, RMSprop, Adam, Adagrad, Adadelta, RMSprop
from keras.models import Sequential, model_from_json
from keras.layers import Activation,Dense, Dropout, Flatten, Conv2D, MaxPool2D
from keras.layers import MaxPooling2D,AveragePooling2D, GlobalAveragePooling2D
,BatchNormalization
from keras.preprocessing.image import array_to_img, img_to_array, load_img, ImageDataGenerator
from keras.callbacks import ReduceLROnPlateau, ModelCheckpoint
from keras import backend as K
from keras.applications.vgg16 import VGG16
from keras.models import Model
from keras.applications.mobilenet import MobileNet
from keras.applications.inception_v3 import InceptionV3
# from imblearn.over_sampling import RandomOverSampler
from sklearn.metrics import roc_auc_score
from sklearn.metrics import roc_curve
from sklearn.metrics import auc
from keras.models import load_model
from keras.utils import np_utils
```

```

from sklearn.metrics import log_loss

%matplotlib inline

In [5]: import tensorflow as tf
config = tf.ConfigProto()
config.gpu_options.allow_growth=True
sess = tf.Session(config=config)

```

## 3.1 Data Preparation

```

In [6]: def load_data(folder):
    """
        This function takes data folder path as input and returns a dataframe
        which contains filename, filepath, class.
    """
    X = []
    Z = []
    Y = []

    for folderName in os.listdir(folder):
        if not folderName.startswith('.'):
            if folderName == 'CNV':
                label = 0
            elif folderName == 'DME':
                label = 1
            elif folderName == 'DRUSEN':
                label = 2
            elif folderName == 'NORMAL':
                label = 3
            else:
                label = 4
            for file in os.listdir(folder+'/'+folderName):
                if not file.startswith('.'):
                    X.append(file)
                    Z.append(folder+'/'+folderName+'/'+file)
                    Y.append(label)

    return pd.DataFrame(np.hstack((np.asarray(X).reshape(-1,1),np.asarray(Z).r
eshape(-1,1), \
                                    np.asarray(Y).reshape(-1,1))), columns=['fi
lename', 'path', "Class"]).sort_values("Class")

```

```
In [7]: if not os.path.isfile("data.csv"):
    data = load_data("data")
    data.to_csv("data.csv", index=False)

data = pd.read_csv("data.csv")
data.head()
```

Out[7]:

	filename	path	Class
0	CNV-3715091-62.jpeg	data/CNV/CNV-3715091-62.jpeg	0
1	CNV-2510299-1.jpeg	data/CNV/CNV-2510299-1.jpeg	0
2	CNV-349021-8.jpeg	data/CNV/CNV-349021-8.jpeg	0
3	CNV-2760476-11.jpeg	data/CNV/CNV-2760476-11.jpeg	0
4	CNV-6652117-549.jpeg	data/CNV/CNV-6652117-549.jpeg	0

In [8]: data.shape

Out[8]: (83495, 3)

```
In [9]: class_weights = data["Class"].value_counts()/data.shape[0]
class_weights = dict(class_weights)
```

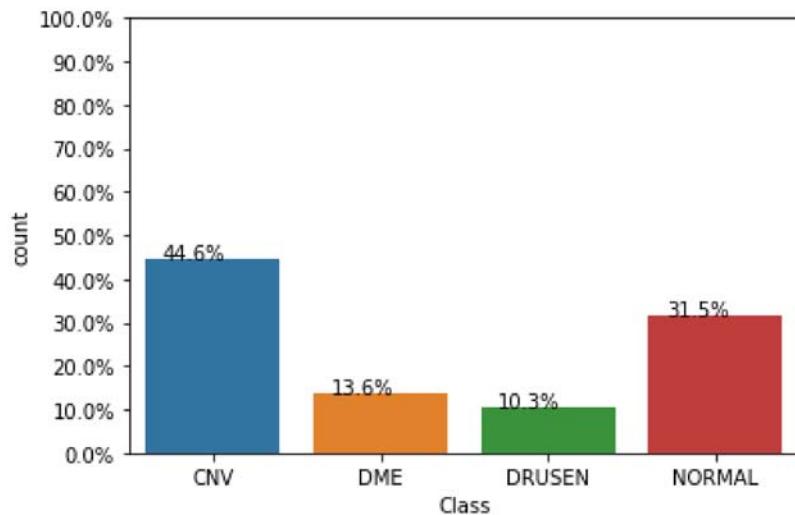
```
In [10]: class_dict = {0:"CNV", 1:"DME", 2:"DRUSEN", 3:"NORMAL"}
class_dict
classLabels = ["CNV", "DME", "DRUSEN", "NORMAL"]
```

## Data Class Distribution

```
In [11]: Y = data.copy()
Y["Class"] = Y["Class"].apply(lambda x: class_dict[x])
total = len(Y)*1
ax=sns.countplot(x="Class", data=Y)
for p in ax.patches:
    ax.annotate('{:.1f}%'.format(100*p.get_height()/total), (p.get_x()+0.1
, p.get_height()+5))

#put 11 ticks (therefore 10 steps), from 0 to the total number of rows in the
#dataframe
ax.yaxis.set_ticks(np.linspace(0, total, 11))

#adjust the ticklabel to the desired format, without changing the position of
#the ticks.
ax.set_yticklabels(map('{:.1f}%'.format, 100*ax.yaxis.get_majorticklocs()/total))
plt.show()
```



```
In [12]: def sample_data(data, sample_size=2000):
    """
        This function take a dataset and select a smaller sample based on sample_size
    """

    if sample_size == data.shape[0]:
        return data

    class_weights = dict(data["Class"].value_counts()/data.shape[0])

    df0 = data[data["Class"]==0].sample(n=min(int(round(sample_size * class_weights[0])), len(data[data["Class"]==0])))
    df1 = data[data["Class"]==1].sample(n=min(int(round(sample_size * class_weights[1])), len(data[data["Class"]==1])))
    df2 = data[data["Class"]==2].sample(n=min(int(round(sample_size * class_weights[2])), len(data[data["Class"]==2])))
    df3 = data[data["Class"]==3].sample(n=min(int(round(sample_size * class_weights[3])), len(data[data["Class"]==3])))

    return pd.concat([df0, df1, df2, df3])
```

Here we are taking 60000 images sample with same class distribution as in whole data.

Train - Validation - Test split ==> 72.5% - 12.75% - 15%

```
In [13]: # Here we are taking 60000 images sample with same class distribution as in whole data
# Train - Validation - Test split ==> 72.5% - 12.75% - 15%

if not (os.path.isfile("X_train.p") and os.path.isfile("X_val.p") and os.path.isfile("X_test.p") and \
        os.path.isfile("y_train.p") and os.path.isfile("y_val.p") and os.path.isfile("y_test.p")):

    data_sampled = sample_data(data, 60000)
    print("Data Size : ", data_sampled.shape[0])

    X_data = data_sampled['path'].values
    Y_data = data_sampled["Class"].values

    # print("Data Size : ", data.shape[0])

    # X_data = data['path'].values
    # Y_data = data["Class"].values

    X_train, X_test, Y_train, Y_test = train_test_split(X_data, Y_data, stratify=Y_data, test_size=0.15, random_state=42)
    X_train, X_val, Y_train, Y_val = train_test_split(X_train, Y_train, stratify=Y_train, test_size=0.15, random_state=42)

    y_train = np_utils.to_categorical(Y_train, 4)
    y_test = np_utils.to_categorical(Y_test, 4)
    y_val = np_utils.to_categorical(Y_val, 4)

    pickle.dump(X_train, open("X_train.p","wb"))
    pickle.dump(X_val, open("X_val.p","wb"))
    pickle.dump(X_test, open("X_test.p","wb"))

    pickle.dump(y_train, open("y_train.p","wb"))
    pickle.dump(y_val, open("y_val.p","wb"))
    pickle.dump(y_test, open("y_test.p","wb"))

X_train = pickle.load(open("X_train.p","rb"))
X_val = pickle.load(open("X_val.p","rb"))
X_test = pickle.load(open("X_test.p","rb"))

y_train = pickle.load(open("y_train.p","rb"))
y_val = pickle.load(open("y_val.p","rb"))
y_test = pickle.load(open("y_test.p","rb"))

print("Train Data Size : ", len(X_train))
print("Val Data Size : ", len(X_val))
print("Test Data Size : ", len(X_test))
```

Train Data Size : 43350  
Val Data Size : 7650  
Test Data Size : 9001

## 4. ML Models

```
In [14]: %matplotlib notebook
import matplotlib.pyplot as plt
import numpy as np
import time
# https://gist.github.com/greydanus/f6eee59eaf1d90fcb3b534a25362cea4
# https://stackoverflow.com/a/14434334
def plt_dynamic(x, vy, ty, ax, plot_label, colors=['b']):
    ''' this function is used to update the plots for each epoch and error'''

    ax.plot(x, vy, 'b', label="Validation " + plot_label)
    ax.plot(x, ty, 'r', label="Train " + plot_label)
    plt.legend()
    plt.grid()
    fig.canvas.draw()

def plot_confusion_matrix(test_y, test_predict_y, data_type):
    ''' This function plots the confusion , precision and recall matrices give
n y_i, y_i_hat. '''

    C = confusion_matrix(test_y, test_predict_y)
    # C = 9,9 matrix, each cell (i,j) represents number of points of class i a
re predicted class j

    A =((C.T)/(C.sum(axis=1))).T
    #divid each element of the confusion matrix with the sum of elements in th
at column

    # C = [[1, 2],
    #      [3, 4]]
    # C.T = [[1, 3],
    #          [2, 4]]
    # C.sum(axis = 1) axis=0 corresonds to columns and axis=1 corresponds to
rows in two diimensional array
    # C.sum(axix =1) = [[3, 7]]
    # ((C.T)/(C.sum(axis=1))) = [[1/3, 3/7]
    #                               [2/3, 4/7]]

    # ((C.T)/(C.sum(axis=1))).T = [[1/3, 2/3]
    #                               [3/7, 4/7]]
    # sum of row elements = 1

    B =(C/C.sum(axis=0))
    #divid each element of the confusion matrix with the sum of elements in th
at row
    # C = [[1, 2],
    #      [3, 4]]
    # C.sum(axis = 0) axis=0 corresonds to columns and axis=1 corresponds to
rows in two diimensional array
    # C.sum(axix =0) = [[4, 6]]
    # (C/C.sum(axis=0)) = [[1/4, 2/6],
    #                      [3/4, 4/6]]
    plt.figure(figsize=(20,5))
    plt.suptitle(data_type + " Confusion, Presicion & Recall Matrix", fontsize
```

```
=24)
```

```
labels = ["CNV", "DME", "DRUSEN", "NORMAL"]
# representing A in heatmap format
plt.subplot(1, 3, 1)
sns.heatmap(C, annot=True, fmt=".3f", xticklabels=labels, yticklabels=labels)
plt.xlabel('Predicted Class')
plt.ylabel('Original Class')
plt.title("Confusion matrix")

plt.subplot(1, 3, 2)
sns.heatmap(B, annot=True, fmt=".3f", xticklabels=labels, yticklabels=labels)
plt.xlabel('Predicted Class')
plt.ylabel('Original Class')
plt.title("Precision matrix")

plt.subplot(1, 3, 3)
# representing B in heatmap format
sns.heatmap(A, annot=True, fmt=".3f", xticklabels=labels, yticklabels=labels)
plt.xlabel('Predicted Class')
plt.ylabel('Original Class')
plt.title("Recall matrix")

plt.show()
```

```
In [15]: def imageToArray(files):
    ''' This function takes List of image files and convert each image into 3d arrays'''
    images = []
    for file in files:
        images.append(np.float32(cv2.resize(cv2.imread(file, 1), (224, 224) )) / 255.0)
    return np.asarray(images)

# Code copied from https://stackoverflow.com/questions/47200146/keras-Load-images-batch-wise-for-large-dataset

def imageLoader(files, ys, batch_size):
    '''this function will help us to load images batch by batch then convert to array'''

    L = len(files)

    #this line is just to make the generator infinite, keras needs that
    while True:

        batch_start = 0
        batch_end = batch_size

        while batch_start < L:
            limit = min(batch_end, L)
            X = imageToArray(files[batch_start:limit])
            Y = ys[batch_start:limit]

            yield (X,Y) #a tuple with two numpy arrays with batch_size samples

            batch_start += batch_size
            batch_end += batch_size
```

```
In [16]: # Copied from https://stackoverflow.com/questions/41032551/how-to-compute-receiving-operating-characteristic-roc-and-auc-in-keras

import tensorflow as tf
import keras
from sklearn.metrics import f1_score

# This function act as custom metric keras models. This function calculates f1-micro score.

def f1(y_true, y_pred):
    return f1_score(np.argmax(y_true, axis=1), np.argmax(y_pred, axis=1), average='micro')

def f1_micro_score(y_true, y_pred):
    return tf.py_function(f1, (y_true, y_pred), tf.double)
```

```
In [17]: model_name = ["model_1.hdf5", "model_2.hdf5", "model_3.hdf5", "model_4.hdf5",
"model_5.hdf5"]
log_dir_name = ["graph_1", "graph_2", "graph_3", "graph_4", "graph_5"]

def train_model(model, batch_size, epochs, model_no):
    """
    This function takes model, batch_size and no. of epochs as input and train the model.
    This function returns model and model history.
    """

    model.compile(optimizer='adam', loss='categorical_crossentropy', metrics=[f1_micro_score])

    train_steps = len(X_train)//batch_size+1
    val_steps = len(X_val)//batch_size+1
    test_steps = len(X_test)//batch_size+1

    checkpoint = ModelCheckpoint(model_name[model_no], monitor='val_loss', verbose=1, save_best_only=True, mode='min')

    early_stopping = EarlyStopping(monitor = 'val_loss', min_delta = 0, patience = 3, verbose = 1, restore_best_weights = True)

    reduce_lr = ReduceLROnPlateau(monitor = 'val_loss', factor = 0.2, patience = 1, verbose = 1, min_delta = 0.00001)

    tensorboard = TensorBoard(log_dir=log_dir_name[model_no], histogram_freq=0, batch_size=batch_size, write_graph=True, write_grads=False, write_images=False, embeddings_freq=0, embeddings_layer_names=None, embeddings_metadata=None, embeddings_data=None, update_freq='epoch')

    callbacks_list = [checkpoint, early_stopping, reduce_lr, tensorboard]

    history = model.fit_generator(imageLoader(X_train, y_train, batch_size), epochs=epochs, steps_per_epoch=train_steps, verbose=1, validation_data=imageLoader(X_val, y_val, batch_size), validation_steps=val_steps, callbacks=callbacks_list)

    return model, history

def report_of_model(model, model_name, batch_size = 64):
    """
    This function prints loss, metric, classification reports and confusion metrics for train, val and test data.
    This takes model, model_name and batch size as input and returns a list containg model_name, loss and metric value.
    """


```

```

"""
train_steps = len(X_train)//batch_size+1
val_steps = len(X_val)//batch_size+1
test_steps = len(X_test)//batch_size+1

train_score = model.evaluate_generator(imageLoader(X_train, y_train, batch_size), steps=train_steps, verbose=1)
val_score = model.evaluate_generator(imageLoader(X_val, y_val, batch_size), steps=val_steps, verbose=1)
test_score = model.evaluate_generator(imageLoader(X_test, y_test, batch_size), steps=test_steps, verbose=1)

print("*"*55)

print("Train log-loss : ", round(train_score[0], 3))
print("Validation log-loss : ", round(val_score[0], 3))
print("Test log-loss : ", round(test_score[0], 3))

print("*"*55)

y_train_pred = model.predict_generator(imageLoader(X_train, y_train, batch_size), steps=train_steps, verbose=1)
y_val_pred = model.predict_generator(imageLoader(X_val, y_val, batch_size), steps=val_steps, verbose=1)
y_test_pred = model.predict_generator(imageLoader(X_test, y_test, batch_size), steps=test_steps, verbose=1)

train_f1_score = f1_score(y_train.argmax(axis=1), y_train_pred.argmax(axis=1), average='micro')
val_f1_score = f1_score(y_val.argmax(axis=1), y_val_pred.argmax(axis=1), average='micro')
test_f1_score = f1_score(y_test.argmax(axis=1), y_test_pred.argmax(axis=1), average='micro')

print("*"*55)

print("Train F1-micro avg score : ", round(train_f1_score,4))
print("Val F1-micro avg score : ", round(val_f1_score,4))
print("Test F1-micro avg score : ", round(test_f1_score,4))

print("*"*55)

class_labels = ['CNV', 'DME', 'DRUSEN', 'NORMAL']

#Confution Matrix and Classification Report

# Classification Report
print('Train Classification Report')
print(classification_report(y_train.argmax(axis=1), y_train_pred.argmax(axis=-1), target_names=class_labels))
# Confusion Matrix
print('Train Confusion Matrix')
plot_confusion_matrix(y_train.argmax(axis=1), y_train_pred.argmax(axis=-1), "Train")

```

```

# Classification Report
print('Validation Classification Report')
print(classification_report(y_val.argmax(axis=1), y_val_pred.argmax(axis=-1), target_names=class_labels))
# Confusion Matrix
print('Validation Confusion Matrix')
plot_confusion_matrix(y_val.argmax(axis=1), y_val_pred.argmax(axis=-1), "Validation")

# Classification Report
print('Test Classification Report')
print(classification_report(y_test.argmax(axis=1), y_test_pred.argmax(axis=-1), target_names=class_labels))
# Confusion Matrix
print('Test Confusion Matrix')
plot_confusion_matrix(y_test.argmax(axis=1), y_test_pred.argmax(axis=-1), "Test")

metric = [model_name] + train_score + val_score + test_score + [train_f1_score] + [val_f1_score] + [test_f1_score]

return metric

```

In [18]: `input_shape = (np.float32(cv2.resize(cv2.imread(data["path"].values[0], 1), (224, 224))) / 255.0).shape  
input_shape`

Out[18]: (224, 224, 3)

In [19]: `metrics = []  
num_classes = 4`

## 4.1. Deep Neural Models

### 1) 3 layer CNN Model

```
In [70]: # Ref : https://github.com/BadreeshShetty/OCT-Retinal/blob/master/OCT-Retinal%20CNN.ipynb

model = Sequential()

model.add(Conv2D(32, kernel_size = (3, 3), activation="relu",padding = 'same',
input_shape = input_shape))
model.add(BatchNormalization())

model.add(Conv2D(64, kernel_size = (3, 3), activation="relu",padding = 'same'))
model.add(BatchNormalization())

model.add(MaxPooling2D((2, 2)))
model.add(Dropout(0.2))

model.add(Conv2D(128, kernel_size = (3, 3), activation="relu",padding = 'same'))
model.add(BatchNormalization())

model.add(GlobalAveragePooling2D())
model.add(Dense(num_classes, activation='softmax'))

print(model.summary())
```

Model: "sequential\_7"

Layer (type)	Output Shape	Param #
<hr/>		
conv2d_31 (Conv2D)	(None, 224, 224, 32)	896
batch_normalization_31 (BatchNormalization)	(None, 224, 224, 32)	128
conv2d_32 (Conv2D)	(None, 224, 224, 64)	18496
batch_normalization_32 (BatchNormalization)	(None, 224, 224, 64)	256
max_pooling2d_13 (MaxPooling2D)	(None, 112, 112, 64)	0
dropout_13 (Dropout)	(None, 112, 112, 64)	0
conv2d_33 (Conv2D)	(None, 112, 112, 128)	73856
batch_normalization_33 (BatchNormalization)	(None, 112, 112, 128)	512
global_average_pooling2d_7 (GlobalAveragePooling2D)	(None, 128)	0
dense_7 (Dense)	(None, 4)	516
<hr/>		
Total params:	94,660	
Trainable params:	94,212	
Non-trainable params:	448	
<hr/>		
None		

```
In [71]: model, history = train_model(model, batch_size = 64, epochs = 20, model_no = 0 )
```

```
Epoch 1/20
678/678 [=====] - 695s 1s/step - loss: 0.7489 - f1_micro_score: 0.7214 - val_loss: 1.2417 - val_f1_micro_score: 0.4534

Epoch 00001: val_loss improved from inf to 1.24168, saving model to model_1.hdf5
Epoch 2/20
678/678 [=====] - 531s 784ms/step - loss: 0.4969 - f1_micro_score: 0.8228 - val_loss: 0.8708 - val_f1_micro_score: 0.7782

Epoch 00002: val_loss improved from 1.24168 to 0.87078, saving model to model_1.hdf5
Epoch 3/20
678/678 [=====] - 531s 783ms/step - loss: 0.3759 - f1_micro_score: 0.8686 - val_loss: 0.8145 - val_f1_micro_score: 0.8202

Epoch 00003: val_loss improved from 0.87078 to 0.81452, saving model to model_1.hdf5
Epoch 4/20
678/678 [=====] - 533s 785ms/step - loss: 0.3139 - f1_micro_score: 0.8894 - val_loss: 0.6899 - val_f1_micro_score: 0.8526

Epoch 00004: val_loss improved from 0.81452 to 0.68991, saving model to model_1.hdf5
Epoch 5/20
678/678 [=====] - 531s 783ms/step - loss: 0.2812 - f1_micro_score: 0.9012 - val_loss: 0.8303 - val_f1_micro_score: 0.7923

Epoch 00005: val_loss did not improve from 0.68991

Epoch 00005: ReduceLROnPlateau reducing learning rate to 0.00020000000949949026.
Epoch 6/20
678/678 [=====] - 531s 784ms/step - loss: 0.2404 - f1_micro_score: 0.9169 - val_loss: 0.4931 - val_f1_micro_score: 0.9160

Epoch 00006: val_loss improved from 0.68991 to 0.49315, saving model to model_1.hdf5
Epoch 7/20
678/678 [=====] - 532s 784ms/step - loss: 0.2312 - f1_micro_score: 0.9190 - val_loss: 0.5444 - val_f1_micro_score: 0.9097

Epoch 00007: val_loss did not improve from 0.49315

Epoch 00007: ReduceLROnPlateau reducing learning rate to 4.0000001899898055e-05.
Epoch 8/20
678/678 [=====] - 531s 783ms/step - loss: 0.2211 - f1_micro_score: 0.9240 - val_loss: 0.4370 - val_f1_micro_score: 0.9200

Epoch 00008: val_loss improved from 0.49315 to 0.43702, saving model to model_1.hdf5
Epoch 9/20
678/678 [=====] - 532s 784ms/step - loss: 0.2197 - f1_micro_score: 0.9240 - val_loss: 0.4276 - val_f1_micro_score: 0.9194

Epoch 00009: val_loss improved from 0.43702 to 0.42760, saving model to model_1.hdf5
```

```
_1.hdf5
Epoch 10/20
678/678 [=====] - 532s 785ms/step - loss: 0.2172 - f
1_micro_score: 0.9250 - val_loss: 0.4370 - val_f1_micro_score: 0.9211

Epoch 00010: val_loss did not improve from 0.42760

Epoch 00010: ReduceLROnPlateau reducing learning rate to 8.00000525498762e-0
6.

Epoch 11/20
678/678 [=====] - 530s 782ms/step - loss: 0.2152 - f
1_micro_score: 0.9259 - val_loss: 0.4399 - val_f1_micro_score: 0.9207

Epoch 00011: val_loss did not improve from 0.42760

Epoch 00011: ReduceLROnPlateau reducing learning rate to 1.6000001778593287e-
06.

Epoch 12/20
678/678 [=====] - 530s 781ms/step - loss: 0.2150 - f
1_micro_score: 0.9243 - val_loss: 0.4370 - val_f1_micro_score: 0.9204

Epoch 00012: val_loss did not improve from 0.42760
Restoring model weights from the end of the best epoch

Epoch 00012: ReduceLROnPlateau reducing learning rate to 3.200000264769187e-0
7.

Epoch 00012: early stopping
```

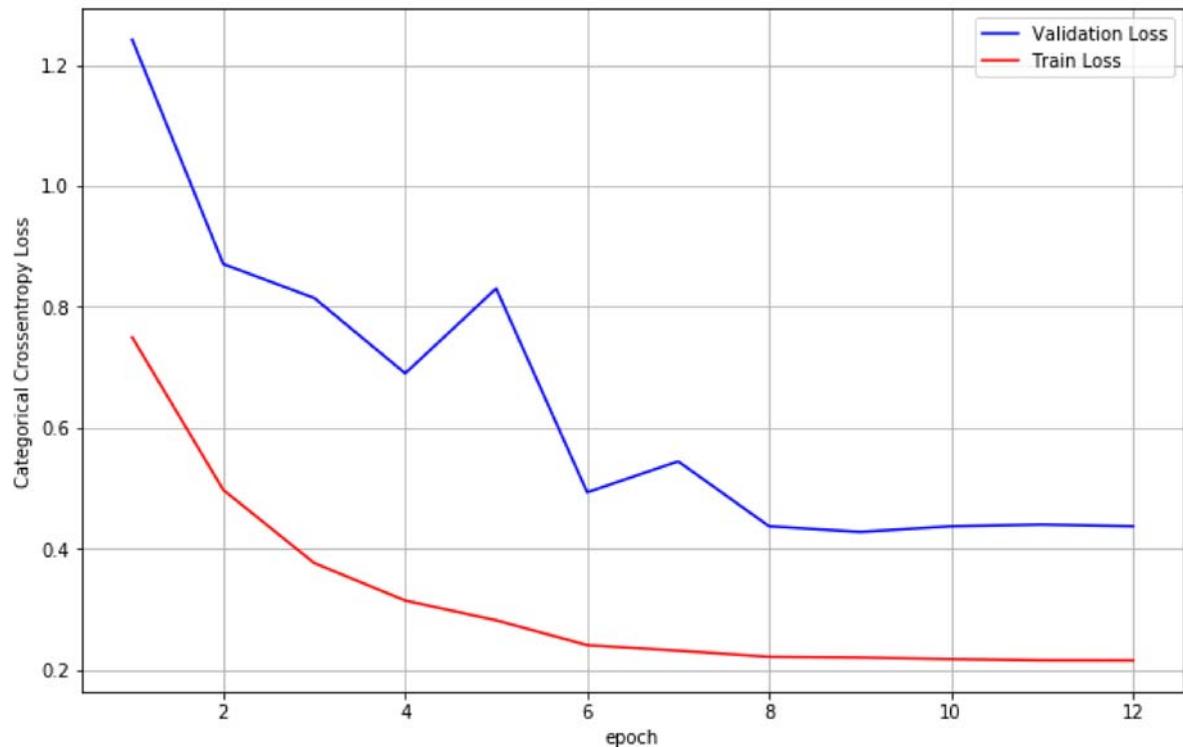
In [72]: `model = load_model("model_1.hdf5", custom_objects={"f1_micro_score": f1_micro_score})`

```
In [76]: nb_epoch = len(history.history['loss'])
%matplotlib inline
import matplotlib.pyplot as plt
plt.rcParams["figure.figsize"] = [11, 7]

fig,ax = plt.subplots(1,1)
ax.set_xlabel('epoch') ; ax.set_ylabel('Categorical Crossentropy Loss')

# List of epoch numbers
x = list(range(1,nb_epoch+1))

vy = history.history['val_loss']
ty = history.history['loss']
plt_dynamic(x, vy, ty, ax, "Loss")
```

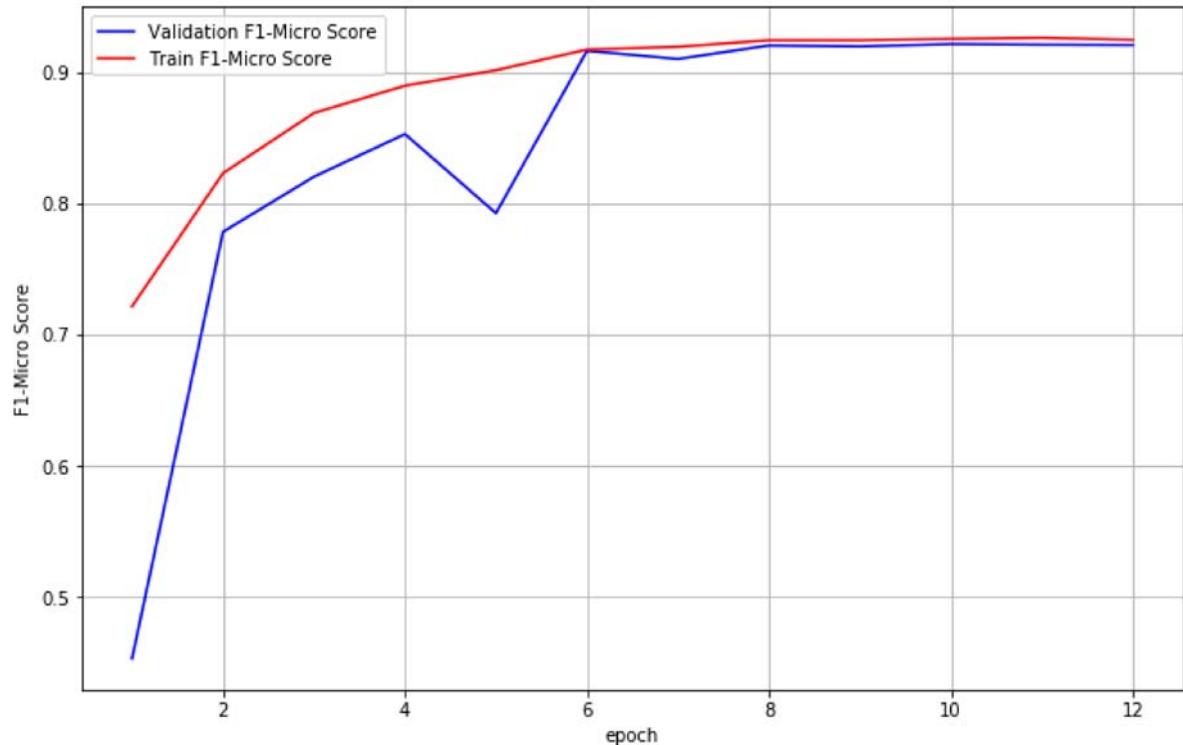


```
In [77]: nb_epoch = len(history.history['f1_micro_score'])
%matplotlib inline
import matplotlib.pyplot as plt
plt.rcParams["figure.figsize"] = [11, 7]

fig,ax = plt.subplots(1,1)
ax.set_xlabel('epoch') ; ax.set_ylabel('F1-Micro Score')

# List of epoch numbers
x = list(range(1,nb_epoch+1))

vy = history.history['val_f1_micro_score']
ty = history.history['f1_micro_score']
plt_dynamic(x, vy, ty, ax, "F1-Micro Score")
```



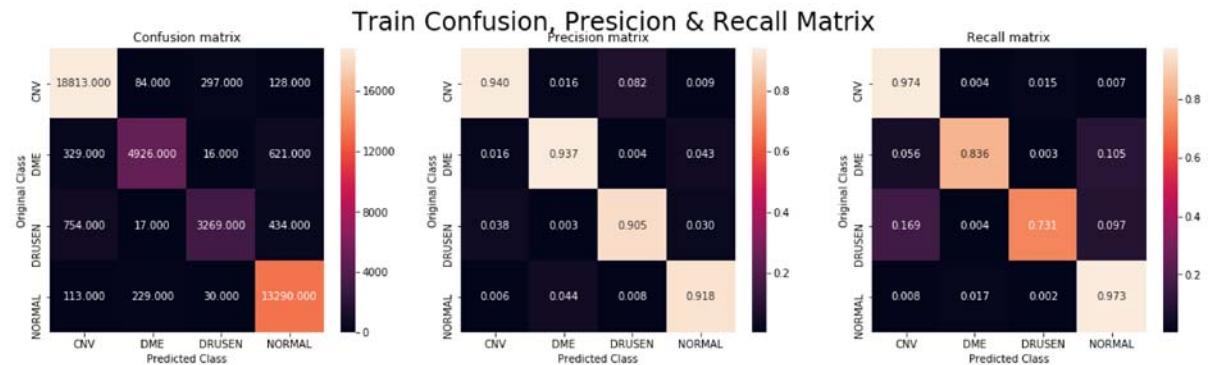
```
In [78]: metric = report_of_model(model, model_name = "3 Layer CNN Model", batch_size = 64)
```

```

678/678 [=====] - 221s 325ms/step
120/120 [=====] - 40s 332ms/step
141/141 [=====] - 107s 756ms/step
=====
Train log-loss : 0.093
Validation log-loss : 0.428
Test log-loss : 0.202
=====
678/678 [=====] - 219s 323ms/step
120/120 [=====] - 40s 329ms/step
141/141 [=====] - 45s 321ms/step
=====
Train F1-micro avg score : 0.9296
Val F1-micro avg score : 0.9199
Test F1-micro avg score : 0.9203
=====
Train Classification Report
      precision    recall   f1-score   support
CNV        0.94     0.97     0.96    19322
DME        0.94     0.84     0.88     5892
DRUSEN     0.91     0.73     0.81     4474
NORMAL     0.92     0.97     0.94    13662
accuracy          0.93     0.93     0.93    43350
macro avg       0.93     0.88     0.90    43350
weighted avg    0.93     0.93     0.93    43350

```

#### Train Confusion Matrix



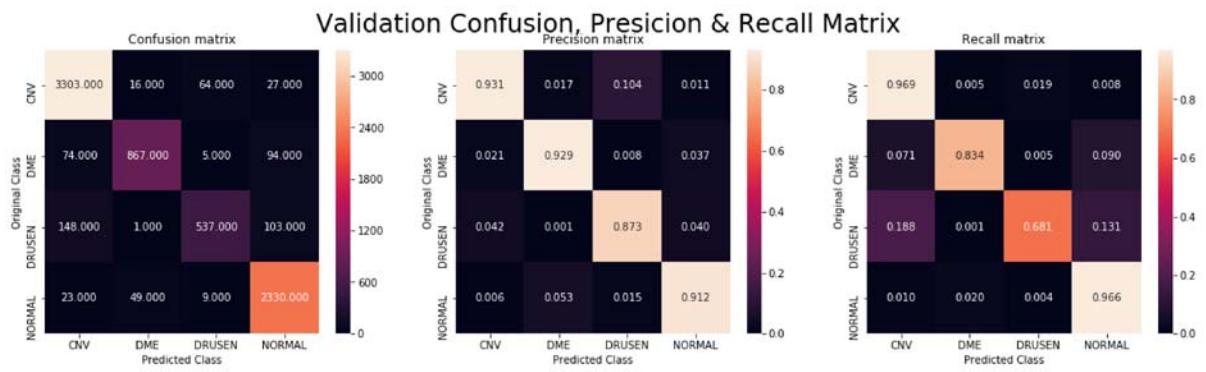
#### Validation Classification Report

```

      precision    recall   f1-score   support
CNV        0.93     0.97     0.95    3410
DME        0.93     0.83     0.88    1040
DRUSEN     0.87     0.68     0.76     789
NORMAL     0.91     0.97     0.94    2411
accuracy          0.92     0.92     0.92    7650
macro avg       0.91     0.86     0.88    7650
weighted avg    0.92     0.92     0.92    7650

```

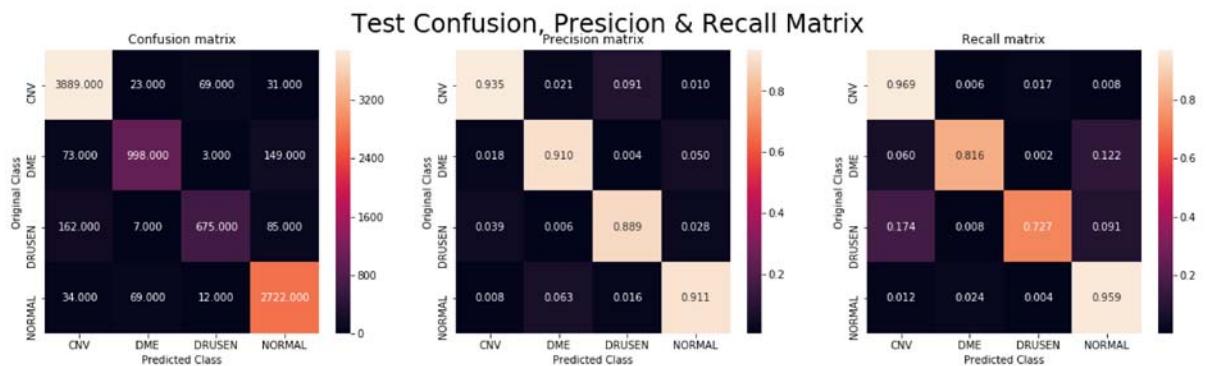
#### Validation Confusion Matrix



### Test Classification Report

	precision	recall	f1-score	support
CNV	0.94	0.97	0.95	4012
DME	0.91	0.82	0.86	1223
DRUSEN	0.89	0.73	0.80	929
NORMAL	0.91	0.96	0.93	2837
accuracy			0.92	9001
macro avg	0.91	0.87	0.89	9001
weighted avg	0.92	0.92	0.92	9001

### Test Confusion Matrix



```
In [80]: metrics.append(metric)
```

```
In [82]: pickle.dump(metrics, open("metrics.p","wb"))
```

```
In [20]: metrics = pickle.load(open("metrics.p","rb"))
```

## 2) 7 layer CNN Model

```
In [84]: # Ref : https://github.com/BadreeshShetty/OCT-Retinal/blob/master/OCT-Retinal%20CNN.ipynb

model = Sequential()

model.add(Conv2D(32, kernel_size = (3, 3), activation="relu",padding = 'same',
input_shape = input_shape))
model.add(BatchNormalization())

model.add(Conv2D(32, kernel_size = (3, 3), activation="relu",padding = 'same'))
model.add(BatchNormalization())

model.add(MaxPooling2D((2, 2)))
model.add(Dropout(0.2))

model.add(Conv2D(128, kernel_size = (3, 3), activation="relu",padding = 'same'))
model.add(BatchNormalization())

model.add(Conv2D(128, kernel_size = (3, 3), activation="relu",padding = 'same'))
model.add(BatchNormalization())

model.add(MaxPooling2D((2, 2)))
model.add(Dropout(0.2))

model.add(Conv2D(256, kernel_size = (3, 3), activation="relu",padding = 'same'))
model.add(BatchNormalization())

model.add(Conv2D(256, kernel_size = (3, 3), activation="relu",padding = 'same'))
model.add(BatchNormalization())

model.add(MaxPooling2D((2, 2)))
model.add(Dropout(0.2))

model.add(Conv2D(512, kernel_size = (3, 3), activation="relu",padding = 'same'))
model.add(BatchNormalization())

model.add(GlobalAveragePooling2D())
model.add(Dense(num_classes, activation='softmax'))

print(model.summary())
```

Model: "sequential\_8"

Layer (type)	Output Shape	Param #
<hr/>		
conv2d_34 (Conv2D)	(None, 224, 224, 32)	896
batch_normalization_34 (Batch Normalization)	(None, 224, 224, 32)	128
conv2d_35 (Conv2D)	(None, 224, 224, 32)	9248
batch_normalization_35 (Batch Normalization)	(None, 224, 224, 32)	128
max_pooling2d_14 (MaxPooling)	(None, 112, 112, 32)	0
dropout_14 (Dropout)	(None, 112, 112, 32)	0
conv2d_36 (Conv2D)	(None, 112, 112, 128)	36992
batch_normalization_36 (Batch Normalization)	(None, 112, 112, 128)	512
conv2d_37 (Conv2D)	(None, 112, 112, 128)	147584
batch_normalization_37 (Batch Normalization)	(None, 112, 112, 128)	512
max_pooling2d_15 (MaxPooling)	(None, 56, 56, 128)	0
dropout_15 (Dropout)	(None, 56, 56, 128)	0
conv2d_38 (Conv2D)	(None, 56, 56, 256)	295168
batch_normalization_38 (Batch Normalization)	(None, 56, 56, 256)	1024
conv2d_39 (Conv2D)	(None, 56, 56, 256)	590080
batch_normalization_39 (Batch Normalization)	(None, 56, 56, 256)	1024
max_pooling2d_16 (MaxPooling)	(None, 28, 28, 256)	0
dropout_16 (Dropout)	(None, 28, 28, 256)	0
conv2d_40 (Conv2D)	(None, 28, 28, 512)	1180160
batch_normalization_40 (Batch Normalization)	(None, 28, 28, 512)	2048
global_average_pooling2d_8 (Global Average Pooling)	(None, 512)	0
dense_8 (Dense)	(None, 4)	2052
<hr/>		
Total params:	2,267,556	
Trainable params:	2,264,868	
Non-trainable params:	2,688	
<hr/>		
None		

```
In [85]: model, history = train_model(model, batch_size = 64, epochs = 20, model_no = 1 )
```

Epoch 1/20  
678/678 [=====] - 872s 1s/step - loss: 0.4637 - f1\_m  
icro\_score: 0.8353 - val\_loss: 1.8722 - val\_f1\_micro\_score: 0.6366

Epoch 00001: val\_loss improved from inf to 1.87217, saving model to model\_2.h  
df5

Epoch 2/20  
678/678 [=====] - 872s 1s/step - loss: 0.2049 - f1\_m  
icro\_score: 0.9305 - val\_loss: 1.1848 - val\_f1\_micro\_score: 0.8151

Epoch 00002: val\_loss improved from 1.87217 to 1.18477, saving model to model  
\_2.hdf5

Epoch 3/20  
678/678 [=====] - 862s 1s/step - loss: 0.1670 - f1\_m  
icro\_score: 0.9428 - val\_loss: 1.1990 - val\_f1\_micro\_score: 0.7727

Epoch 00003: val\_loss did not improve from 1.18477

Epoch 00003: ReduceLROnPlateau reducing learning rate to 0.00020000009499490  
26.

Epoch 4/20  
678/678 [=====] - 860s 1s/step - loss: 0.1206 - f1\_m  
icro\_score: 0.9589 - val\_loss: 0.1873 - val\_f1\_micro\_score: 0.9540

Epoch 00004: val\_loss improved from 1.18477 to 0.18734, saving model to model  
\_2.hdf5

Epoch 5/20  
678/678 [=====] - 870s 1s/step - loss: 0.1076 - f1\_m  
icro\_score: 0.9634 - val\_loss: 0.2471 - val\_f1\_micro\_score: 0.9510

Epoch 00005: val\_loss did not improve from 0.18734

Epoch 00005: ReduceLROnPlateau reducing learning rate to 4.0000001899898055e-  
05.

Epoch 6/20  
678/678 [=====] - 870s 1s/step - loss: 0.0904 - f1\_m  
icro\_score: 0.9692 - val\_loss: 0.2240 - val\_f1\_micro\_score: 0.9570

Epoch 00006: val\_loss did not improve from 0.18734

Epoch 00006: ReduceLROnPlateau reducing learning rate to 8.000000525498762e-0  
6.

Epoch 7/20  
678/678 [=====] - 861s 1s/step - loss: 0.0837 - f1\_m  
icro\_score: 0.9712 - val\_loss: 0.2360 - val\_f1\_micro\_score: 0.9568

Epoch 00007: val\_loss did not improve from 0.18734  
Restoring model weights from the end of the best epoch

Epoch 00007: ReduceLROnPlateau reducing learning rate to 1.6000001778593287e-  
06.

Epoch 00007: early stopping

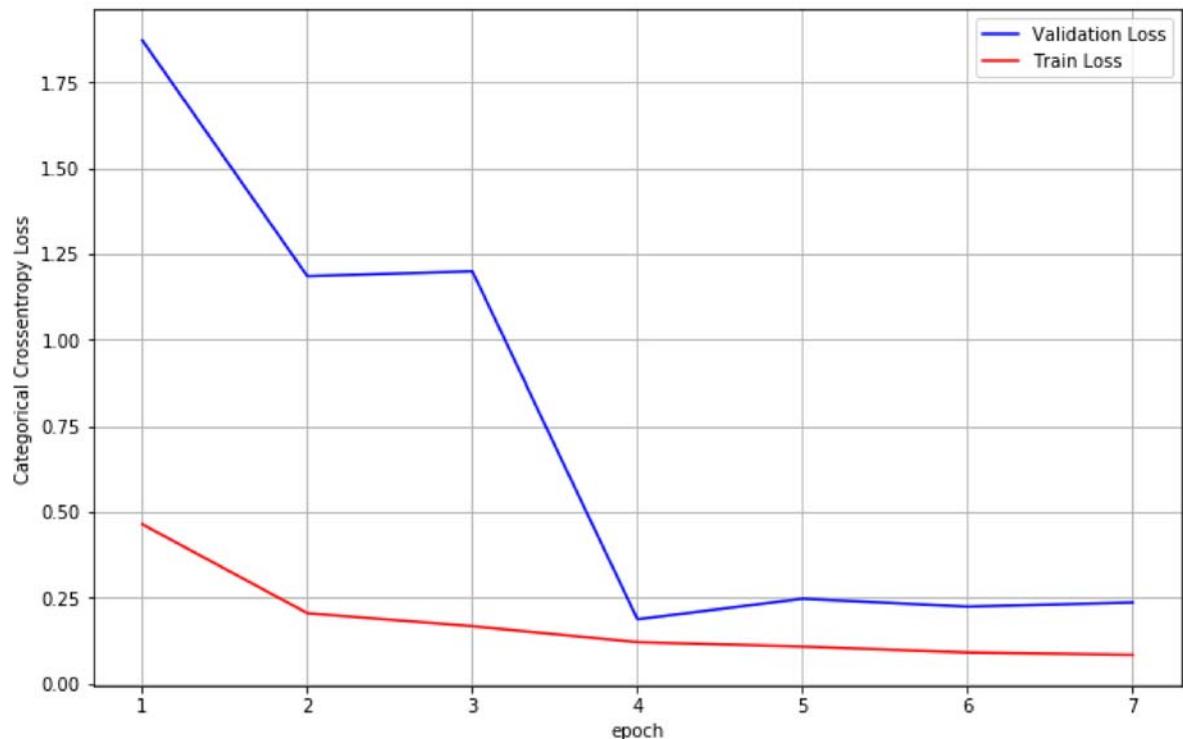
```
In [86]: model = load_model("model_2.hdf5", custom_objects={"f1_micro_score": f1_micro_score})
```

```
In [87]: nb_epoch = len(history.history['loss'])
%matplotlib inline
import matplotlib.pyplot as plt
plt.rcParams["figure.figsize"] = [11, 7]

fig,ax = plt.subplots(1,1)
ax.set_xlabel('epoch') ; ax.set_ylabel('Categorical Crossentropy Loss')

# List of epoch numbers
x = list(range(1,nb_epoch+1))

vy = history.history['val_loss']
ty = history.history['loss']
plt_dynamic(x, vy, ty, ax, "Loss")
```

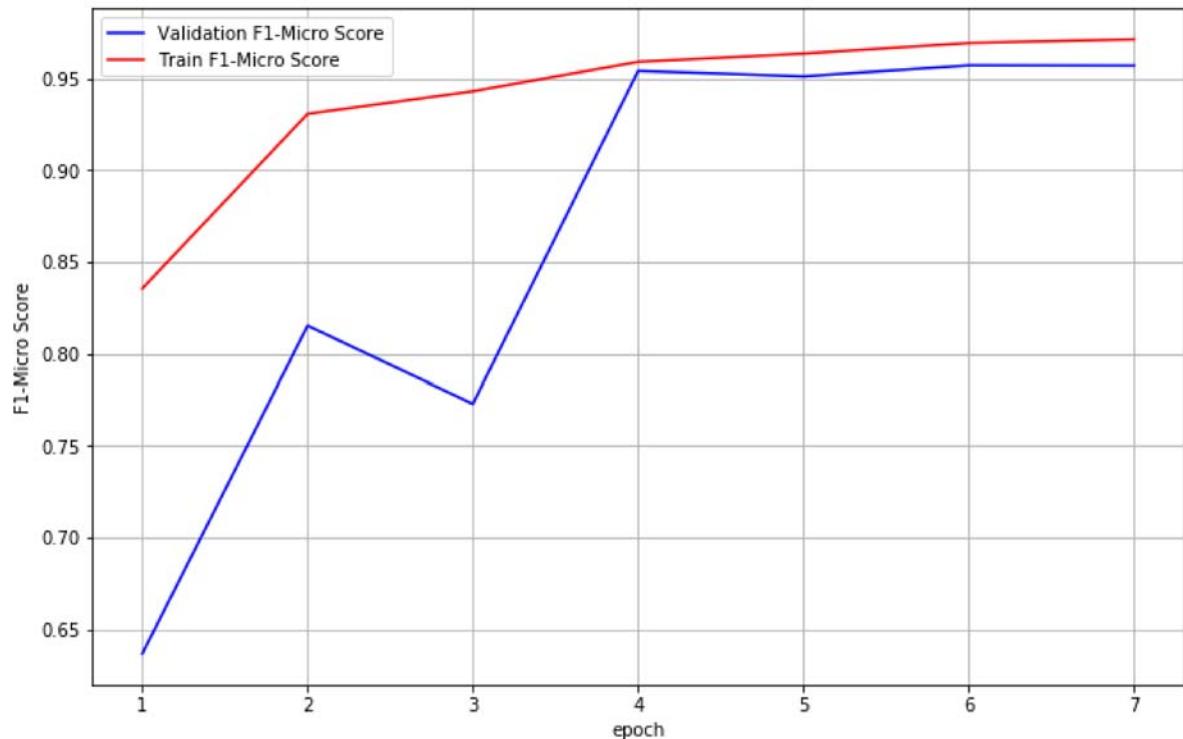


```
In [88]: nb_epoch = len(history.history['f1_micro_score'])
%matplotlib inline
import matplotlib.pyplot as plt
plt.rcParams["figure.figsize"] = [11, 7]

fig,ax = plt.subplots(1,1)
ax.set_xlabel('epoch') ; ax.set_ylabel('F1-Micro Score')

# List of epoch numbers
x = list(range(1,nb_epoch+1))

vy = history.history['val_f1_micro_score']
ty = history.history['f1_micro_score']
plt_dynamic(x, vy, ty, ax, "F1-Micro Score")
```



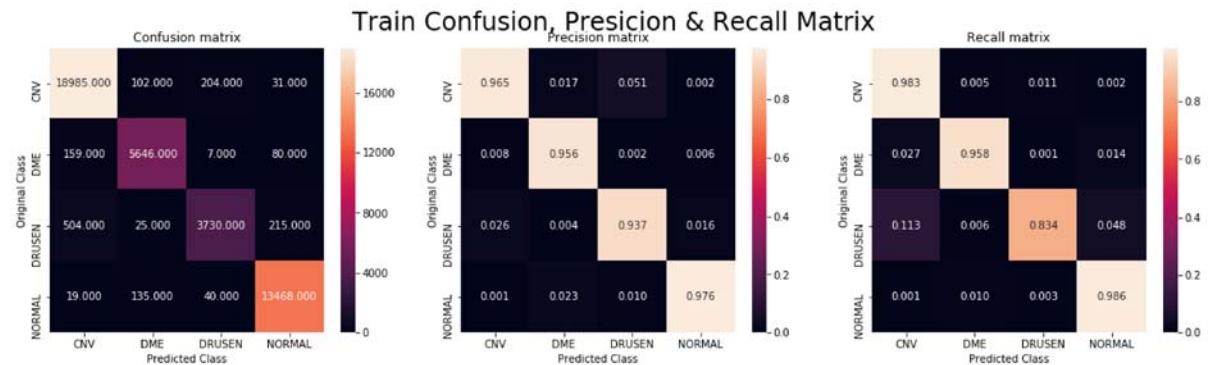
```
In [89]: metric = report_of_model(model, model_name = "7 Layer CNN Model", batch_size = 64)
```

```

678/678 [=====] - 309s 455ms/step
120/120 [=====] - 55s 454ms/step
141/141 [=====] - 66s 471ms/step
=====
Train log-loss : 0.042
Validation log-loss : 0.187
Test log-loss : 0.233
=====
678/678 [=====] - 313s 462ms/step
120/120 [=====] - 56s 464ms/step
141/141 [=====] - 65s 460ms/step
=====
Train F1-micro avg score : 0.9649
Val F1-micro avg score : 0.9542
Test F1-micro avg score : 0.9582
=====
Train Classification Report
      precision    recall   f1-score   support
CNV        0.97     0.98     0.97    19322
DME        0.96     0.96     0.96     5892
DRUSEN     0.94     0.83     0.88     4474
NORMAL     0.98     0.99     0.98    13662
accuracy          0.96     0.96     0.96    43350
macro avg       0.96     0.94     0.95    43350
weighted avg    0.96     0.96     0.96    43350

```

#### Train Confusion Matrix



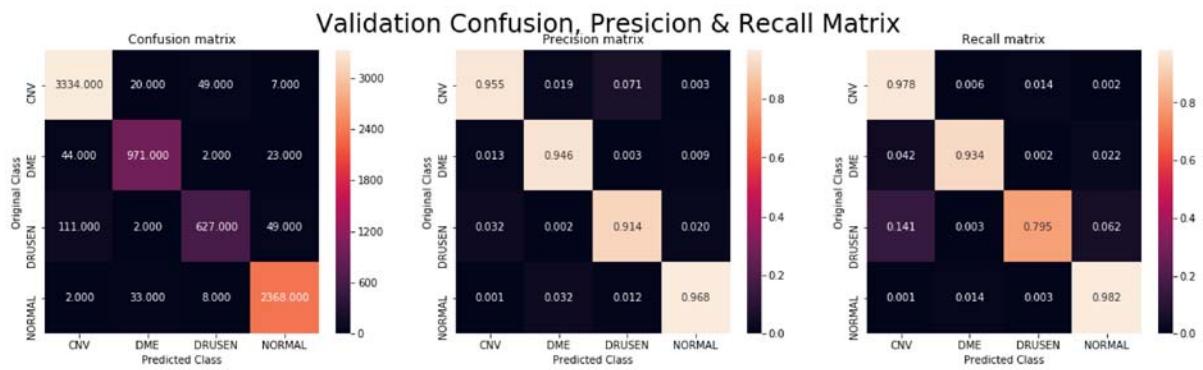
#### Validation Classification Report

```

      precision    recall   f1-score   support
CNV        0.96     0.98     0.97    3410
DME        0.95     0.93     0.94    1040
DRUSEN     0.91     0.79     0.85     789
NORMAL     0.97     0.98     0.97    2411
accuracy          0.95     0.92     0.93    7650
macro avg       0.95     0.95     0.95    7650
weighted avg    0.95     0.95     0.95    7650

```

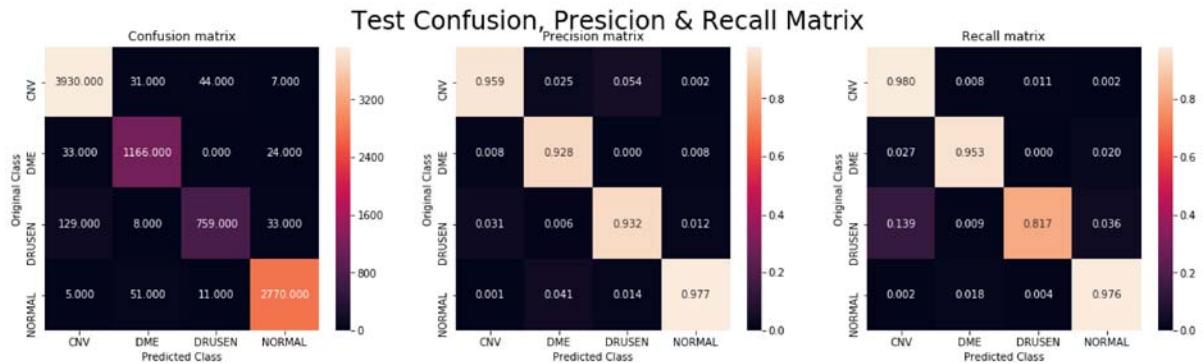
#### Validation Confusion Matrix



### Test Classification Report

	precision	recall	f1-score	support
CNV	0.96	0.98	0.97	4012
DME	0.93	0.95	0.94	1223
DRUSEN	0.93	0.82	0.87	929
NORMAL	0.98	0.98	0.98	2837
accuracy			0.96	9001
macro avg	0.95	0.93	0.94	9001
weighted avg	0.96	0.96	0.96	9001

### Test Confusion Matrix



```
In [91]: metrics.append(metric)
```

```
In [106]: pickle.dump(metrics, open("metrics.p", "wb"))
```

```
In [20]: metrics = pickle.load(open("metrics.p", "rb"))
```

## 3) VGG16 Transfer Learning

```
In [22]: # Ref: https://github.com/BadreeshShetty/OCT-Retinal/blob/master/OCT-Retinal-VGG.ipynb

base_model = VGG16(weights='imagenet', include_top=False, pooling=None, input_shape=(224,224,3))

x = base_model.output
x = Flatten(name = "flatten")(x)
x = Dense(384, activation = "relu")(x)
x = BatchNormalization()(x)
x = Dropout(0.5)(x)
x = Dense(256, activation = "relu")(x)
x = Dropout(0.5)(x)
x = Dense(32, activation = "relu")(x)

output = Dense(4, activation='softmax')(x)
model = Model(inputs=base_model.input, outputs=output)
# Train top layer
for layer in base_model.layers:
    layer.trainable = False

print(model.summary())
```

WARNING:tensorflow:From /usr/local/lib/python3.5/dist-packages/tensorflow\_core/python/ops/resource\_variable\_ops.py:1630: calling BaseResourceVariable.\_init\_\_ (from tensorflow.python.ops.resource\_variable\_ops) with constraint is deprecated and will be removed in a future version.

Instructions for updating:

If using Keras pass \*\_constraint arguments to layers.

WARNING:tensorflow:From /usr/local/lib/python3.5/dist-packages/keras/backend/tensorflow\_backend.py:4070: The name tf.nn.max\_pool is deprecated. Please use tf.nn.max\_pool2d instead.

Model: "model\_1"

Layer (type)	Output Shape	Param #
<hr/>		
input_1 (InputLayer)	(None, 224, 224, 3)	0
block1_conv1 (Conv2D)	(None, 224, 224, 64)	1792
block1_conv2 (Conv2D)	(None, 224, 224, 64)	36928
block1_pool (MaxPooling2D)	(None, 112, 112, 64)	0
block2_conv1 (Conv2D)	(None, 112, 112, 128)	73856
block2_conv2 (Conv2D)	(None, 112, 112, 128)	147584
block2_pool (MaxPooling2D)	(None, 56, 56, 128)	0
block3_conv1 (Conv2D)	(None, 56, 56, 256)	295168
block3_conv2 (Conv2D)	(None, 56, 56, 256)	590080
block3_conv3 (Conv2D)	(None, 56, 56, 256)	590080
block3_pool (MaxPooling2D)	(None, 28, 28, 256)	0
block4_conv1 (Conv2D)	(None, 28, 28, 512)	1180160
block4_conv2 (Conv2D)	(None, 28, 28, 512)	2359808
block4_conv3 (Conv2D)	(None, 28, 28, 512)	2359808
block4_pool (MaxPooling2D)	(None, 14, 14, 512)	0
block5_conv1 (Conv2D)	(None, 14, 14, 512)	2359808
block5_conv2 (Conv2D)	(None, 14, 14, 512)	2359808
block5_conv3 (Conv2D)	(None, 14, 14, 512)	2359808
block5_pool (MaxPooling2D)	(None, 7, 7, 512)	0
flatten (Flatten)	(None, 25088)	0
dense_1 (Dense)	(None, 384)	9634176
batch_normalization_1 (Batch Normalization)	(None, 384)	1536

dropout_1 (Dropout)	(None, 384)	0
dense_2 (Dense)	(None, 256)	98560
dropout_2 (Dropout)	(None, 256)	0
dense_3 (Dense)	(None, 32)	8224
dense_4 (Dense)	(None, 4)	132
<hr/>		
Total params: 24,457,316		
Trainable params: 9,741,860		
Non-trainable params: 14,715,456		
<hr/>		
None		

```
In [23]: model, history = train_model(model, batch_size = 64, epochs = 20, model_no = 2  
)
```

WARNING:tensorflow:From /usr/local/lib/python3.5/dist-packages/keras/backend/tensorflow\_backend.py:422: The name tf.global\_variables is deprecated. Please use tf.compat.v1.global\_variables instead.

WARNING:tensorflow:From /usr/local/lib/python3.5/dist-packages/keras/callbacks/tensorboard\_v1.py:200: The name tf.summary.merge\_all is deprecated. Please use tf.compat.v1.summary.merge\_all instead.

WARNING:tensorflow:From /usr/local/lib/python3.5/dist-packages/keras/callbacks/tensorboard\_v1.py:203: The name tf.summary.FileWriter is deprecated. Please use tf.compat.v1.summary.FileWriter instead.

Epoch 1/20

678/678 [=====] - 1589s 2s/step - loss: 0.5031 - f1\_micro\_score: 0.8153 - val\_loss: 0.4571 - val\_f1\_micro\_score: 0.8518

Epoch 00001: val\_loss improved from inf to 0.45706, saving model to model\_3.hdf5

WARNING:tensorflow:From /usr/local/lib/python3.5/dist-packages/keras/callbacks/tensorboard\_v1.py:343: The name tf.Summary is deprecated. Please use tf.compat.v1.Summary instead.

Epoch 2/20

678/678 [=====] - 465s 685ms/step - loss: 0.3473 - f1\_micro\_score: 0.8760 - val\_loss: 0.3651 - val\_f1\_micro\_score: 0.8927

Epoch 00002: val\_loss improved from 0.45706 to 0.36509, saving model to model\_3.hdf5

Epoch 3/20

678/678 [=====] - 465s 685ms/step - loss: 0.3040 - f1\_micro\_score: 0.8909 - val\_loss: 0.3125 - val\_f1\_micro\_score: 0.8992

Epoch 00003: val\_loss improved from 0.36509 to 0.31249, saving model to model\_3.hdf5

Epoch 4/20

678/678 [=====] - 462s 682ms/step - loss: 0.2739 - f1\_micro\_score: 0.9031 - val\_loss: 0.2959 - val\_f1\_micro\_score: 0.9070

Epoch 00004: val\_loss improved from 0.31249 to 0.29593, saving model to model\_3.hdf5

Epoch 5/20

678/678 [=====] - 463s 683ms/step - loss: 0.2545 - f1\_micro\_score: 0.9105 - val\_loss: 0.3521 - val\_f1\_micro\_score: 0.8997

Epoch 00005: val\_loss did not improve from 0.29593

Epoch 00005: ReduceLROnPlateau reducing learning rate to 0.0002000000949949026.

Epoch 6/20

678/678 [=====] - 464s 685ms/step - loss: 0.2014 - f1\_micro\_score: 0.9291 - val\_loss: 0.3327 - val\_f1\_micro\_score: 0.9193

Epoch 00006: val\_loss did not improve from 0.29593

Epoch 00006: ReduceLROnPlateau reducing learning rate to 4.0000001899898055e-05.

Epoch 7/20

```
678/678 [=====] - 464s 684ms/step - loss: 0.1808 - f
1_micro_score: 0.9365 - val_loss: 0.3311 - val_f1_micro_score: 0.9220
```

Epoch 00007: val\_loss did not improve from 0.29593  
Restoring model weights from the end of the best epoch

Epoch 00007: ReduceLROnPlateau reducing learning rate to 8.000000525498762e-06.

Epoch 00007: early stopping

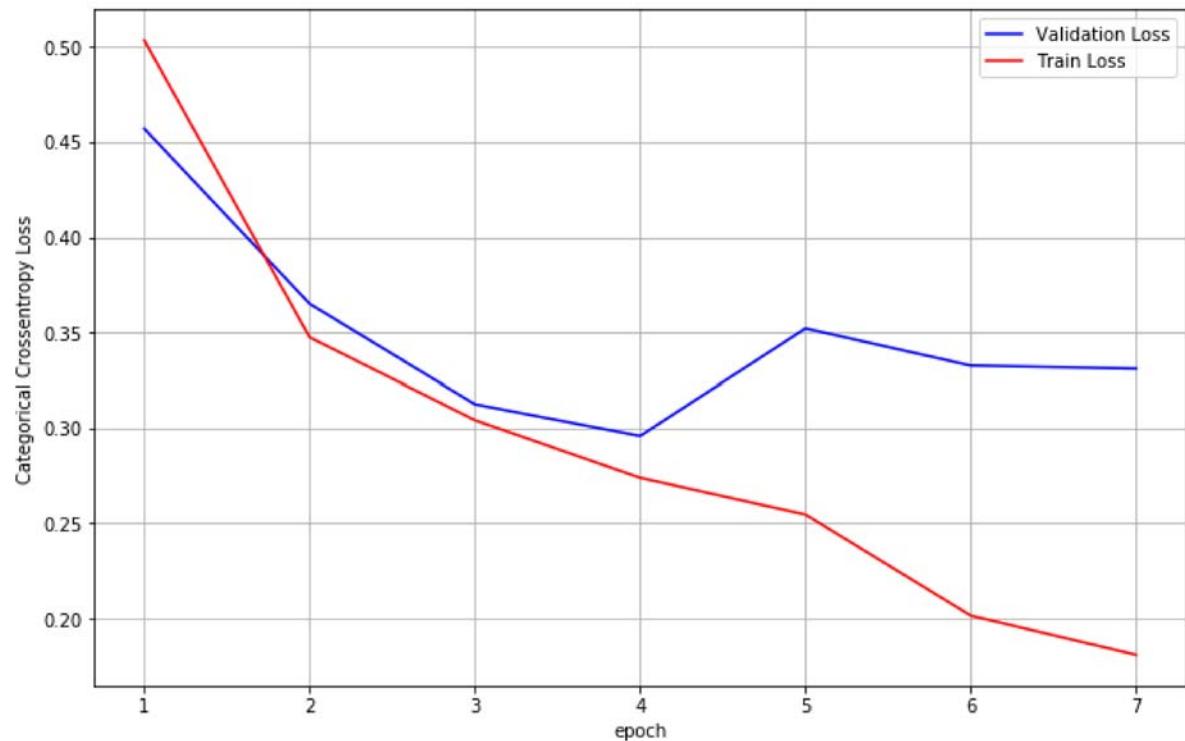
```
In [35]: model = load_model("model_3.hdf5", custom_objects={"f1_micro_score": f1_micro_score})
```

```
In [24]: nb_epoch = len(history.history['loss'])
%matplotlib inline
import matplotlib.pyplot as plt
plt.rcParams["figure.figsize"] = [11, 7]

fig,ax = plt.subplots(1,1)
ax.set_xlabel('epoch') ; ax.set_ylabel('Categorical Crossentropy Loss')

# List of epoch numbers
x = list(range(1,nb_epoch+1))

vy = history.history['val_loss']
ty = history.history['loss']
plt_dynamic(x, vy, ty, ax, "Loss")
```

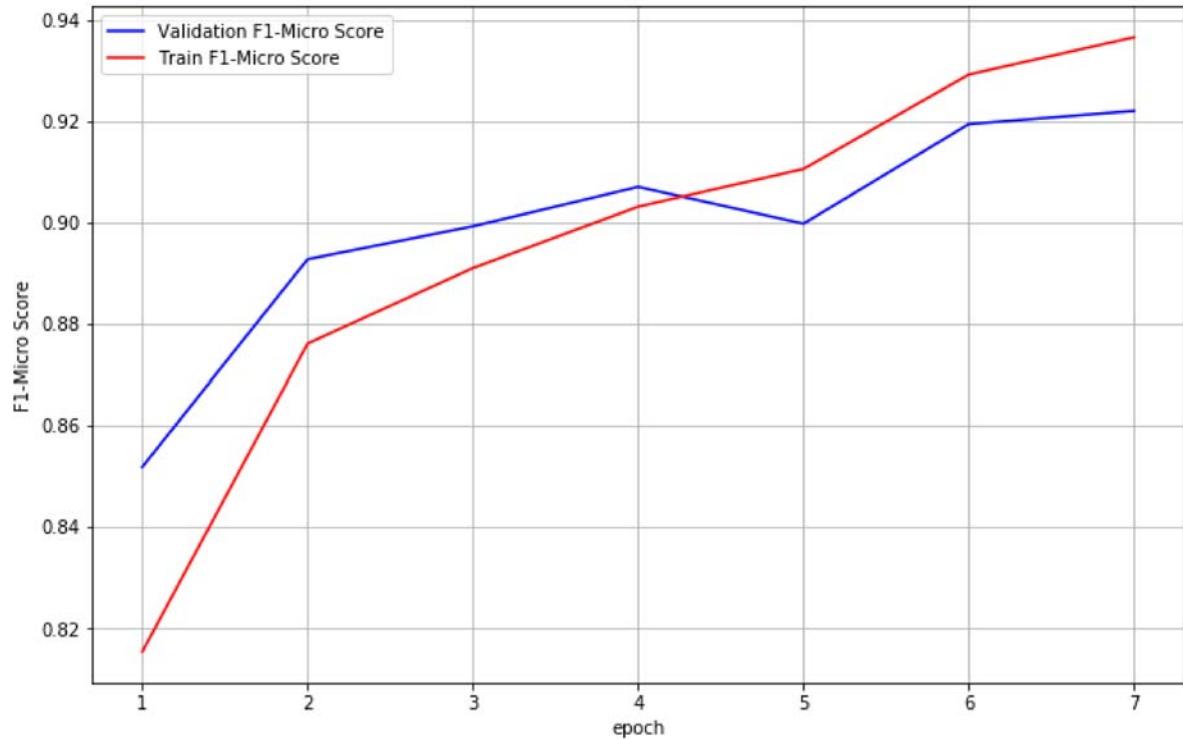


```
In [25]: nb_epoch = len(history.history['f1_micro_score'])
%matplotlib inline
import matplotlib.pyplot as plt
plt.rcParams["figure.figsize"] = [11, 7]

fig,ax = plt.subplots(1,1)
ax.set_xlabel('epoch') ; ax.set_ylabel('F1-Micro Score')

# List of epoch numbers
x = list(range(1,nb_epoch+1))

vy = history.history['val_f1_micro_score']
ty = history.history['f1_micro_score']
plt_dynamic(x, vy, ty, ax, "F1-Micro Score")
```



```
In [26]: metric = report_of_model(model, model_name = "VGG16 Transfer Learning", batch_size = 64)
```

```

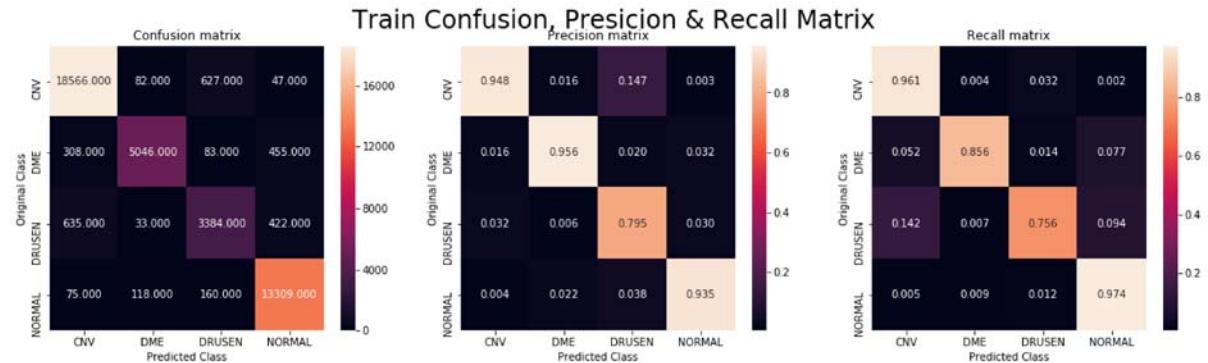
678/678 [=====] - 388s 573ms/step
120/120 [=====] - 69s 572ms/step
141/141 [=====] - 282s 2s/step
=====
Train log-loss : 0.063
Validation log-loss : 0.296
Test log-loss : 0.312
=====
678/678 [=====] - 386s 570ms/step
120/120 [=====] - 68s 570ms/step
141/141 [=====] - 80s 570ms/step
=====
Train F1-micro avg score : 0.9298
Val F1-micro avg score : 0.9071
Test F1-micro avg score : 0.9089
=====

```

#### Train Classification Report

	precision	recall	f1-score	support
CNV	0.95	0.96	0.95	19322
DME	0.96	0.86	0.90	5892
DRUSEN	0.80	0.76	0.78	4474
NORMAL	0.94	0.97	0.95	13662
accuracy			0.93	43350
macro avg	0.91	0.89	0.90	43350
weighted avg	0.93	0.93	0.93	43350

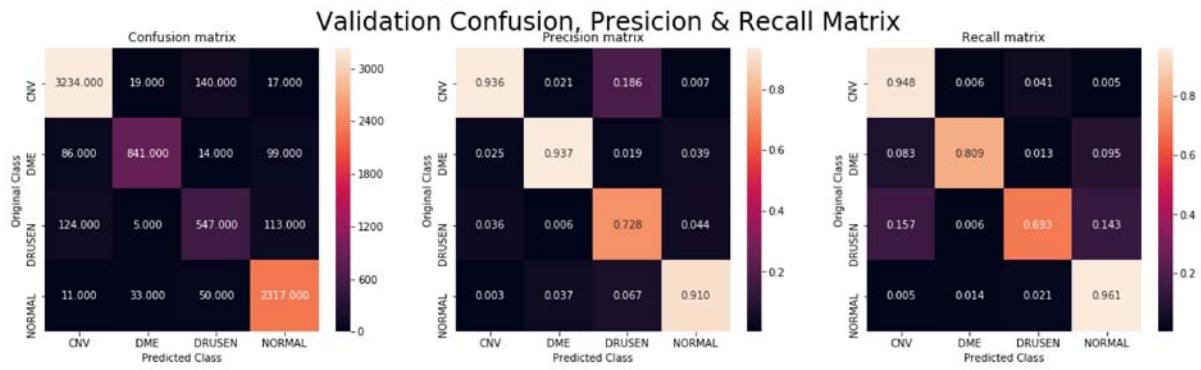
#### Train Confusion Matrix



#### Validation Classification Report

	precision	recall	f1-score	support
CNV	0.94	0.95	0.94	3410
DME	0.94	0.81	0.87	1040
DRUSEN	0.73	0.69	0.71	789
NORMAL	0.91	0.96	0.93	2411
accuracy			0.91	7650
macro avg	0.88	0.85	0.86	7650
weighted avg	0.91	0.91	0.91	7650

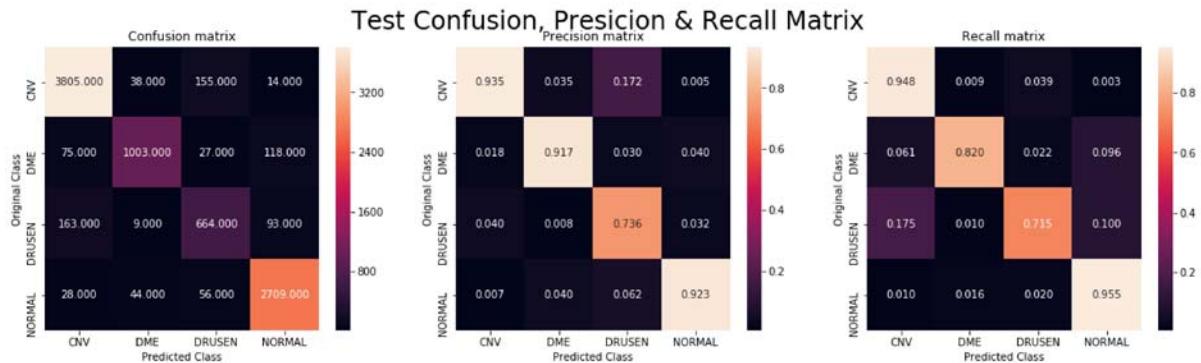
#### Validation Confusion Matrix



### Test Classification Report

	precision	recall	f1-score	support
CNV	0.93	0.95	0.94	4012
DME	0.92	0.82	0.87	1223
DRUSEN	0.74	0.71	0.73	929
NORMAL	0.92	0.95	0.94	2837
accuracy			0.91	9001
macro avg	0.88	0.86	0.87	9001
weighted avg	0.91	0.91	0.91	9001

### Test Confusion Matrix



```
In [28]: metrics.append(metric)
```

```
In [30]: pickle.dump(metrics, open("metrics.p", "wb"))
```

```
In [31]: metrics = pickle.load(open("metrics.p", "rb"))
```

## 4) ReNet50 Transfer Learning

```
In [21]: # Ref: https://github.com/BadreeshShetty/OCT-Retinal/blob/master/OCT-Retinal-VGG.ipynb

base_model = ResNet50(weights='imagenet',include_top=False, pooling='max', input_shape=(224,224,3))

x = base_model.output
x = Dropout(0.6)(x)

output = Dense(4, activation='softmax')(x)
model = Model(inputs=base_model.input, outputs=output)
# Train top layer
for layer in base_model.layers[:3]:
    layer.trainable = False

print(model.summary())
```

WARNING:tensorflow:From /usr/local/lib/python3.5/dist-packages/tensorflow\_core/python/ops/resource\_variable\_ops.py:1630: calling BaseResourceVariable.\_init\_\_ (from tensorflow.python.ops.resource\_variable\_ops) with constraint is deprecated and will be removed in a future version.

Instructions for updating:

If using Keras pass \*\_constraint arguments to layers.

WARNING:tensorflow:From /usr/local/lib/python3.5/dist-packages/keras/backend/tensorflow\_backend.py:4070: The name tf.nn.max\_pool is deprecated. Please use tf.nn.max\_pool2d instead.

WARNING:tensorflow:Large dropout rate: 0.6 (>0.5). In TensorFlow 2.x, dropout() uses dropout rate instead of keep\_prob. Please ensure that this is intended.

Model: "model\_1"

---

Layer (type)	Output Shape	Param #	Connected to
input_1 (InputLayer)	(None, 224, 224, 3)	0	
conv1_pad (ZeroPadding2D) [0]	(None, 230, 230, 3)	0	input_1[0]
conv1 (Conv2D) [0]	(None, 112, 112, 64)	9472	conv1_pad[0]
bn_conv1 (BatchNormalization)	(None, 112, 112, 64)	256	conv1[0][0]
activation_1 (Activation) [0]	(None, 112, 112, 64)	0	bn_conv1[0]
pool1_pad (ZeroPadding2D) [0][0]	(None, 114, 114, 64)	0	activation_1
max_pooling2d_1 (MaxPooling2D) [0]	(None, 56, 56, 64)	0	pool1_pad[0]
res2a_branch2a (Conv2D) d_1[0][0]	(None, 56, 56, 64)	4160	max_pooling2
bn2a_branch2a (BatchNormalizati 2a[0][0]	(None, 56, 56, 64)	256	res2a_branch2a[0]
activation_2 (Activation) a[0][0]	(None, 56, 56, 64)	0	bn2a_branch2a[0]

---

res2a_branch2b (Conv2D) [0][0]	(None, 56, 56, 64)	36928	activation_2
bn2a_branch2b (BatchNormalizati 2b[0][0]	(None, 56, 56, 64)	256	res2a_branch
activation_3 (Activation) b[0][0]	(None, 56, 56, 64)	0	bn2a_branch2
res2a_branch2c (Conv2D) [0][0]	(None, 56, 56, 256)	16640	activation_3
res2a_branch1 (Conv2D) d_1[0][0]	(None, 56, 56, 256)	16640	max_pooling2
bn2a_branch2c (BatchNormalizati 2c[0][0]	(None, 56, 56, 256)	1024	res2a_branch
bn2a_branch1 (BatchNormalizatio 1[0][0]	(None, 56, 56, 256)	1024	res2a_branch
add_1 (Add) c[0][0]	(None, 56, 56, 256)	0	bn2a_branch2
[0][0]			bn2a_branch1
activation_4 (Activation)	(None, 56, 56, 256)	0	add_1[0][0]
res2b_branch2a (Conv2D) [0][0]	(None, 56, 56, 64)	16448	activation_4
bn2b_branch2a (BatchNormalizati 2a[0][0]	(None, 56, 56, 64)	256	res2b_branch
activation_5 (Activation) a[0][0]	(None, 56, 56, 64)	0	bn2b_branch2
res2b_branch2b (Conv2D) [0][0]	(None, 56, 56, 64)	36928	activation_5
bn2b_branch2b (BatchNormalizati 2b[0][0]	(None, 56, 56, 64)	256	res2b_branch

activation_6 (Activation) b[0][0]	(None, 56, 56, 64)	0	bn2b_branch2
res2b_branch2c (Conv2D) [0][0]	(None, 56, 56, 256)	16640	activation_6
bn2b_branch2c (BatchNormalizati 2c[0][0]	(None, 56, 56, 256)	1024	res2b_branch
add_2 (Add) c[0][0]	(None, 56, 56, 256)	0	bn2b_branch2 activation_4
[0][0]			
activation_7 (Activation)	(None, 56, 56, 256)	0	add_2[0][0]
res2c_branch2a (Conv2D) [0][0]	(None, 56, 56, 64)	16448	activation_7
bn2c_branch2a (BatchNormalizati 2a[0][0]	(None, 56, 56, 64)	256	res2c_branch
activation_8 (Activation) a[0][0]	(None, 56, 56, 64)	0	bn2c_branch2
res2c_branch2b (Conv2D) [0][0]	(None, 56, 56, 64)	36928	activation_8
bn2c_branch2b (BatchNormalizati 2b[0][0]	(None, 56, 56, 64)	256	res2c_branch
activation_9 (Activation) b[0][0]	(None, 56, 56, 64)	0	bn2c_branch2
res2c_branch2c (Conv2D) [0][0]	(None, 56, 56, 256)	16640	activation_9
bn2c_branch2c (BatchNormalizati 2c[0][0]	(None, 56, 56, 256)	1024	res2c_branch
add_3 (Add) c[0][0]	(None, 56, 56, 256)	0	bn2c_branch2 activation_7
[0][0]			

activation_10 (Activation)	(None, 56, 56, 256) 0	add_3[0][0]
res3a_branch2a (Conv2D) 0[0][0]	(None, 28, 28, 128) 32896	activation_1
bn3a_branch2a (BatchNormalizati 2a[0][0]	(None, 28, 28, 128) 512	res3a_branch
activation_11 (Activation) a[0][0]	(None, 28, 28, 128) 0	bn3a_branch2
res3a_branch2b (Conv2D) 1[0][0]	(None, 28, 28, 128) 147584	activation_1
bn3a_branch2b (BatchNormalizati 2b[0][0]	(None, 28, 28, 128) 512	res3a_branch
activation_12 (Activation) b[0][0]	(None, 28, 28, 128) 0	bn3a_branch2
res3a_branch2c (Conv2D) 2[0][0]	(None, 28, 28, 512) 66048	activation_1
res3a_branch1 (Conv2D) 0[0][0]	(None, 28, 28, 512) 131584	activation_1
bn3a_branch2c (BatchNormalizati 2c[0][0]	(None, 28, 28, 512) 2048	res3a_branch
bn3a_branch1 (BatchNormalizatio 1[0][0]	(None, 28, 28, 512) 2048	res3a_branch
add_4 (Add) c[0][0]	(None, 28, 28, 512) 0	bn3a_branch2
[0][0]		bn3a_branch1
activation_13 (Activation)	(None, 28, 28, 512) 0	add_4[0][0]
res3b_branch2a (Conv2D) 3[0][0]	(None, 28, 28, 128) 65664	activation_1

bn3b_branch2a (BatchNormalizati 2a[0][0]	(None, 28, 28, 128)	512	res3b_branch
activation_14 (Activation) a[0][0]	(None, 28, 28, 128)	0	bn3b_branch2
res3b_branch2b (Conv2D) 4[0][0]	(None, 28, 28, 128)	147584	activation_1
bn3b_branch2b (BatchNormalizati 2b[0][0]	(None, 28, 28, 128)	512	res3b_branch
activation_15 (Activation) b[0][0]	(None, 28, 28, 128)	0	bn3b_branch2
res3b_branch2c (Conv2D) 5[0][0]	(None, 28, 28, 512)	66048	activation_1
bn3b_branch2c (BatchNormalizati 2c[0][0]	(None, 28, 28, 512)	2048	res3b_branch
add_5 (Add) c[0][0]	(None, 28, 28, 512)	0	bn3b_branch2
activation_16 (Activation) 3[0][0]	(None, 28, 28, 512)	0	activation_1
res3c_branch2a (Conv2D) 6[0][0]	(None, 28, 28, 128)	65664	activation_1
bn3c_branch2a (BatchNormalizati 2a[0][0]	(None, 28, 28, 128)	512	res3c_branch
activation_17 (Activation) a[0][0]	(None, 28, 28, 128)	0	bn3c_branch2
res3c_branch2b (Conv2D) 7[0][0]	(None, 28, 28, 128)	147584	activation_1
bn3c_branch2b (BatchNormalizati 2b[0][0]	(None, 28, 28, 128)	512	res3c_branch

activation_18 (Activation) b[0][0]	(None, 28, 28, 128) 0	bn3c_branch2
res3c_branch2c (Conv2D) 8[0][0]	(None, 28, 28, 512) 66048	activation_1
bn3c_branch2c (BatchNormalizati 2c[0][0]	(None, 28, 28, 512) 2048	res3c_branch
add_6 (Add) c[0][0]	(None, 28, 28, 512) 0	bn3c_branch2
activation_1	6[0][0]	activation_1
activation_19 (Activation)	(None, 28, 28, 512) 0	add_6[0][0]
res3d_branch2a (Conv2D) 9[0][0]	(None, 28, 28, 128) 65664	activation_1
bn3d_branch2a (BatchNormalizati 2a[0][0]	(None, 28, 28, 128) 512	res3d_branch
activation_20 (Activation) a[0][0]	(None, 28, 28, 128) 0	bn3d_branch2
res3d_branch2b (Conv2D) 0[0][0]	(None, 28, 28, 128) 147584	activation_2
bn3d_branch2b (BatchNormalizati 2b[0][0]	(None, 28, 28, 128) 512	res3d_branch
activation_21 (Activation) b[0][0]	(None, 28, 28, 128) 0	bn3d_branch2
res3d_branch2c (Conv2D) 1[0][0]	(None, 28, 28, 512) 66048	activation_2
bn3d_branch2c (BatchNormalizati 2c[0][0]	(None, 28, 28, 512) 2048	res3d_branch
add_7 (Add) c[0][0]	(None, 28, 28, 512) 0	bn3d_branch2
activation_1		

9[0][0]

activation_22 (Activation)	(None, 28, 28, 512) 0	add_7[0][0]
res4a_branch2a (Conv2D) 2[0][0]	(None, 14, 14, 256) 131328	activation_2
bn4a_branch2a (BatchNormalizati 2a[0][0]	(None, 14, 14, 256) 1024	res4a_branch
activation_23 (Activation) a[0][0]	(None, 14, 14, 256) 0	bn4a_branch2
res4a_branch2b (Conv2D) 3[0][0]	(None, 14, 14, 256) 590080	activation_2
bn4a_branch2b (BatchNormalizati 2b[0][0]	(None, 14, 14, 256) 1024	res4a_branch
activation_24 (Activation) b[0][0]	(None, 14, 14, 256) 0	bn4a_branch2
res4a_branch2c (Conv2D) 4[0][0]	(None, 14, 14, 1024) 263168	activation_2
res4a_branch1 (Conv2D) 2[0][0]	(None, 14, 14, 1024) 525312	activation_2
bn4a_branch2c (BatchNormalizati 2c[0][0]	(None, 14, 14, 1024) 4096	res4a_branch
bn4a_branch1 (BatchNormalizatio 1[0][0]	(None, 14, 14, 1024) 4096	res4a_branch
add_8 (Add) c[0][0]	(None, 14, 14, 1024) 0	bn4a_branch2
[0][0]		bn4a_branch1
activation_25 (Activation)	(None, 14, 14, 1024) 0	add_8[0][0]
res4b_branch2a (Conv2D) 5[0][0]	(None, 14, 14, 256) 262400	activation_2

bn4b_branch2a (BatchNormalizati 2a[0][0]	(None, 14, 14, 256) 1024	res4b_branch
activation_26 (Activation) a[0][0]	(None, 14, 14, 256) 0	bn4b_branch2
res4b_branch2b (Conv2D) 6[0][0]	(None, 14, 14, 256) 590080	activation_2
bn4b_branch2b (BatchNormalizati 2b[0][0]	(None, 14, 14, 256) 1024	res4b_branch
activation_27 (Activation) b[0][0]	(None, 14, 14, 256) 0	bn4b_branch2
res4b_branch2c (Conv2D) 7[0][0]	(None, 14, 14, 1024) 263168	activation_2
bn4b_branch2c (BatchNormalizati 2c[0][0]	(None, 14, 14, 1024) 4096	res4b_branch
add_9 (Add) c[0][0]	(None, 14, 14, 1024) 0	bn4b_branch2
activation_28 (Activation) 5[0][0]	(None, 14, 14, 1024) 0	activation_2
activation_28 (Activation)	(None, 14, 14, 1024) 0	add_9[0][0]
res4c_branch2a (Conv2D) 8[0][0]	(None, 14, 14, 256) 262400	activation_2
bn4c_branch2a (BatchNormalizati 2a[0][0]	(None, 14, 14, 256) 1024	res4c_branch
activation_29 (Activation) a[0][0]	(None, 14, 14, 256) 0	bn4c_branch2
res4c_branch2b (Conv2D) 9[0][0]	(None, 14, 14, 256) 590080	activation_2
bn4c_branch2b (BatchNormalizati 2b[0][0]	(None, 14, 14, 256) 1024	res4c_branch

activation_30 (Activation) b[0][0]	(None, 14, 14, 256) 0	bn4c_branch2
res4c_branch2c (Conv2D) 0[0][0]	(None, 14, 14, 1024) 263168	activation_3
bn4c_branch2c (BatchNormalizati 2c[0][0]	(None, 14, 14, 1024) 4096	res4c_branch
add_10 (Add) c[0][0]	(None, 14, 14, 1024) 0	bn4c_branch2
		activation_2
8[0][0]		
activation_31 (Activation)	(None, 14, 14, 1024) 0	add_10[0][0]
res4d_branch2a (Conv2D) 1[0][0]	(None, 14, 14, 256) 262400	activation_3
bn4d_branch2a (BatchNormalizati 2a[0][0]	(None, 14, 14, 256) 1024	res4d_branch
activation_32 (Activation) a[0][0]	(None, 14, 14, 256) 0	bn4d_branch2
res4d_branch2b (Conv2D) 2[0][0]	(None, 14, 14, 256) 590080	activation_3
bn4d_branch2b (BatchNormalizati 2b[0][0]	(None, 14, 14, 256) 1024	res4d_branch
activation_33 (Activation) b[0][0]	(None, 14, 14, 256) 0	bn4d_branch2
res4d_branch2c (Conv2D) 3[0][0]	(None, 14, 14, 1024) 263168	activation_3
bn4d_branch2c (BatchNormalizati 2c[0][0]	(None, 14, 14, 1024) 4096	res4d_branch
add_11 (Add) c[0][0]	(None, 14, 14, 1024) 0	bn4d_branch2

activation\_3

1[0][0]

---

activation\_34 (Activation) (None, 14, 14, 1024) 0 add\_11[0][0]

---

res4e\_branch2a (Conv2D) (None, 14, 14, 256) 262400 activation\_3  
4[0][0]

---

bn4e\_branch2a (BatchNormalizati (None, 14, 14, 256) 1024 res4e\_branch  
2a[0][0]

---

activation\_35 (Activation) (None, 14, 14, 256) 0 bn4e\_branch2  
a[0][0]

---

res4e\_branch2b (Conv2D) (None, 14, 14, 256) 590080 activation\_3  
5[0][0]

---

bn4e\_branch2b (BatchNormalizati (None, 14, 14, 256) 1024 res4e\_branch  
2b[0][0]

---

activation\_36 (Activation) (None, 14, 14, 256) 0 bn4e\_branch2  
b[0][0]

---

res4e\_branch2c (Conv2D) (None, 14, 14, 1024) 263168 activation\_3  
6[0][0]

---

bn4e\_branch2c (BatchNormalizati (None, 14, 14, 1024) 4096 res4e\_branch  
2c[0][0]

---

add\_12 (Add) (None, 14, 14, 1024) 0 bn4e\_branch2  
c[0][0]
activation\_3  
4[0][0]

---

activation\_37 (Activation) (None, 14, 14, 1024) 0 add\_12[0][0]

---

res4f\_branch2a (Conv2D) (None, 14, 14, 256) 262400 activation\_3  
7[0][0]

---

bn4f\_branch2a (BatchNormalizati (None, 14, 14, 256) 1024 res4f\_branch  
2a[0][0]

---

activation\_38 (Activation) (None, 14, 14, 256) 0 bn4f\_branch2

a[0][0]

---

res4f_branch2b (Conv2D) 8[0][0]	(None, 14, 14, 256)	590080	activation_3
bn4f_branch2b (BatchNormalizati 2b[0][0]	(None, 14, 14, 256)	1024	res4f_branch
activation_39 (Activation) b[0][0]	(None, 14, 14, 256)	0	bn4f_branch2
res4f_branch2c (Conv2D) 9[0][0]	(None, 14, 14, 1024)	263168	activation_3
bn4f_branch2c (BatchNormalizati 2c[0][0]	(None, 14, 14, 1024)	4096	res4f_branch
add_13 (Add) c[0][0]	(None, 14, 14, 1024)	0	bn4f_branch2
activation_3 7[0][0]			activation_3
activation_40 (Activation)	(None, 14, 14, 1024)	0	add_13[0][0]
res5a_branch2a (Conv2D) 0[0][0]	(None, 7, 7, 512)	524800	activation_4
bn5a_branch2a (BatchNormalizati 2a[0][0]	(None, 7, 7, 512)	2048	res5a_branch
activation_41 (Activation) a[0][0]	(None, 7, 7, 512)	0	bn5a_branch2
res5a_branch2b (Conv2D) 1[0][0]	(None, 7, 7, 512)	2359808	activation_4
bn5a_branch2b (BatchNormalizati 2b[0][0]	(None, 7, 7, 512)	2048	res5a_branch
activation_42 (Activation) b[0][0]	(None, 7, 7, 512)	0	bn5a_branch2
res5a_branch2c (Conv2D)	(None, 7, 7, 2048)	1050624	activation_4

2[0][0]

res5a_branch1 (Conv2D) 0[0][0]	(None, 7, 7, 2048)	2099200	activation_4
bn5a_branch2c (BatchNormalizati 2c[0][0]	(None, 7, 7, 2048)	8192	res5a_branch
bn5a_branch1 (BatchNormalizatio 1[0][0]	(None, 7, 7, 2048)	8192	res5a_branch
add_14 (Add) c[0][0]	(None, 7, 7, 2048)	0	bn5a_branch2
			bn5a_branch1
[0][0]			
activation_43 (Activation)	(None, 7, 7, 2048)	0	add_14[0][0]
res5b_branch2a (Conv2D) 3[0][0]	(None, 7, 7, 512)	1049088	activation_4
bn5b_branch2a (BatchNormalizati 2a[0][0]	(None, 7, 7, 512)	2048	res5b_branch
activation_44 (Activation) a[0][0]	(None, 7, 7, 512)	0	bn5b_branch2
res5b_branch2b (Conv2D) 4[0][0]	(None, 7, 7, 512)	2359808	activation_4
bn5b_branch2b (BatchNormalizati 2b[0][0]	(None, 7, 7, 512)	2048	res5b_branch
activation_45 (Activation) b[0][0]	(None, 7, 7, 512)	0	bn5b_branch2
res5b_branch2c (Conv2D) 5[0][0]	(None, 7, 7, 2048)	1050624	activation_4
bn5b_branch2c (BatchNormalizati 2c[0][0]	(None, 7, 7, 2048)	8192	res5b_branch
add_15 (Add)	(None, 7, 7, 2048)	0	bn5b_branch2

c[0][0]				activation_4
3[0][0]				
activation_46 (Activation)	(None, 7, 7, 2048)	0		add_15[0][0]
res5c_branch2a (Conv2D)	(None, 7, 7, 512)	1049088		activation_4
6[0][0]				
bn5c_branch2a (BatchNormalizati	(None, 7, 7, 512)	2048		res5c_branch
2a[0][0]				
activation_47 (Activation)	(None, 7, 7, 512)	0		bn5c_branch2
a[0][0]				
res5c_branch2b (Conv2D)	(None, 7, 7, 512)	2359808		activation_4
7[0][0]				
bn5c_branch2b (BatchNormalizati	(None, 7, 7, 512)	2048		res5c_branch
2b[0][0]				
activation_48 (Activation)	(None, 7, 7, 512)	0		bn5c_branch2
b[0][0]				
res5c_branch2c (Conv2D)	(None, 7, 7, 2048)	1050624		activation_4
8[0][0]				
bn5c_branch2c (BatchNormalizati	(None, 7, 7, 2048)	8192		res5c_branch
2c[0][0]				
add_16 (Add)	(None, 7, 7, 2048)	0		bn5c_branch2
c[0][0]				
6[0][0]				activation_4
activation_49 (Activation)	(None, 7, 7, 2048)	0		add_16[0][0]
global_max_pooling2d_1 (GlobalM	(None, 2048)	0		activation_4
9[0][0]				
dropout_1 (Dropout)	(None, 2048)	0		global_max_p
ooling2d_1[0][0]				

```
dense_1 (Dense)           (None, 4)          8196      dropout_1[0]  
[0]  
=====  
=====  
Total params: 23,595,908  
Trainable params: 23,533,316  
Non-trainable params: 62,592
```

---

---

None



```
In [22]: model, history = train_model(model, batch_size = 64, epochs = 20, model_no = 3  
)
```

```
WARNING:tensorflow:From /usr/local/lib/python3.5/dist-packages/keras/backend/
tensorflow_backend.py:422: The name tf.global_variables is deprecated. Please
use tf.compat.v1.global_variables instead.

WARNING:tensorflow:From /usr/local/lib/python3.5/dist-packages/keras/callbacks/tensorboard_v1.py:200: The name tf.summary.merge_all is deprecated. Please
use tf.compat.v1.summary.merge_all instead.

WARNING:tensorflow:From /usr/local/lib/python3.5/dist-packages/keras/callbacks/tensorboard_v1.py:203: The name tf.summary.FileWriter is deprecated. Please
use tf.compat.v1.summary.FileWriter instead.

Epoch 1/20
678/678 [=====] - 1114s 2s/step - loss: 1.0415 - f1_
micro_score: 0.8261 - val_loss: 1.3597 - val_f1_micro_score: 0.4454

Epoch 00001: val_loss improved from inf to 1.35966, saving model to model_4.h
df5
WARNING:tensorflow:From /usr/local/lib/python3.5/dist-packages/keras/callbacks/tensorboard_v1.py:343: The name tf.Summary is deprecated. Please use tf.com
pat.v1.Summary instead.

Epoch 2/20
678/678 [=====] - 1104s 2s/step - loss: 0.8665 - f1_
micro_score: 0.8691 - val_loss: 0.3800 - val_f1_micro_score: 0.9019

Epoch 00002: val_loss improved from 1.35966 to 0.38003, saving model to model
_4.hdf5
Epoch 3/20
678/678 [=====] - 1106s 2s/step - loss: 0.3596 - f1_
micro_score: 0.8989 - val_loss: 0.3202 - val_f1_micro_score: 0.8918

Epoch 00003: val_loss improved from 0.38003 to 0.32025, saving model to model
_4.hdf5
Epoch 4/20
678/678 [=====] - 1100s 2s/step - loss: 0.2481 - f1_
micro_score: 0.9225 - val_loss: 0.2942 - val_f1_micro_score: 0.9283

Epoch 00004: val_loss improved from 0.32025 to 0.29422, saving model to model
_4.hdf5
Epoch 5/20
678/678 [=====] - 1103s 2s/step - loss: 1.2794 - f1_
micro_score: 0.8394 - val_loss: 0.5904 - val_f1_micro_score: 0.7428

Epoch 00005: val_loss did not improve from 0.29422

Epoch 00005: ReduceLROnPlateau reducing learning rate to 0.00020000009499490
26.

Epoch 6/20
678/678 [=====] - 1100s 2s/step - loss: 0.4507 - f1_
micro_score: 0.9195 - val_loss: 0.3185 - val_f1_micro_score: 0.9296

Epoch 00006: val_loss did not improve from 0.29422

Epoch 00006: ReduceLROnPlateau reducing learning rate to 4.0000001899898055e-
05.

Epoch 7/20
```

```
678/678 [=====] - 1102s 2s/step - loss: 0.2041 - f1_
micro_score: 0.9355 - val_loss: 0.2734 - val_f1_micro_score: 0.9364

Epoch 00007: val_loss improved from 0.29422 to 0.27338, saving model to model
_4.hdf5
Epoch 8/20
678/678 [=====] - 1099s 2s/step - loss: 0.1857 - f1_
micro_score: 0.9397 - val_loss: 0.2791 - val_f1_micro_score: 0.9383

Epoch 00008: val_loss did not improve from 0.27338

Epoch 00008: ReduceLROnPlateau reducing learning rate to 8.00000525498762e-0
6.
Epoch 9/20
678/678 [=====] - 1101s 2s/step - loss: 0.1769 - f1_
micro_score: 0.9417 - val_loss: 0.2814 - val_f1_micro_score: 0.9398

Epoch 00009: val_loss did not improve from 0.27338

Epoch 00009: ReduceLROnPlateau reducing learning rate to 1.6000001778593287e-
06.
Epoch 10/20
678/678 [=====] - 1103s 2s/step - loss: 0.1742 - f1_
micro_score: 0.9428 - val_loss: 0.2745 - val_f1_micro_score: 0.9395

Epoch 00010: val_loss did not improve from 0.27338
Restoring model weights from the end of the best epoch

Epoch 00010: ReduceLROnPlateau reducing learning rate to 3.200000264769187e-0
7.
Epoch 00010: early stopping
```

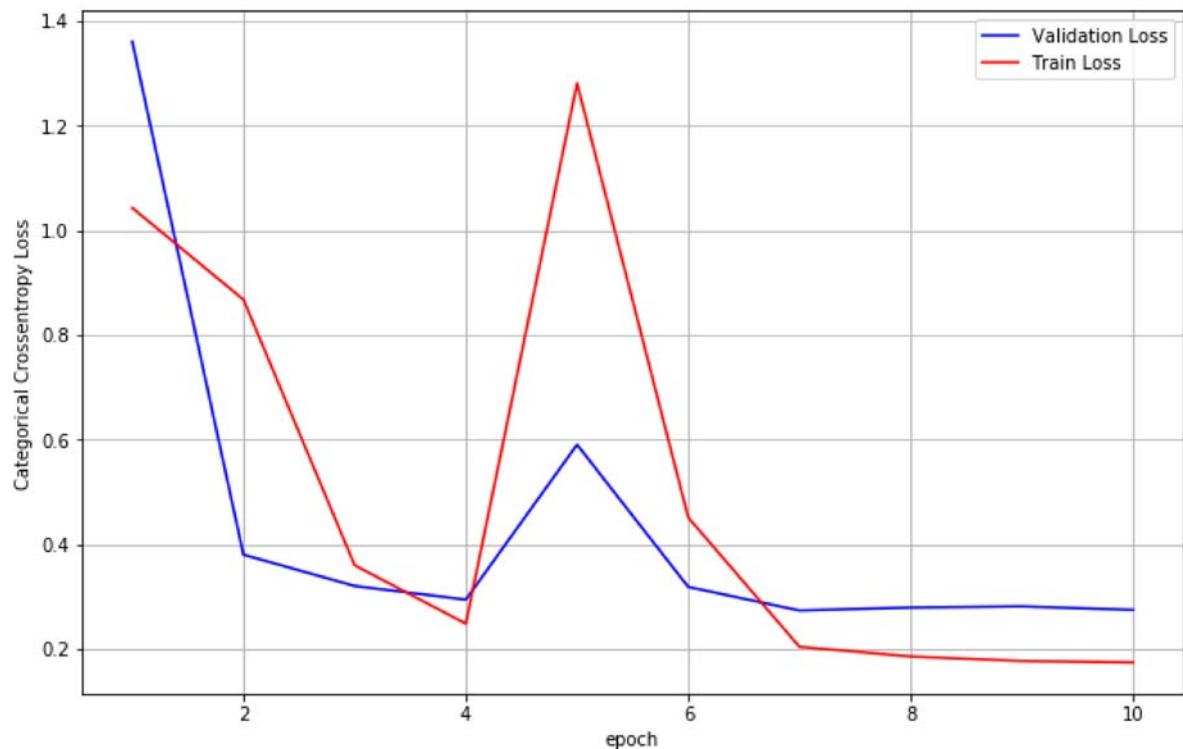
```
In [86]: model = load_model("model_4.hdf5", custom_objects={"f1_micro_
score"})
```

```
In [23]: nb_epoch = len(history.history['loss'])
%matplotlib inline
import matplotlib.pyplot as plt
plt.rcParams["figure.figsize"] = [11, 7]

fig,ax = plt.subplots(1,1)
ax.set_xlabel('epoch') ; ax.set_ylabel('Categorical Crossentropy Loss')

# List of epoch numbers
x = list(range(1,nb_epoch+1))

vy = history.history['val_loss']
ty = history.history['loss']
plt_dynamic(x, vy, ty, ax, "Loss")
```

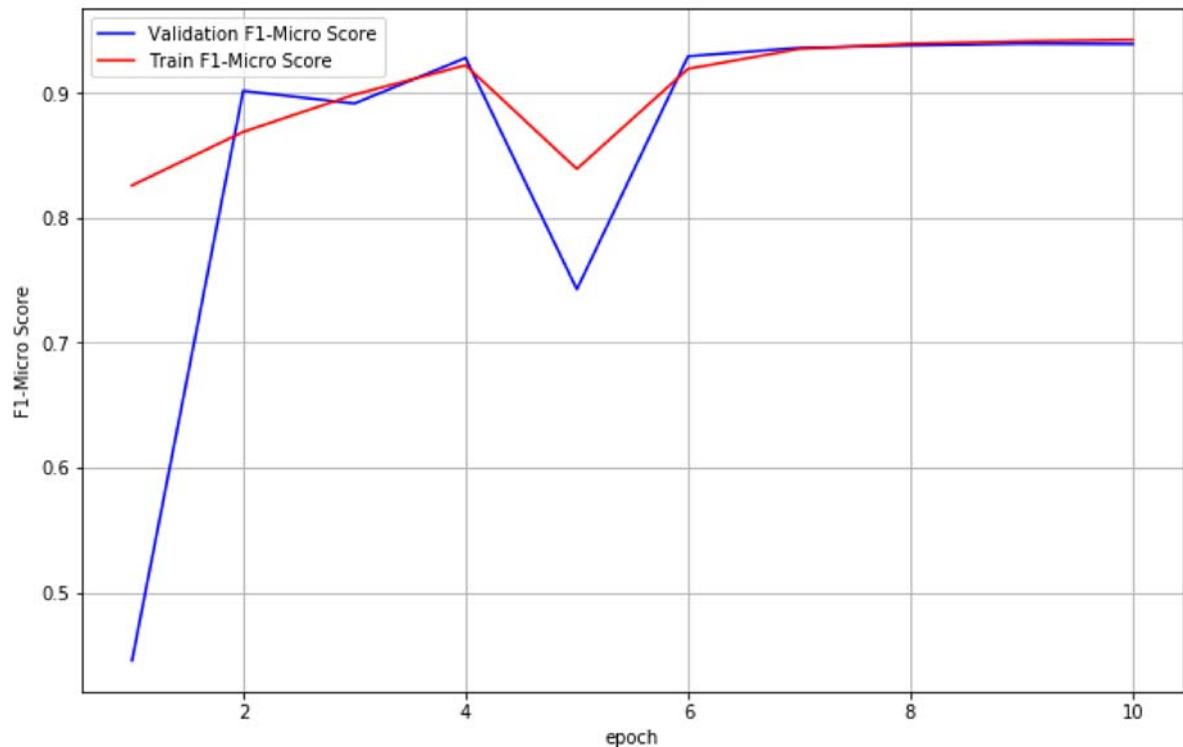


```
In [24]: nb_epoch = len(history.history['f1_micro_score'])
%matplotlib inline
import matplotlib.pyplot as plt
plt.rcParams["figure.figsize"] = [11, 7]

fig,ax = plt.subplots(1,1)
ax.set_xlabel('epoch') ; ax.set_ylabel('F1-Micro Score')

# List of epoch numbers
x = list(range(1,nb_epoch+1))

vy = history.history['val_f1_micro_score']
ty = history.history['f1_micro_score']
plt_dynamic(x, vy, ty, ax, "F1-Micro Score")
```



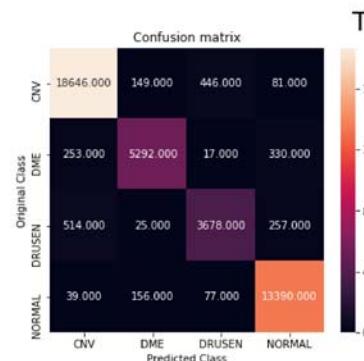
```
In [25]: metric = report_of_model(model, model_name = "ResNet50 Transfer Learning", batch_size = 64)
```

```

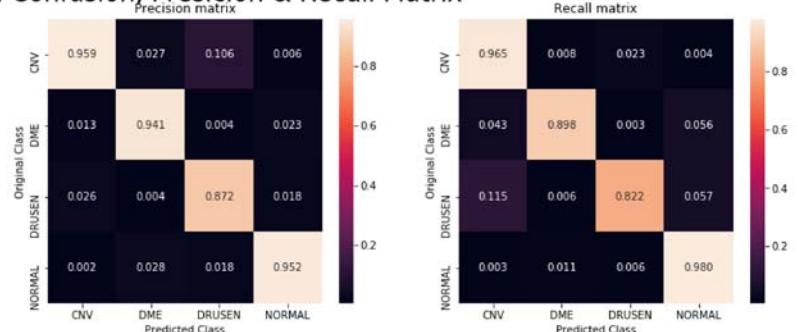
678/678 [=====] - 347s 512ms/step
120/120 [=====] - 62s 513ms/step
141/141 [=====] - 74s 522ms/step
=====
Train log-loss : 0.019
Validation log-loss : 0.273
Test log-loss : 0.291
=====
678/678 [=====] - 356s 524ms/step
120/120 [=====] - 63s 523ms/step
141/141 [=====] - 74s 523ms/step
=====
Train F1-micro avg score : 0.9459
Val F1-micro avg score : 0.9366
Test F1-micro avg score : 0.9399
=====
Train Classification Report
      precision    recall   f1-score   support
CNV        0.96     0.97     0.96    19322
DME        0.94     0.90     0.92     5892
DRUSEN     0.87     0.82     0.85     4474
NORMAL     0.95     0.98     0.97    13662
accuracy          0.95     0.95     0.95    43350
macro avg       0.93     0.92     0.92    43350
weighted avg    0.95     0.95     0.95    43350

```

#### Train Confusion Matrix



#### Train Confusion, Precision & Recall Matrix



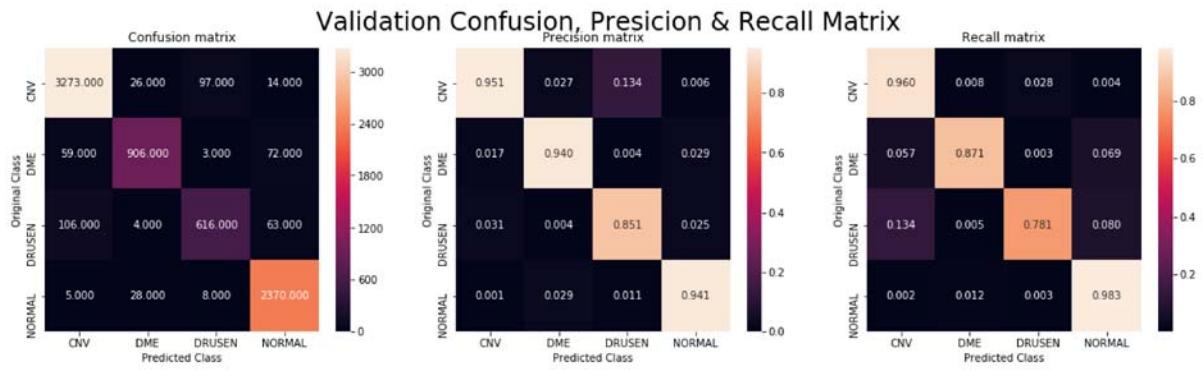
#### Validation Classification Report

```

      precision    recall   f1-score   support
CNV        0.95     0.96     0.96    3410
DME        0.94     0.87     0.90    1040
DRUSEN     0.85     0.78     0.81     789
NORMAL     0.94     0.98     0.96    2411
accuracy          0.94     0.94     0.94    7650
macro avg       0.92     0.90     0.91    7650
weighted avg    0.94     0.94     0.94    7650

```

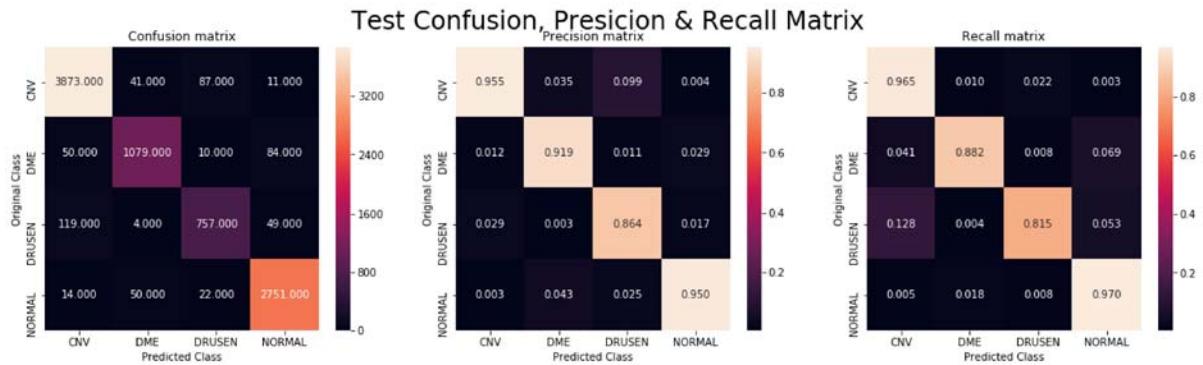
#### Validation Confusion Matrix



### Test Classification Report

	precision	recall	f1-score	support
CNV	0.95	0.97	0.96	4012
DME	0.92	0.88	0.90	1223
DRUSEN	0.86	0.81	0.84	929
NORMAL	0.95	0.97	0.96	2837
accuracy			0.94	9001
macro avg	0.92	0.91	0.91	9001
weighted avg	0.94	0.94	0.94	9001

### Test Confusion Matrix



```
In [26]: metrics.append(metric)
```

```
In [27]: pickle.dump(metrics, open("metrics.p", "wb"))
```

```
In [20]: metrics = pickle.load(open("metrics.p", "rb"))
```

## 5) DenseNet121 Transfer Learning

```
In [22]: base_model = DenseNet121(weights='imagenet',include_top=False, pooling='max',
input_shape=(224,224,3))

x = base_model.output
x = Dropout(0.4)(x)

output = Dense(4, activation='softmax')(x)
model = Model(inputs=base_model.input, outputs=output)
# Train top layer
for layer in base_model.layers[:3]:
    layer.trainable = False

print(model.summary())
```

WARNING:tensorflow:From /usr/local/lib/python3.5/dist-packages/tensorflow\_core/python/ops/resource\_variable\_ops.py:1630: calling BaseResourceVariable.\_init\_\_ (from tensorflow.python.ops.resource\_variable\_ops) with constraint is deprecated and will be removed in a future version.

Instructions for updating:

If using Keras pass \*\_constraint arguments to layers.

WARNING:tensorflow:From /usr/local/lib/python3.5/dist-packages/keras/backend/tensorflow\_backend.py:4070: The name tf.nn.max\_pool is deprecated. Please use tf.nn.max\_pool2d instead.

WARNING:tensorflow:From /usr/local/lib/python3.5/dist-packages/keras/backend/tensorflow\_backend.py:4074: The name tf.nn.avg\_pool is deprecated. Please use tf.nn.avg\_pool2d instead.

Model: "model\_1"

---

Layer (type)	Output Shape	Param #	Connected to
input_1 (InputLayer)	(None, 224, 224, 3)	0	
zero_padding2d_1 (ZeroPadding2D)	(None, 230, 230, 3)	0	input_1[0]
[0]			
conv1/conv (Conv2D)	(None, 112, 112, 64)	9408	zero_padding
2d_1[0][0]			
conv1/bn (BatchNormalization)	(None, 112, 112, 64)	256	conv1/conv
[0][0]			
conv1/relu (Activation)	(None, 112, 112, 64)	0	conv1/bn[0]
[0]			
zero_padding2d_2 (ZeroPadding2D)	(None, 114, 114, 64)	0	conv1/relu
[0][0]			
pool1 (MaxPooling2D)	(None, 56, 56, 64)	0	zero_padding
2d_2[0][0]			
conv2_block1_0_bn (BatchNormali	(None, 56, 56, 64)	256	pool1[0][0]
zation			
conv2_block1_0_relu (Activation)	(None, 56, 56, 64)	0	conv2_block1
_0_bn[0][0]			
conv2_block1_1_conv (Conv2D)	(None, 56, 56, 128)	8192	conv2_block1
_0_relu[0][0]			

---

conv2_block1_1_bn (BatchNormali (None, 56, 56, 128) 512		conv2_block1
_1_conv[0][0]		
conv2_block1_1_relu (Activation (None, 56, 56, 128) 0		conv2_block1
_1_bn[0][0]		
conv2_block1_2_conv (Conv2D) (None, 56, 56, 32) 36864		conv2_block1
_1_relu[0][0]		
conv2_block1_concat (Concatenat (None, 56, 56, 96) 0		pool1[0][0]
_2_conv[0][0]		conv2_block1
conv2_block2_0_bn (BatchNormali (None, 56, 56, 96) 384		conv2_block1
_concat[0][0]		
conv2_block2_0_relu (Activation (None, 56, 56, 96) 0		conv2_block2
_0_bn[0][0]		
conv2_block2_1_conv (Conv2D) (None, 56, 56, 128) 12288		conv2_block2
_0_relu[0][0]		
conv2_block2_1_bn (BatchNormali (None, 56, 56, 128) 512		conv2_block2
_1_conv[0][0]		
conv2_block2_1_relu (Activation (None, 56, 56, 128) 0		conv2_block2
_1_bn[0][0]		
conv2_block2_2_conv (Conv2D) (None, 56, 56, 32) 36864		conv2_block2
_1_relu[0][0]		
conv2_block2_concat (Concatenat (None, 56, 56, 128) 0		conv2_block1
_concat[0][0]		conv2_block2
_2_conv[0][0]		
conv2_block3_0_bn (BatchNormali (None, 56, 56, 128) 512		conv2_block2
_concat[0][0]		
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_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_concat (Concatenat (None, 56, 56, 256) 0	conv2_block5_concat[0][0]	
conv2_block6_2_conv[0][0]	conv2_block6_2_conv[0][0]	
pool2_bn (BatchNormalization) (None, 56, 56, 256) 1024	conv2_block6_concat[0][0]	
	conv2_block6_2_conv[0][0]	

pool2_relu (Activation) [0]	(None, 56, 56, 256) 0	pool2_bn[0]
pool2_conv (Conv2D) [0][0]	(None, 56, 56, 128) 32768	pool2_relu
pool2_pool (AveragePooling2D) [0][0]	(None, 28, 28, 128) 0	pool2_conv
conv3_block1_0_bn (BatchNormali [0][0]	(None, 28, 28, 128) 512	pool2_pool
conv3_block1_0_relu (Activation [0][0]	(None, 28, 28, 128) 0	conv3_block1 _0_bn[0][0]
conv3_block1_1_conv (Conv2D) _0_relu[0][0]	(None, 28, 28, 128) 16384	conv3_block1
conv3_block1_1_bn (BatchNormali [0][0]	(None, 28, 28, 128) 512	conv3_block1 _1_conv[0][0]
conv3_block1_1_relu (Activation [0][0]	(None, 28, 28, 128) 0	conv3_block1 _1_bn[0][0]
conv3_block1_2_conv (Conv2D) _1_relu[0][0]	(None, 28, 28, 32) 36864	conv3_block1
conv3_block1_concat (Concatenat [0][0]	(None, 28, 28, 160) 0	pool2_pool
conv3_block1_2_conv[0][0]		conv3_block1
conv3_block2_0_bn (BatchNormali _concat[0][0]	(None, 28, 28, 160) 640	conv3_block1
conv3_block2_0_relu (Activation _0_bn[0][0]	(None, 28, 28, 160) 0	conv3_block2
conv3_block2_1_conv (Conv2D) _0_relu[0][0]	(None, 28, 28, 128) 20480	conv3_block2
conv3_block2_1_bn (BatchNormali _1_conv[0][0]	(None, 28, 28, 128) 512	conv3_block2

conv3_block2_1_relu (Activation (None, 28, 28, 128) 0 _1_bn[0][0]			conv3_block2
conv3_block2_2_conv (Conv2D) (None, 28, 28, 32) 36864 _1_relu[0][0]			conv3_block2
conv3_block2_concat (Concatenat (None, 28, 28, 192) 0 _concat[0][0]			conv3_block1
			conv3_block2
_2_conv[0][0]			
conv3_block3_0_bn (BatchNormali (None, 28, 28, 192) 768 _concat[0][0]			conv3_block2
conv3_block3_0_relu (Activation (None, 28, 28, 192) 0 _0_bn[0][0]			conv3_block3
conv3_block3_1_conv (Conv2D) (None, 28, 28, 128) 24576 _0_relu[0][0]			conv3_block3
conv3_block3_1_bn (BatchNormali (None, 28, 28, 128) 512 _1_conv[0][0]			conv3_block3
conv3_block3_1_relu (Activation (None, 28, 28, 128) 0 _1_bn[0][0]			conv3_block3
conv3_block3_2_conv (Conv2D) (None, 28, 28, 32) 36864 _1_relu[0][0]			conv3_block3
conv3_block3_concat (Concatenat (None, 28, 28, 224) 0 _concat[0][0]			conv3_block2
			conv3_block3
_2_conv[0][0]			
conv3_block4_0_bn (BatchNormali (None, 28, 28, 224) 896 _concat[0][0]			conv3_block3
conv3_block4_0_relu (Activation (None, 28, 28, 224) 0 _0_bn[0][0]			conv3_block4
conv3_block4_1_conv (Conv2D) (None, 28, 28, 128) 28672 _0_relu[0][0]			conv3_block4

conv3_block4_1_bn (BatchNormali (None, 28, 28, 128) 512 _1_conv[0][0]		conv3_block4
conv3_block4_1_relu (Activation (None, 28, 28, 128) 0 _1_bn[0][0]		conv3_block4
conv3_block4_2_conv (Conv2D) (None, 28, 28, 32) 36864 _1_relu[0][0]		conv3_block4
conv3_block4_concat (Concatenat (None, 28, 28, 256) 0 _concat[0][0]		conv3_block3
		conv3_block4
_2_conv[0][0]		
conv3_block5_0_bn (BatchNormali (None, 28, 28, 256) 1024 _concat[0][0]		conv3_block4
conv3_block5_0_relu (Activation (None, 28, 28, 256) 0 _0_bn[0][0]		conv3_block5
conv3_block5_1_conv (Conv2D) (None, 28, 28, 128) 32768 _0_relu[0][0]		conv3_block5
conv3_block5_1_bn (BatchNormali (None, 28, 28, 128) 512 _1_conv[0][0]		conv3_block5
conv3_block5_1_relu (Activation (None, 28, 28, 128) 0 _1_bn[0][0]		conv3_block5
conv3_block5_2_conv (Conv2D) (None, 28, 28, 32) 36864 _1_relu[0][0]		conv3_block5
conv3_block5_concat (Concatenat (None, 28, 28, 288) 0 _concat[0][0]		conv3_block4
		conv3_block5
_2_conv[0][0]		
conv3_block6_0_bn (BatchNormali (None, 28, 28, 288) 1152 _concat[0][0]		conv3_block5
conv3_block6_0_relu (Activation (None, 28, 28, 288) 0 _0_bn[0][0]		conv3_block6

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_block6
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_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_block7
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 0_0_bn[0][0]			conv3_block1
conv3_block10_1_conv (Conv2D) (None, 28, 28, 128) 53248 0_0_relu[0][0]			conv3_block1
conv3_block10_1_bn (BatchNormal (None, 28, 28, 128) 512 0_1_conv[0][0]			conv3_block1
conv3_block10_1_relu (Activatio (None, 28, 28, 128) 0 0_1_bn[0][0]			conv3_block1
conv3_block10_2_conv (Conv2D) (None, 28, 28, 32) 36864 0_1_relu[0][0]			conv3_block1
conv3_block10_concat (Concatena (None, 28, 28, 448) 0 _concat[0][0]			conv3_block9
conv3_block10_2_conv[0][0]			conv3_block1
conv3_block11_0_bn (BatchNormal (None, 28, 28, 448) 1792 0_concat[0][0]			conv3_block1
conv3_block11_0_relu (Activatio (None, 28, 28, 448) 0 1_0_bn[0][0]			conv3_block1
conv3_block11_1_conv (Conv2D) (None, 28, 28, 128) 57344 1_0_relu[0][0]			conv3_block1
conv3_block11_1_bn (BatchNormal (None, 28, 28, 128) 512 1_1_conv[0][0]			conv3_block1
conv3_block11_1_relu (Activatio (None, 28, 28, 128) 0 1_1_bn[0][0]			conv3_block1
conv3_block11_2_conv (Conv2D) (None, 28, 28, 32) 36864 1_1_relu[0][0]			conv3_block1
conv3_block11_concat (Concatena (None, 28, 28, 480) 0 0_concat[0][0]			conv3_block1
conv3_block11_2_conv[0][0]			conv3_block1

conv3_block12_0_bn (BatchNormal (None, 28, 28, 480) 1920 1_concat[0][0]		conv3_block1
conv3_block12_0_relu (Activatio (None, 28, 28, 480) 0 2_bn[0][0]		conv3_block1
conv3_block12_1_conv (Conv2D) (None, 28, 28, 128) 61440 2_0_relu[0][0]		conv3_block1
conv3_block12_1_bn (BatchNormal (None, 28, 28, 128) 512 2_1_conv[0][0]		conv3_block1
conv3_block12_1_relu (Activatio (None, 28, 28, 128) 0 2_1_bn[0][0]		conv3_block1
conv3_block12_2_conv (Conv2D) (None, 28, 28, 32) 36864 2_1_relu[0][0]		conv3_block1
conv3_block12_concat (Concatena (None, 28, 28, 512) 0 1_concat[0][0]		conv3_block1
		conv3_block1
2_2_conv[0][0]		
pool3_bn (BatchNormalization) (None, 28, 28, 512) 2048 2_concat[0][0]		conv3_block1
pool3_relu (Activation) [0]	(None, 28, 28, 512) 0	pool3_bn[0]
pool3_conv (Conv2D) [0][0]	(None, 28, 28, 256) 131072	pool3_relu
pool3_pool (AveragePooling2D) [0][0]	(None, 14, 14, 256) 0	pool3_conv
conv4_block1_0_bn (BatchNormali (None, 14, 14, 256) 1024 [0][0]		pool3_pool
conv4_block1_0_relu (Activation (None, 14, 14, 256) 0 _0_bn[0][0]		conv4_block1
conv4_block1_1_conv (Conv2D) (None, 14, 14, 128) 32768 _0_relu[0][0]		conv4_block1

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_block3
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_block4
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 _0_bn[0][0]			conv4_block5
conv4_block5_1_conv (Conv2D) (None, 14, 14, 128) 49152 _0_relu[0][0]			conv4_block5
conv4_block5_1_bn (BatchNormali (None, 14, 14, 128) 512 _1_conv[0][0]			conv4_block5
conv4_block5_1_relu (Activation (None, 14, 14, 128) 0 _1_bn[0][0]			conv4_block5
conv4_block5_2_conv (Conv2D) (None, 14, 14, 32) 36864 _1_relu[0][0]			conv4_block5
conv4_block5_concat (Concatenat (None, 14, 14, 416) 0 _concat[0][0]			conv4_block4
			conv4_block5
_2_conv[0][0]			
conv4_block6_0_bn (BatchNormali (None, 14, 14, 416) 1664 _concat[0][0]			conv4_block5
conv4_block6_0_relu (Activation (None, 14, 14, 416) 0 _0_bn[0][0]			conv4_block6
conv4_block6_1_conv (Conv2D) (None, 14, 14, 128) 53248 _0_relu[0][0]			conv4_block6
conv4_block6_1_bn (BatchNormali (None, 14, 14, 128) 512 _1_conv[0][0]			conv4_block6
conv4_block6_1_relu (Activation (None, 14, 14, 128) 0 _1_bn[0][0]			conv4_block6
conv4_block6_2_conv (Conv2D) (None, 14, 14, 32) 36864 _1_relu[0][0]			conv4_block6
conv4_block6_concat (Concatenat (None, 14, 14, 448) 0 _concat[0][0]			conv4_block5
			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\_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 _concat[0][0]		conv4_block8
conv4_block9_0_relu (Activation (None, 14, 14, 512) 0 _bn[0][0]		conv4_block9
conv4_block9_1_conv (Conv2D) (None, 14, 14, 128) 65536 _relu[0][0]		conv4_block9
conv4_block9_1_bn (BatchNormali (None, 14, 14, 128) 512 _1_conv[0][0]		conv4_block9
conv4_block9_1_relu (Activation (None, 14, 14, 128) 0 _1_bn[0][0]		conv4_block9
conv4_block9_2_conv (Conv2D) (None, 14, 14, 32) 36864 _1_relu[0][0]		conv4_block9
conv4_block9_concat (Concatenat (None, 14, 14, 544) 0 _concat[0][0]		conv4_block8
		conv4_block9
conv4_block10_0_bn (BatchNormal (None, 14, 14, 544) 2176 _concat[0][0]		conv4_block9
conv4_block10_0_relu (Activatio (None, 14, 14, 544) 0 _0_bn[0][0]		conv4_block1
conv4_block10_1_conv (Conv2D) (None, 14, 14, 128) 69632 _0_0_relu[0][0]		conv4_block1
conv4_block10_1_bn (BatchNormal (None, 14, 14, 128) 512 _0_1_conv[0][0]		conv4_block1
conv4_block10_1_relu (Activatio (None, 14, 14, 128) 0 _0_1_bn[0][0]		conv4_block1
conv4_block10_2_conv (Conv2D) (None, 14, 14, 32) 36864 _0_1_relu[0][0]		conv4_block1
conv4_block10_concat (Concatena (None, 14, 14, 576) 0		conv4_block9

_concat[0][0]			conv4_block1
0_2_conv[0][0]			
conv4_block11_0_bn (BatchNormal (None, 14, 14, 576) 2304			conv4_block1
0_concat[0][0]			
conv4_block11_0_relu (Activatio (None, 14, 14, 576) 0			conv4_block1
1_0_bn[0][0]			
conv4_block11_1_conv (Conv2D) (None, 14, 14, 128) 73728			conv4_block1
1_0_relu[0][0]			
conv4_block11_1_bn (BatchNormal (None, 14, 14, 128) 512			conv4_block1
1_1_conv[0][0]			
conv4_block11_1_relu (Activatio (None, 14, 14, 128) 0			conv4_block1
1_1_bn[0][0]			
conv4_block11_2_conv (Conv2D) (None, 14, 14, 32) 36864			conv4_block1
1_1_relu[0][0]			
conv4_block11_concat (Concatena (None, 14, 14, 608) 0			conv4_block1
0_concat[0][0]			
conv4_block11_2_conv[0][0]			conv4_block1
conv4_block12_0_bn (BatchNormal (None, 14, 14, 608) 2432			conv4_block1
1_concat[0][0]			
conv4_block12_0_relu (Activatio (None, 14, 14, 608) 0			conv4_block1
2_0_bn[0][0]			
conv4_block12_1_conv (Conv2D) (None, 14, 14, 128) 77824			conv4_block1
2_0_relu[0][0]			
conv4_block12_1_bn (BatchNormal (None, 14, 14, 128) 512			conv4_block1
2_1_conv[0][0]			
conv4_block12_1_relu (Activatio (None, 14, 14, 128) 0			conv4_block1
2_1_bn[0][0]			
conv4_block12_2_conv (Conv2D) (None, 14, 14, 32) 36864			conv4_block1
2_1_relu[0][0]			

conv4_block12_concat (Concatenation (None, 14, 14, 640) 0 1_concat[0][0]		conv4_block1
2_2_conv[0][0]		conv4_block1
conv4_block13_0_bn (BatchNormal (None, 14, 14, 640) 2560 2_concat[0][0]		conv4_block1
conv4_block13_0_relu (Activation (None, 14, 14, 640) 0 3_0_bn[0][0]		conv4_block1
conv4_block13_1_conv (Conv2D) (None, 14, 14, 128) 81920 3_0_relu[0][0]		conv4_block1
conv4_block13_1_bn (BatchNormal (None, 14, 14, 128) 512 3_1_conv[0][0]		conv4_block1
conv4_block13_1_relu (Activation (None, 14, 14, 128) 0 3_1_bn[0][0]		conv4_block1
conv4_block13_2_conv (Conv2D) (None, 14, 14, 32) 36864 3_1_relu[0][0]		conv4_block1
conv4_block13_concat (Concatenation (None, 14, 14, 672) 0 2_concat[0][0]		conv4_block1
3_2_conv[0][0]		conv4_block1
conv4_block14_0_bn (BatchNormal (None, 14, 14, 672) 2688 3_concat[0][0]		conv4_block1
conv4_block14_0_relu (Activation (None, 14, 14, 672) 0 4_0_bn[0][0]		conv4_block1
conv4_block14_1_conv (Conv2D) (None, 14, 14, 128) 86016 4_0_relu[0][0]		conv4_block1
conv4_block14_1_bn (BatchNormal (None, 14, 14, 128) 512 4_1_conv[0][0]		conv4_block1
conv4_block14_1_relu (Activation (None, 14, 14, 128) 0 4_1_bn[0][0]		conv4_block1

conv4_block14_2_conv (Conv2D) (None, 14, 14, 32) 36864	conv4_block14_1_relu[0][0]	
conv4_block14_concat (Concatenation) (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_bn[0][0]
conv4_block15_0_relu (Activation) (None, 14, 14, 704) 0	conv4_block15_0_bn[0][0]	conv4_block15_0_relu[0][0]
conv4_block15_1_conv (Conv2D) (None, 14, 14, 128) 90112	conv4_block15_0_relu[0][0]	conv4_block15_1_conv[0][0]
conv4_block15_1_bn (BatchNormal) (None, 14, 14, 128) 512	conv4_block15_1_conv[0][0]	conv4_block15_1_bn[0][0]
conv4_block15_1_relu (Activation) (None, 14, 14, 128) 0	conv4_block15_1_bn[0][0]	conv4_block15_1_relu[0][0]
conv4_block15_2_conv (Conv2D) (None, 14, 14, 32) 36864	conv4_block15_1_relu[0][0]	conv4_block15_2_conv[0][0]
conv4_block15_concat (Concatenation) (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_2_conv[0][0]	conv4_block15_0_bn[0][0]
conv4_block16_0_relu (Activation) (None, 14, 14, 736) 0	conv4_block15_0_bn[0][0]	conv4_block16_0_relu[0][0]
conv4_block16_1_conv (Conv2D) (None, 14, 14, 128) 94208	conv4_block16_0_relu[0][0]	conv4_block16_1_conv[0][0]
conv4_block16_1_bn (BatchNormal) (None, 14, 14, 128) 512	conv4_block16_1_conv[0][0]	conv4_block16_1_bn[0][0]

conv4_block16_1_relu (Activation) (None, 14, 14, 128) 0		conv4_block1
6_1_bn[0][0]		
conv4_block16_2_conv (Conv2D) (None, 14, 14, 32) 36864		conv4_block1
6_1_relu[0][0]		
conv4_block16_concat (Concatenation) (None, 14, 14, 768) 0		conv4_block1
5_concat[0][0]		
conv4_block16_2_conv[0][0]		conv4_block1
conv4_block17_0_bn (BatchNormal) (None, 14, 14, 768) 3072		conv4_block1
6_concat[0][0]		
conv4_block17_0_relu (Activation) (None, 14, 14, 768) 0		conv4_block1
7_0_bn[0][0]		
conv4_block17_1_conv (Conv2D) (None, 14, 14, 128) 98304		conv4_block1
7_0_relu[0][0]		
conv4_block17_1_bn (BatchNormal) (None, 14, 14, 128) 512		conv4_block1
7_1_conv[0][0]		
conv4_block17_1_relu (Activation) (None, 14, 14, 128) 0		conv4_block1
7_1_bn[0][0]		
conv4_block17_2_conv (Conv2D) (None, 14, 14, 32) 36864		conv4_block1
7_1_relu[0][0]		
conv4_block17_concat (Concatenation) (None, 14, 14, 800) 0		conv4_block1
6_concat[0][0]		
conv4_block17_2_conv[0][0]		conv4_block1
conv4_block18_0_bn (BatchNormal) (None, 14, 14, 800) 3200		conv4_block1
7_concat[0][0]		
conv4_block18_0_relu (Activation) (None, 14, 14, 800) 0		conv4_block1
8_0_bn[0][0]		
conv4_block18_1_conv (Conv2D) (None, 14, 14, 128) 102400		conv4_block1
8_0_relu[0][0]		
conv4_block18_1_bn (BatchNormal) (None, 14, 14, 128) 512		conv4_block1
conv4_block18_1_bn[0][0]		

8\_1\_conv[0][0]

conv4_block18_1_relu (Activatio (None, 14, 14, 128) 0	conv4_block1
8_1_bn[0][0]	

conv4_block18_2_conv (Conv2D) (None, 14, 14, 32) 36864	conv4_block1
8_1_relu[0][0]	

conv4_block18_concat (Concatena (None, 14, 14, 832) 0	conv4_block1
7_concat[0][0]	
conv4_block1	
8_2_conv[0][0]	

conv4_block19_0_bn (BatchNormal (None, 14, 14, 832) 3328	conv4_block1
8_concat[0][0]	

conv4_block19_0_relu (Activatio (None, 14, 14, 832) 0	conv4_block1
9_0_bn[0][0]	

conv4_block19_1_conv (Conv2D) (None, 14, 14, 128) 106496	conv4_block1
9_0_relu[0][0]	

conv4_block19_1_bn (BatchNormal (None, 14, 14, 128) 512	conv4_block1
9_1_conv[0][0]	

conv4_block19_1_relu (Activatio (None, 14, 14, 128) 0	conv4_block1
9_1_bn[0][0]	

conv4_block19_2_conv (Conv2D) (None, 14, 14, 32) 36864	conv4_block1
9_1_relu[0][0]	

conv4_block19_concat (Concatena (None, 14, 14, 864) 0	conv4_block1
8_concat[0][0]	
conv4_block1	
9_2_conv[0][0]	

conv4_block20_0_bn (BatchNormal (None, 14, 14, 864) 3456	conv4_block1
9_concat[0][0]	

conv4_block20_0_relu (Activatio (None, 14, 14, 864) 0	conv4_block2
0_0_bn[0][0]	

conv4_block20_1_conv (Conv2D) (None, 14, 14, 128) 110592	conv4_block2
0_0_relu[0][0]	

conv4_block20_1_bn (BatchNormal (None, 14, 14, 128) 512 0_1_conv[0][0]		conv4_block2
conv4_block20_1_relu (Activatio (None, 14, 14, 128) 0 0_1_bn[0][0]		conv4_block2
conv4_block20_2_conv (Conv2D) (None, 14, 14, 32) 36864 0_1_relu[0][0]		conv4_block2
conv4_block20_concat (Concatena (None, 14, 14, 896) 0 9_concat[0][0]		conv4_block1
		conv4_block2
0_2_conv[0][0]		
conv4_block21_0_bn (BatchNormal (None, 14, 14, 896) 3584 0_concat[0][0]		conv4_block2
conv4_block21_0_relu (Activatio (None, 14, 14, 896) 0 1_0_bn[0][0]		conv4_block2
conv4_block21_1_conv (Conv2D) (None, 14, 14, 128) 114688 1_0_relu[0][0]		conv4_block2
conv4_block21_1_bn (BatchNormal (None, 14, 14, 128) 512 1_1_conv[0][0]		conv4_block2
conv4_block21_1_relu (Activatio (None, 14, 14, 128) 0 1_1_bn[0][0]		conv4_block2
conv4_block21_2_conv (Conv2D) (None, 14, 14, 32) 36864 1_1_relu[0][0]		conv4_block2
conv4_block21_concat (Concatena (None, 14, 14, 928) 0 0_concat[0][0]		conv4_block2
		conv4_block2
1_2_conv[0][0]		
conv4_block22_0_bn (BatchNormal (None, 14, 14, 928) 3712 1_concat[0][0]		conv4_block2
conv4_block22_0_relu (Activatio (None, 14, 14, 928) 0 2_0_bn[0][0]		conv4_block2

conv4_block22_1_conv (Conv2D) (None, 14, 14, 128) 118784	conv4_block2
2_0_relu[0][0]	
conv4_block22_1_bn (BatchNormal (None, 14, 14, 128) 512	conv4_block2
2_1_conv[0][0]	
conv4_block22_1_relu (Activatio (None, 14, 14, 128) 0	conv4_block2
2_1_bn[0][0]	
conv4_block22_2_conv (Conv2D) (None, 14, 14, 32) 36864	conv4_block2
2_1_relu[0][0]	
conv4_block22_concat (Concatena (None, 14, 14, 960) 0	conv4_block2
1_concat[0][0]	
conv4_block22_2_conv[0][0]	
conv4_block23_0_bn (BatchNormal (None, 14, 14, 960) 3840	conv4_block2
2_concat[0][0]	
conv4_block23_0_relu (Activatio (None, 14, 14, 960) 0	conv4_block2
3_0_bn[0][0]	
conv4_block23_0_relu[0][0]	
conv4_block23_1_conv (Conv2D) (None, 14, 14, 128) 122880	conv4_block2
3_0_relu[0][0]	
conv4_block23_1_bn (BatchNormal (None, 14, 14, 128) 512	conv4_block2
3_1_conv[0][0]	
conv4_block23_1_relu (Activatio (None, 14, 14, 128) 0	conv4_block2
3_1_bn[0][0]	
conv4_block23_1_relu[0][0]	
conv4_block23_2_conv (Conv2D) (None, 14, 14, 32) 36864	conv4_block2
3_1_relu[0][0]	
conv4_block23_concat (Concatena (None, 14, 14, 992) 0	conv4_block2
2_concat[0][0]	
conv4_block23_2_conv[0][0]	
conv4_block23_3_conv (Conv2D) (None, 14, 14, 992) 3968	conv4_block2
3_concat[0][0]	
conv4_block23_3_conv[0][0]	

conv4_block24_0_relu (Activation) (None, 14, 14, 992) 0		conv4_block2
4_0_bn[0][0]		
conv4_block24_1_conv (Conv2D) (None, 14, 14, 128) 126976		conv4_block2
4_0_relu[0][0]		
conv4_block24_1_bn (BatchNormal (None, 14, 14, 128) 512		conv4_block2
4_1_conv[0][0]		
conv4_block24_1_relu (Activatio (None, 14, 14, 128) 0		conv4_block2
4_1_bn[0][0]		
conv4_block24_2_conv (Conv2D) (None, 14, 14, 32) 36864		conv4_block2
4_1_relu[0][0]		
conv4_block24_concat (Concatena (None, 14, 14, 1024) 0		conv4_block2
3_concat[0][0]		
conv4_block24_2_conv[0][0]		
pool4_bn (BatchNormalization) (None, 14, 14, 1024) 4096		conv4_block2
4_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_block1
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_block2
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) _1_conv[0][0]	512	conv5_block3
conv5_block3_1_relu (Activation (None, 7, 7, 128) _1_bn[0][0]	0	conv5_block3
conv5_block3_2_conv (Conv2D) (None, 7, 7, 32) _1_relu[0][0]	36864	conv5_block3
conv5_block3_concat (Concatenat (None, 7, 7, 608) _concat[0][0]	0	conv5_block2
conv5_block3_2_conv[0][0]		conv5_block3
conv5_block4_0_bn (BatchNormali (None, 7, 7, 608) _concat[0][0]	2432	conv5_block3
conv5_block4_0_relu (Activation (None, 7, 7, 608) _0_bn[0][0]	0	conv5_block4
conv5_block4_1_conv (Conv2D) (None, 7, 7, 128) _0_relu[0][0]	77824	conv5_block4
conv5_block4_1_bn (BatchNormali (None, 7, 7, 128) _1_conv[0][0]	512	conv5_block4
conv5_block4_1_relu (Activation (None, 7, 7, 128) _1_bn[0][0]	0	conv5_block4
conv5_block4_2_conv (Conv2D) (None, 7, 7, 32) _1_relu[0][0]	36864	conv5_block4
conv5_block4_concat (Concatenat (None, 7, 7, 640) _concat[0][0]	0	conv5_block3
conv5_block4_2_conv[0][0]		conv5_block4
conv5_block5_0_bn (BatchNormali (None, 7, 7, 640) _concat[0][0]	2560	conv5_block4
conv5_block5_0_relu (Activation (None, 7, 7, 640) _0_bn[0][0]	0	conv5_block5
conv5_block5_1_conv (Conv2D) (None, 7, 7, 128) _1_conv[0][0]	81920	conv5_block5

<u>_0_relu[0][0]</u>			
<u>conv5_block5_1_bn</u> (BatchNormali (None, 7, 7, 128))	512		conv5_block5_1_conv[0][0]
<u>conv5_block5_1_relu</u> (Activation (None, 7, 7, 128))	0		conv5_block5_1_bn[0][0]
<u>conv5_block5_2_conv</u> (Conv2D) (None, 7, 7, 32)	36864		conv5_block5_1_relu[0][0]
<u>conv5_block5_concat</u> (Concatenat (None, 7, 7, 672))	0		conv5_block4_concat[0][0]
			conv5_block5_2_conv[0][0]
<u>conv5_block6_0_bn</u> (BatchNormali (None, 7, 7, 672))	2688		conv5_block5_concat[0][0]
<u>conv5_block6_0_relu</u> (Activation (None, 7, 7, 672))	0		conv5_block6_0_bn[0][0]
<u>conv5_block6_1_conv</u> (Conv2D) (None, 7, 7, 128)	86016		conv5_block6_0_relu[0][0]
<u>conv5_block6_1_bn</u> (BatchNormali (None, 7, 7, 128))	512		conv5_block6_1_conv[0][0]
<u>conv5_block6_1_relu</u> (Activation (None, 7, 7, 128))	0		conv5_block6_1_bn[0][0]
<u>conv5_block6_2_conv</u> (Conv2D) (None, 7, 7, 32)	36864		conv5_block6_1_relu[0][0]
<u>conv5_block6_concat</u> (Concatenat (None, 7, 7, 704))	0		conv5_block5_concat[0][0]
			conv5_block6_2_conv[0][0]
<u>conv5_block7_0_bn</u> (BatchNormali (None, 7, 7, 704))	2816		conv5_block6_concat[0][0]
<u>conv5_block7_0_relu</u> (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_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_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_block9
conv5_block10_0_bn (BatchNormal (None, 7, 7, 800) 3200		conv5_block9
_concat[0][0]		
conv5_block10_0_relu (Activatio (None, 7, 7, 800) 0		conv5_block1
0_0_bn[0][0]		
conv5_block10_1_conv (Conv2D) (None, 7, 7, 128) 102400		conv5_block1
0_0_relu[0][0]		
conv5_block10_1_bn (BatchNormal (None, 7, 7, 128) 512		conv5_block1
0_1_conv[0][0]		
conv5_block10_1_relu (Activatio (None, 7, 7, 128) 0		conv5_block1
0_1_bn[0][0]		
conv5_block10_2_conv (Conv2D) (None, 7, 7, 32) 36864		conv5_block1
0_1_relu[0][0]		
conv5_block10_concat (Concatena (None, 7, 7, 832) 0		conv5_block9
_concat[0][0]		
conv5_block10_2_conv[0][0]		conv5_block1
conv5_block10_2_conv[0][0]		

conv5_block11_0_bn (BatchNormal (None, 7, 7, 832)	3328	conv5_block1
0_concat[0][0]		
conv5_block11_0_relu (Activatio (None, 7, 7, 832)	0	conv5_block1
1_0_bn[0][0]		
conv5_block11_1_conv (Conv2D) (None, 7, 7, 128)	106496	conv5_block1
1_0_relu[0][0]		
conv5_block11_1_bn (BatchNormal (None, 7, 7, 128)	512	conv5_block1
1_1_conv[0][0]		
conv5_block11_1_relu (Activatio (None, 7, 7, 128)	0	conv5_block1
1_1_bn[0][0]		
conv5_block11_2_conv (Conv2D) (None, 7, 7, 32)	36864	conv5_block1
1_1_relu[0][0]		
conv5_block11_concat (Concatena (None, 7, 7, 864)	0	conv5_block1
0_concat[0][0]		
conv5_block11_2_conv[0][0]		conv5_block1
conv5_block12_0_bn (BatchNormal (None, 7, 7, 864)	3456	conv5_block1
1_concat[0][0]		
conv5_block12_0_relu (Activatio (None, 7, 7, 864)	0	conv5_block1
2_0_bn[0][0]		
conv5_block12_1_conv (Conv2D) (None, 7, 7, 128)	110592	conv5_block1
2_0_relu[0][0]		
conv5_block12_1_bn (BatchNormal (None, 7, 7, 128)	512	conv5_block1
2_1_conv[0][0]		
conv5_block12_1_relu (Activatio (None, 7, 7, 128)	0	conv5_block1
2_1_bn[0][0]		
conv5_block12_2_conv (Conv2D) (None, 7, 7, 32)	36864	conv5_block1
2_1_relu[0][0]		
conv5_block12_concat (Concatena (None, 7, 7, 896)	0	conv5_block1
1_concat[0][0]		
conv5_block12_2_conv[0][0]		conv5_block1

2\_2\_conv[0][0]

---

conv5\_block13\_0\_bn (BatchNormal (None, 7, 7, 896) 3584 conv5\_block1  
2\_concat[0][0]

---

conv5\_block13\_0\_relu (Activatio (None, 7, 7, 896) 0 conv5\_block1  
3\_0\_bn[0][0]

---

conv5\_block13\_1\_conv (Conv2D) (None, 7, 7, 128) 114688 conv5\_block1  
3\_0\_relu[0][0]

---

conv5\_block13\_1\_bn (BatchNormal (None, 7, 7, 128) 512 conv5\_block1  
3\_1\_conv[0][0]

---

conv5\_block13\_1\_relu (Activatio (None, 7, 7, 128) 0 conv5\_block1  
3\_1\_bn[0][0]

---

conv5\_block13\_2\_conv (Conv2D) (None, 7, 7, 32) 36864 conv5\_block1  
3\_1\_relu[0][0]

---

conv5\_block13\_concat (Concatena (None, 7, 7, 928) 0 conv5\_block1  
2\_concat[0][0]  
conv5\_block1  
3\_2\_conv[0][0]

---

conv5\_block14\_0\_bn (BatchNormal (None, 7, 7, 928) 3712 conv5\_block1  
3\_concat[0][0]

---

conv5\_block14\_0\_relu (Activatio (None, 7, 7, 928) 0 conv5\_block1  
4\_0\_bn[0][0]

---

conv5\_block14\_1\_conv (Conv2D) (None, 7, 7, 128) 118784 conv5\_block1  
4\_0\_relu[0][0]

---

conv5\_block14\_1\_bn (BatchNormal (None, 7, 7, 128) 512 conv5\_block1  
4\_1\_conv[0][0]

---

conv5\_block14\_1\_relu (Activatio (None, 7, 7, 128) 0 conv5\_block1  
4\_1\_bn[0][0]

---

conv5\_block14\_2\_conv (Conv2D) (None, 7, 7, 32) 36864 conv5\_block1  
4\_1\_relu[0][0]

---

conv5_block14_concat (Concatena (None, 7, 7, 960) 0 conv5_block1 3_concat[0][0]			
conv5_block15_0_bn (BatchNormal (None, 7, 7, 960) 3840 conv5_block1 4_concat[0][0]			
conv5_block15_0_relu (Activatio (None, 7, 7, 960) 0 conv5_block1 5_0_bn[0][0]			
conv5_block15_1_conv (Conv2D) (None, 7, 7, 128) 122880 conv5_block1 5_0_relu[0][0]			
conv5_block15_1_bn (BatchNormal (None, 7, 7, 128) 512 conv5_block1 5_1_conv[0][0]			
conv5_block15_1_relu (Activatio (None, 7, 7, 128) 0 conv5_block1 5_1_bn[0][0]			
conv5_block15_2_conv (Conv2D) (None, 7, 7, 32) 36864 conv5_block1 5_1_relu[0][0]			
conv5_block15_concat (Concatena (None, 7, 7, 992) 0 conv5_block1 4_concat[0][0]			
conv5_block15_2_conv[0][0] conv5_block15_2_conv (Conv2D) (None, 7, 7, 32) 36864 conv5_block1 5_2_conv[0][0]			
conv5_block16_0_bn (BatchNormal (None, 7, 7, 992) 3968 conv5_block1 5_concat[0][0]			
conv5_block16_0_relu (Activatio (None, 7, 7, 992) 0 conv5_block1 6_0_bn[0][0]			
conv5_block16_1_conv (Conv2D) (None, 7, 7, 128) 126976 conv5_block1 6_0_relu[0][0]			
conv5_block16_1_bn (BatchNormal (None, 7, 7, 128) 512 conv5_block1 6_1_conv[0][0]			
conv5_block16_1_relu (Activatio (None, 7, 7, 128) 0 conv5_block1 6_1_bn[0][0]			
conv5_block16_2_conv (Conv2D) (None, 7, 7, 32) 36864 conv5_block1 6_1_conv[0][0]			

6\_1\_relu[0][0]

---

conv5\_block16\_concat (Concatenation) (None, 7, 7, 1024) 0 conv5\_block1  
5\_concat[0][0] conv5\_block1  
6\_2\_conv[0][0]

---

bn (BatchNormalization) (None, 7, 7, 1024) 4096 conv5\_block1  
6\_concat[0][0]

---

relu (Activation) (None, 7, 7, 1024) 0 bn[0][0]

---

max\_pool (GlobalMaxPooling2D) (None, 1024) 0 relu[0][0]

---

dropout\_1 (Dropout) (None, 1024) 0 max\_pool[0]  
[0]

---

dense\_1 (Dense) (None, 4) 4100 dropout\_1[0]  
[0]

---

=====

Total params: 7,041,604  
Trainable params: 6,948,548  
Non-trainable params: 93,056

---

None

---



```
In [23]: model, history = train_model(model, batch_size = 32, epochs = 20, model_no = 4  
)
```

WARNING:tensorflow:From /usr/local/lib/python3.5/dist-packages/keras/backend/tensorflow\_backend.py:422: The name tf.global\_variables is deprecated. Please use tf.compat.v1.global\_variables instead.

WARNING:tensorflow:From /usr/local/lib/python3.5/dist-packages/keras/callbacks/tensorboard\_v1.py:200: The name tf.summary.merge\_all is deprecated. Please use tf.compat.v1.summary.merge\_all instead.

WARNING:tensorflow:From /usr/local/lib/python3.5/dist-packages/keras/callbacks/tensorboard\_v1.py:203: The name tf.summary.FileWriter is deprecated. Please use tf.compat.v1.summary.FileWriter instead.

Epoch 1/20

1355/1355 [=====] - 1919s 1s/step - loss: 0.5356 - f1\_micro\_score: 0.8684 - val\_loss: 10960.7422 - val\_f1\_micro\_score: 0.4487

Epoch 00001: val\_loss improved from inf to 10960.74219, saving model to model\_5.hdf5

WARNING:tensorflow:From /usr/local/lib/python3.5/dist-packages/keras/callbacks/tensorboard\_v1.py:343: The name tf.Summary is deprecated. Please use tf.compat.v1.Summary instead.

Epoch 2/20

1355/1355 [=====] - 1245s 919ms/step - loss: 0.6177 - f1\_micro\_score: 0.8029 - val\_loss: 0.0105 - val\_f1\_micro\_score: 0.8943

Epoch 00002: val\_loss improved from 10960.74219 to 0.01049, saving model to model\_5.hdf5

Epoch 3/20

1355/1355 [=====] - 1245s 919ms/step - loss: 0.3681 - f1\_micro\_score: 0.8876 - val\_loss: 0.0117 - val\_f1\_micro\_score: 0.9059

Epoch 00003: val\_loss did not improve from 0.01049

Epoch 00003: ReduceLROnPlateau reducing learning rate to 0.0002000000949949026.

Epoch 4/20

1355/1355 [=====] - 1244s 918ms/step - loss: 0.1981 - f1\_micro\_score: 0.9328 - val\_loss: 0.0072 - val\_f1\_micro\_score: 0.9276

Epoch 00004: val\_loss improved from 0.01049 to 0.00721, saving model to model\_5.hdf5

Epoch 5/20

1355/1355 [=====] - 1250s 923ms/step - loss: 0.1745 - f1\_micro\_score: 0.9403 - val\_loss: 0.0056 - val\_f1\_micro\_score: 0.9288

Epoch 00005: val\_loss improved from 0.00721 to 0.00562, saving model to model\_5.hdf5

Epoch 6/20

1355/1355 [=====] - 1253s 925ms/step - loss: 0.1566 - f1\_micro\_score: 0.9470 - val\_loss: 0.0032 - val\_f1\_micro\_score: 0.9350

Epoch 00006: val\_loss improved from 0.00562 to 0.00316, saving model to model\_5.hdf5

Epoch 7/20

1355/1355 [=====] - 1251s 923ms/step - loss: 0.1401 - f1\_micro\_score: 0.9523 - val\_loss: 0.0014 - val\_f1\_micro\_score: 0.9398

```
Epoch 00007: val_loss improved from 0.00316 to 0.00144, saving model to model
_5.hdf5
Epoch 8/20
1355/1355 [=====] - 1249s 922ms/step - loss: 0.1235
- f1_micro_score: 0.9569 - val_loss: 2.4512e-04 - val_f1_micro_score: 0.9380

Epoch 00008: val_loss improved from 0.00144 to 0.00025, saving model to model
_5.hdf5
Epoch 9/20
1355/1355 [=====] - 1253s 925ms/step - loss: 0.1085
- f1_micro_score: 0.9638 - val_loss: 5.4775e-05 - val_f1_micro_score: 0.9419

Epoch 00009: val_loss improved from 0.00025 to 0.00005, saving model to model
_5.hdf5
Epoch 10/20
1355/1355 [=====] - 1256s 927ms/step - loss: 0.0974
- f1_micro_score: 0.9658 - val_loss: 3.7014e-05 - val_f1_micro_score: 0.9428

Epoch 00010: val_loss improved from 0.00005 to 0.00004, saving model to model
_5.hdf5
Epoch 11/20
1355/1355 [=====] - 1253s 924ms/step - loss: 0.0826
- f1_micro_score: 0.9717 - val_loss: 1.8656e-05 - val_f1_micro_score: 0.9398

Epoch 00011: val_loss improved from 0.00004 to 0.00002, saving model to model
_5.hdf5
Epoch 12/20
1355/1355 [=====] - 1248s 921ms/step - loss: 0.0702
- f1_micro_score: 0.9756 - val_loss: 1.5795e-05 - val_f1_micro_score: 0.9426

Epoch 00012: val_loss improved from 0.00002 to 0.00002, saving model to model
_5.hdf5

Epoch 00012: ReduceLROnPlateau reducing learning rate to 4.0000001899898055e-
05.
Epoch 13/20
1355/1355 [=====] - 1248s 921ms/step - loss: 0.0485
- f1_micro_score: 0.9832 - val_loss: 0.0000e+00 - val_f1_micro_score: 0.9563

Epoch 00013: val_loss improved from 0.00002 to 0.00000, saving model to model
_5.hdf5
Epoch 14/20
1355/1355 [=====] - 1251s 923ms/step - loss: 0.0348
- f1_micro_score: 0.9884 - val_loss: 0.0000e+00 - val_f1_micro_score: 0.9573

Epoch 00014: val_loss did not improve from 0.00000

Epoch 00014: ReduceLROnPlateau reducing learning rate to 8.000000525498762e-0
6.
Epoch 15/20
1355/1355 [=====] - 1252s 924ms/step - loss: 0.0251
- f1_micro_score: 0.9914 - val_loss: 0.0000e+00 - val_f1_micro_score: 0.9579

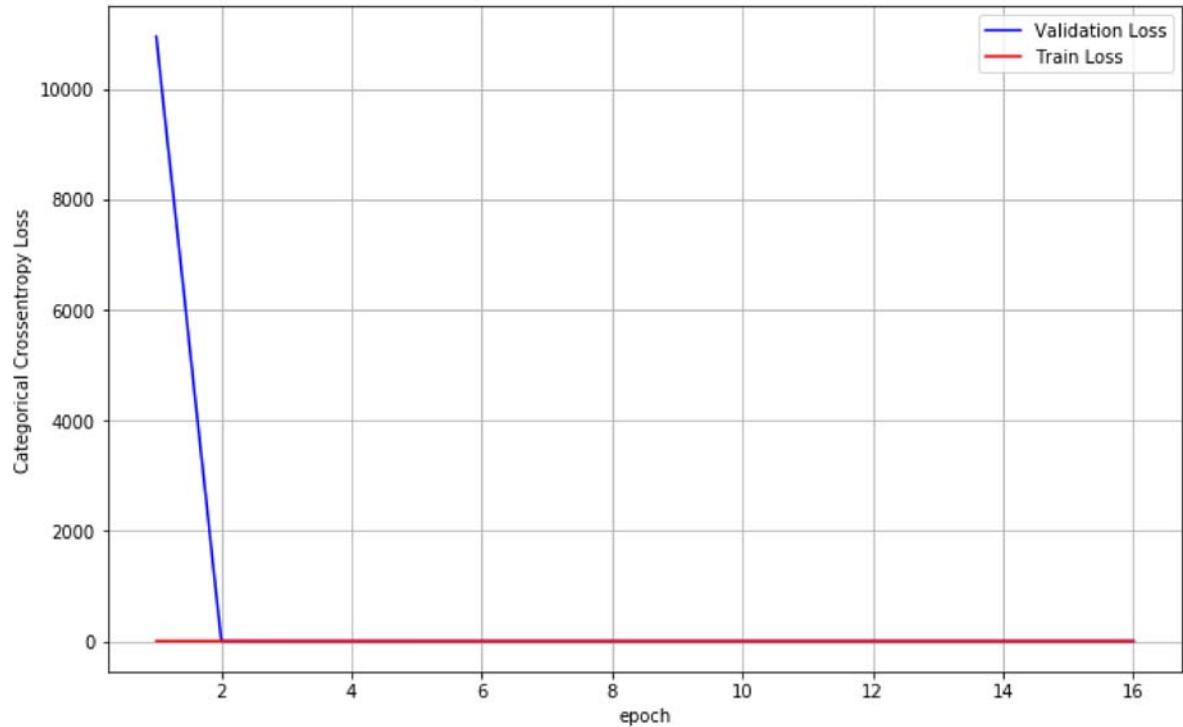
Epoch 00015: val_loss did not improve from 0.00000

Epoch 00015: ReduceLROnPlateau reducing learning rate to 1.6000001778593287e-
```

```
06.  
Epoch 16/20  
1355/1355 [=====] - 1254s 926ms/step - loss: 0.0210  
- f1_micro_score: 0.9930 - val_loss: 0.0000e+00 - val_f1_micro_score: 0.9577  
  
Epoch 00016: val_loss did not improve from 0.00000  
Restoring model weights from the end of the best epoch  
  
Epoch 00016: ReduceLROnPlateau reducing learning rate to 3.200000264769187e-0  
7.  
Epoch 00016: early stopping
```

```
In [86]: model = load_model("model_5.hdf5", custom_objects={"f1_micro_score": f1_micro_score})
```

```
In [24]: nb_epoch = len(history.history['loss'])  
%matplotlib inline  
import matplotlib.pyplot as plt  
plt.rcParams["figure.figsize"] = [11, 7]  
  
fig,ax = plt.subplots(1,1)  
ax.set_xlabel('epoch') ; ax.set_ylabel('Categorical Crossentropy Loss')  
  
# List of epoch numbers  
x = list(range(1,nb_epoch+1))  
  
vy = history.history['val_loss']  
ty = history.history['loss']  
plt_dynamic(x, vy, ty, ax, "Loss")
```

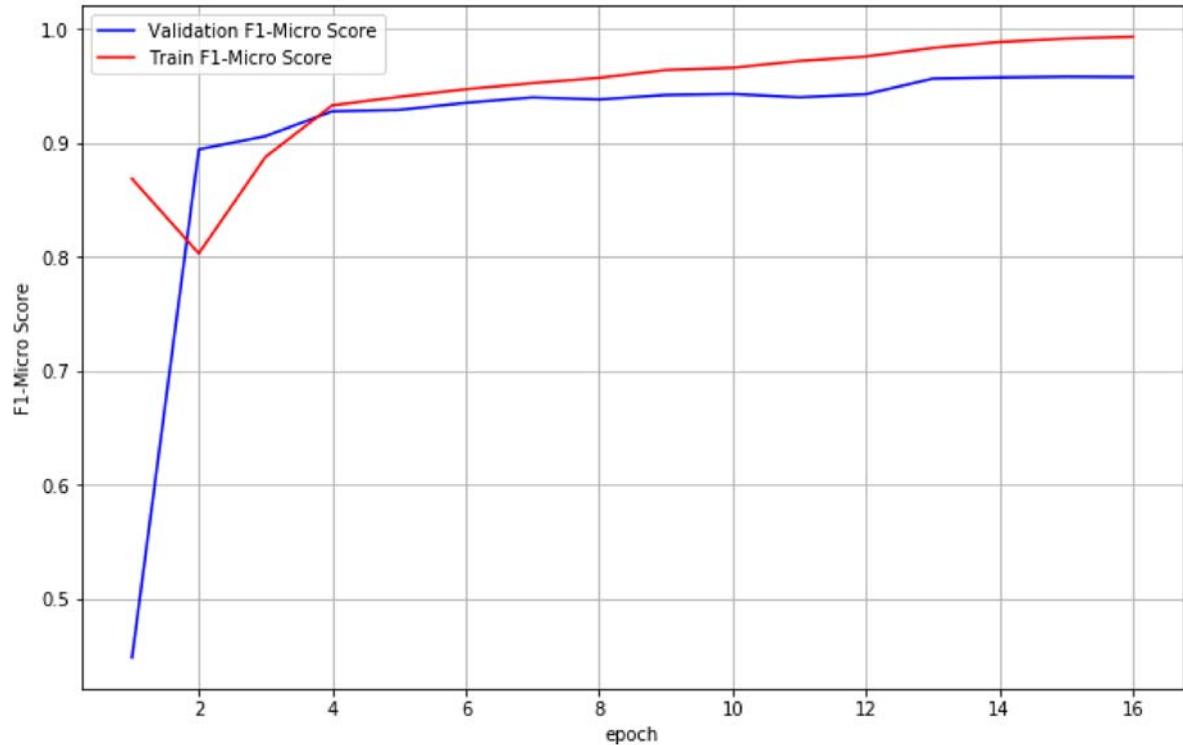


```
In [25]: nb_epoch = len(history.history['f1_micro_score'])
%matplotlib inline
import matplotlib.pyplot as plt
plt.rcParams["figure.figsize"] = [11, 7]
```

```
fig,ax = plt.subplots(1,1)
ax.set_xlabel('epoch') ; ax.set_ylabel('F1-Micro Score')

# List of epoch numbers
x = list(range(1,nb_epoch+1))

vy = history.history['val_f1_micro_score']
ty = history.history['f1_micro_score']
plt_dynamic(x, vy, ty, ax, "F1-Micro Score")
```



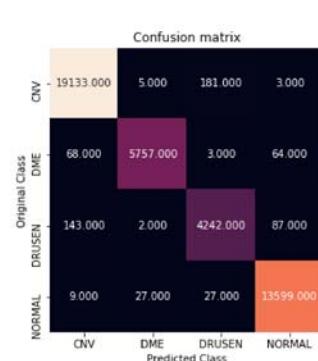
```
In [26]: metric = report_of_model(model, model_name = "DenseNet121 Transfer Learning",  
batch_size = 64)
```

```

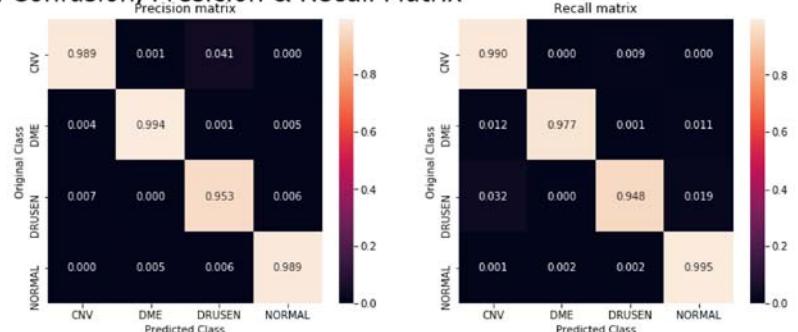
678/678 [=====] - 358s 528ms/step
120/120 [=====] - 65s 538ms/step
141/141 [=====] - 333s 2s/step
=====
Train log-loss : 0.009
Validation log-loss : 0.092
Test log-loss : 0.427
=====
678/678 [=====] - 359s 530ms/step
120/120 [=====] - 63s 521ms/step
141/141 [=====] - 73s 521ms/step
=====
Train F1-micro avg score : 0.9857
Val F1-micro avg score : 0.9561
Test F1-micro avg score : 0.9651
=====
Train Classification Report
      precision    recall   f1-score   support
CNV        0.99     0.99     0.99     19322
DME        0.99     0.98     0.99     5892
DRUSEN     0.95     0.95     0.95     4474
NORMAL     0.99     1.00     0.99     13662
accuracy          0.99     0.99     0.99     43350
macro avg       0.98     0.98     0.98     43350
weighted avg    0.99     0.99     0.99     43350

```

#### Train Confusion Matrix



#### Train Confusion, Precision & Recall Matrix



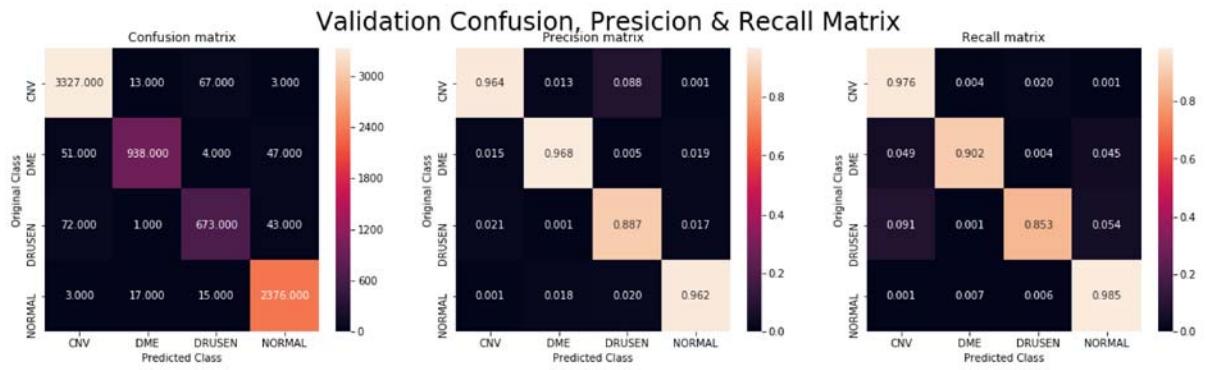
#### Validation Classification Report

```

      precision    recall   f1-score   support
CNV        0.96     0.98     0.97     3410
DME        0.97     0.90     0.93     1040
DRUSEN     0.89     0.85     0.87     789
NORMAL     0.96     0.99     0.97     2411
accuracy          0.96     0.96     0.96     7650
macro avg       0.95     0.93     0.94     7650
weighted avg    0.96     0.96     0.96     7650

```

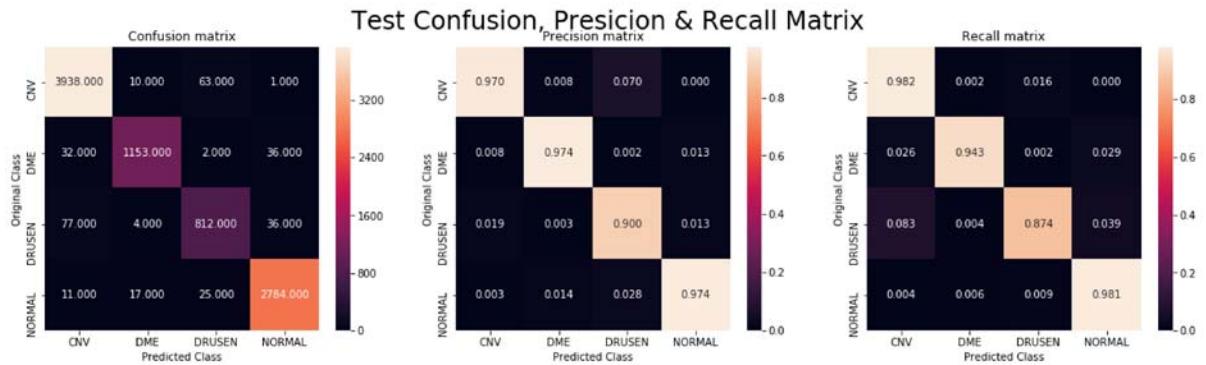
#### Validation Confusion Matrix



### Test Classification Report

	precision	recall	f1-score	support
CNV	0.97	0.98	0.98	4012
DME	0.97	0.94	0.96	1223
DRUSEN	0.90	0.87	0.89	929
NORMAL	0.97	0.98	0.98	2837
accuracy			0.97	9001
macro avg	0.95	0.94	0.95	9001
weighted avg	0.96	0.97	0.96	9001

### Test Confusion Matrix



```
In [27]: metrics.append(metric)
```

```
In [35]: pickle.dump(metrics, open("metrics.p", "wb"))
```

```
In [36]: metrics = pickle.load(open("metrics.p", "rb"))
```

## Conclusion

In [38]: `from prettytable import PrettyTable`

```
x = PrettyTable()

x.field_names = ["Model", "Train Loss", "Val Loss", "Test Loss"]
for i in metrics:
    x.add_row([i[0], round(i[1],4), round(i[3],4), round(i[5],4)])
print(x)
```

Model	Train Loss	Val Loss	Test Loss
3 Layer CNN Model	0.0929	0.4276	0.2018
7 Layer CNN Model	0.042	0.1873	0.2335
VGG16 Transfer Learning	0.0628	0.2959	0.3119
ResNet50 Transfer Learning	0.019	0.2734	0.2909
DenseNet121 Transfer Learning	0.0091	0.0921	0.4273

In [39]: `from prettytable import PrettyTable`

```
x = PrettyTable()

x.field_names = ["Model", "Train Micro-avg f1 score", "Val Micro-avg f1 score",
                 "Test Micro-avg f1 score"]
for i in metrics:
    x.add_row([i[0], round(i[7],4), round(i[8],4), round(i[9],4)])
print(x)
```

Model	Train Micro-avg f1 score	Val Micro-avg f1 score	Test Micro-avg f1 score
3 Layer CNN Model	0.9296	0.9199	0.9203
7 Layer CNN Model	0.9649	0.9542	0.9582
VGG16 Transfer Learning	0.9298	0.9071	0.9089
ResNet50 Transfer Learning	0.9459	0.9366	0.9399
DenseNet121 Transfer Learning	0.9857	0.9561	0.9651

- **DenseNet121 Transfer Learning Model performed best as it has highest micro-avg f1 score**

In [ ]: