

IRV2_on_GZ2_v6

May 3, 2022

```
[ ]: import os
import pandas as pd
import numpy as np
import tensorflow as tf
import matplotlib.pyplot as plt
import zipfile
import io
from PIL import Image
import matplotlib.pyplot as plt
from skimage.transform import resize
from tensorflow import keras
from tensorflow.keras.models import Model, load_model, Sequential
from tensorflow.keras.layers import Input, Dense, Conv2D, Flatten
from tensorflow.keras.optimizers import SGD, Adam
from keras.applications.inception_resnet_v2 import InceptionResNetV2 as PretrainedModel, preprocess_input
from tensorflow.keras.preprocessing import image
from tensorflow.keras.preprocessing.image import ImageDataGenerator, array_to_img, img_to_array, load_img
from tensorflow.keras.callbacks import ModelCheckpoint, Callback, EarlyStopping
```

```
[ ]: # zippath = '/content/drive/MyDrive/Major_Project/GZ-2/archive.zip'
# z = zipfile.ZipFile(zippath)
# imgname = 'images_gz2/images/233063.jpg'
# im = Image.open(io.BytesIO(z.read(imgname)))
# im_list = np.asarray(im)
# plt.imshow(im_list)
# plt.show()

# z.close()
```

```
[ ]: # plt.figure(figsize=(16,4))
# for i in range(3):
#     plt.subplot(1,3,i+1)
#     plt.imshow(im_list[:, :, i])
#     plt.colorbar()
# plt.show()
```

```
[ ]: # imgname = 'images_gz2/images/233063.jpg'
# img = load_img(imgname)
# data = img_to_array(img)
# samples = np.expand_dims(data, 0)
```

```
[ ]: # def visualiseAugmentation(datagen):
#     it = datagen.flow(samples, batch_size=1)
#     plt.figure(figsize=(15,15))
#     for i in range(9):
#         plt.subplot(330 + 1 + i)
#         batch = it.next()
#         image = batch[0].astype('uint8')
#         plt.imshow(image)
#     plt.show()
```

```
[ ]: # widthShift = ImageDataGenerator(width_shift_range=[-200,200])
# visualiseAugmentation(widthShift)
```

```
[ ]: # zoomRange = ImageDataGenerator(zoom_range=[0.4, 0.7])
# visualiseAugmentation(zoomRange)
```

```
[ ]: # rotation_range = ImageDataGenerator(rotation_range=90)
# visualiseAugmentation(rotation_range)
```

```
[ ]: # shear_range = ImageDataGenerator(shear_range=0.7)
# visualiseAugmentation(shear_range)
```

```
[ ]: def append_ext(fn):
    """
    This function is used to take the GalaxyID from the CSV and append .jpg to
    it in order to denote the image names.
    """
    return fn + ".jpg"

traindf = pd.read_csv('D:/OneDrive/Major Project/HybridModel_37Classes/
    ↪GZ_2_Processed_classes.csv')

traindf["id"] = traindf['GalaxyID'].astype(str).apply(append_ext)
```

```
[ ]: classes = [
    'Class1.1', 'Class1.2', 'Class1.3', 'Class2.1', 'Class2.2', 'Class3.1',
    'Class3.2', 'Class4.1', 'Class4.2', 'Class5.1', 'Class5.2', 'Class5.3',
    'Class5.4', 'Class6.1', 'Class6.2', 'Class7.1', 'Class7.2', 'Class7.3',
    'Class8.1', 'Class8.2', 'Class8.3', 'Class8.4', 'Class8.5', 'Class8.6',
    'Class8.7', 'Class9.1', 'Class9.2', 'Class9.3', 'Class10.1', 'Class10.2',
    'Class10.3', 'Class11.1', 'Class11.2', 'Class11.3', 'Class11.4',
```

```
'Class11.5', 'Class11.6']
```

```
[ ]: datagenerator = ImageDataGenerator(  
    fill_mode='nearest',  
    cval=0,  
    rescale=1/255,  
    rotation_range=25,  
    shear_range=0.2,  
    width_shift_range=[0.1, 0.15],  
    height_shift_range=[0.1, 0.15],  
    horizontal_flip=True,  
    vertical_flip=True,  
    zoom_range=[0.4, 0.7],  
    validation_split=0.025)
```

```
[ ]: train_generator = datagenerator.flow_from_dataframe(  
    dataframe=traindf,  
    directory="D:/Rahul Noronha/Shared Folder/Eighth Semester/Major Project/  
↳Data/images",  
    x_col="id",  
    y_col=classes,  
    subset="training",  
    batch_size=64,  
    seed=123,  
    shuffle=True,  
    class_mode="raw",  
    target_size=(299, 299))
```

```
validation_generator = datagenerator.flow_from_dataframe(  
    dataframe=traindf,  
    directory="D:/Rahul Noronha/Shared Folder/Eighth Semester/Major Project/  
↳Data/images",  
    x_col="id",  
    y_col=classes,  
    subset="validation",  
    batch_size=16,  
    seed=123,  
    shuffle=True,  
    class_mode="raw",  
    target_size=(299, 299))
```

```
STEP_SIZE_TRAIN = train_generator.n // train_generator.batch_size  
STEP_SIZE_VALID = validation_generator.n // validation_generator.batch_size
```

D:\anaconda\envs\python37majorproject\lib\site-packages\keras_preprocessing\image\dataframe_iterator.py:282: UserWarning: Found

```
108 invalid image filename(s) in x_col="id". These filename(s) will be ignored.
    .format(n_invalid, x_col)
```

Found 198632 validated image filenames.

Found 5093 validated image filenames.

```
[ ]: import os
import re
import sys
import time
import numpy as np
from typing import Any, List, Tuple, Union
from tensorflow.keras.datasets import mnist
from tensorflow.keras import backend as K
import tensorflow as tf
import tensorflow.keras
import tensorflow as tf
from tensorflow.keras.callbacks import EarlyStopping, \
    LearningRateScheduler, ModelCheckpoint
from tensorflow.keras import regularizers
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, Dropout, Flatten
from tensorflow.keras.layers import Conv2D, MaxPooling2D
from tensorflow.keras.models import load_model
import pickle

[ ]: def generate_output_dir(outdir, run_desc):
    prev_run_dirs = []
    if os.path.isdir(outdir):
        prev_run_dirs = [x for x in os.listdir(outdir) if os.path.isdir(\
            os.path.join(outdir, x))]
    prev_run_ids = [re.match(r'^\d+', x) for x in prev_run_dirs]
    prev_run_ids = [int(x.group()) for x in prev_run_ids if x is not None]
    cur_run_id = max(prev_run_ids, default=-1) + 1
    run_dir = os.path.join(outdir, f'{cur_run_id:05d}-{run_desc}')
    assert not os.path.exists(run_dir)
    os.makedirs(run_dir)
    return run_dir

# From StyleGAN2
class Logger(object):
    """Redirect stderr to stdout, optionally print stdout to a file, and
    optionally force flushing on both stdout and the file."""

    def __init__(self, file_name: str = None, file_mode: str = "w", \
        should_flush: bool = True):
        self.file = None
```

```

    if file_name is not None:
        self.file = open(file_name, file_mode)

    self.should_flush = should_flush
    self.stdout = sys.stdout
    self.stderr = sys.stderr

    sys.stdout = self
    sys.stderr = self

def __enter__(self) -> "Logger":
    return self

def __exit__(self, exc_type: Any, exc_value: Any, \
             traceback: Any) -> None:
    self.close()

def write(self, text: str) -> None:
    """Write text to stdout (and a file) and optionally flush."""
    if len(text) == 0:
        return

    if self.file is not None:
        self.file.write(text)

    self.stdout.write(text)

    if self.should_flush:
        self.flush()

def flush(self) -> None:
    """Flush written text to both stdout and a file, if open."""
    if self.file is not None:
        self.file.flush()

    self.stdout.flush()

def close(self) -> None:
    """Flush, close possible files, and remove  
stdout/stderr mirroring."""
    self.flush()

    # if using multiple loggers, prevent closing in wrong order
    if sys.stdout is self:
        sys.stdout = self.stdout
    if sys.stderr is self:
        sys.stderr = self.stderr

```

```

        if self.file is not None:
            self.file.close()

```

```

[ ]: outdir = "D:/OneDrive/Major Project/HybridModel_37Classes/params/"
run_desc = "test-train"
batch_size = 128
num_classes = len(classes)

run_dir = generate_output_dir(outdir, run_desc)
print(f"Results saved to: {run_dir}")

```

Results saved to: D:/OneDrive/Major
Project/HybridModel_37Classes/params/00009-test-train

```

[ ]: class MyModelCheckpoint(ModelCheckpoint):
    def __init__(self, *args, **kwargs):
        super().__init__(*args, **kwargs)

    def on_epoch_end(self, epoch, logs):
        super().on_epoch_end(epoch, logs)\

        # Also save the optimizer state
        filepath = self._get_file_path(epoch, logs=logs, batch=2)
        filepath = filepath.rsplit( ".", 1 )[ 0 ]
        filepath += ".pkl"

        with open(filepath, 'wb') as fp:
            pickle.dump(
                {
                    'opt': hybridModel.optimizer.get_config(),
                    'epoch': epoch+1
                    # Add additional keys if you need to store more values
                }, fp, protocol=pickle.HIGHEST_PROTOCOL)
        print('\nEpoch %05d: saving optimizer to %s' % (epoch + 1, filepath))

```

```

[ ]: def step_decay_schedule(initial_lr=1e-3, decay_factor=0.75, step_size=10):
    def schedule(epoch):
        return initial_lr * (decay_factor ** np.floor(epoch/step_size))
    return LearningRateScheduler(schedule)

```

```

[ ]: # from tensorflow.keras.applications import DenseNet121, VGG16, ResNet50V2,
↳ MobileNetV2, EfficientNetB0, Xception

img_shape = (299, 299, 3)
num_classes = len(classes)

```

```

def build_model(img_shape, num_classes):
    hybridModel = Sequential()

    pretrained_model = PretrainedModel(
        input_shape = img_shape,
        weights = 'imagenet',
        include_top = False
    )
    for layer in pretrained_model.layers:
        layer.trainable=False

    hybridModel.add(pretrained_model)
    hybridModel.add(Flatten())
    hybridModel.add(Dense(len(classes), activation='softmax'))
    optimizer = keras.optimizers.Adam()
    hybridModel.compile(optimizer, loss='mse', metrics=["accuracy"])
    return hybridModel

def train_model(hybridModel, initial_epoch=0, max_epochs=10):
    start_time = time.time()

    checkpoint_cb = MyModelCheckpoint(
        os.path.join(run_dir, 'model-{epoch:02d}-{val_loss:.2f}.hdf5'),
        monitor='val_loss', verbose=1)

    lr_sched_cb = step_decay_schedule(initial_lr=2.9908e-20, decay_factor=0.75,
    ↪\
                                   step_size=9)

    cb = [checkpoint_cb, lr_sched_cb]

    hist = hybridModel.fit(
        train_generator,
        steps_per_epoch=STEP_SIZE_TRAIN,
        validation_data=validation_generator,
        validation_steps=STEP_SIZE_VALID,
        epochs=max_epochs,
        initial_epoch = initial_epoch,
        callbacks=cb)

```

```

[ ]: # with Logger(os.path.join(run_dir, 'log.txt')):
#     hybridModel = build_model(img_shape, num_classes)
#     train_model(hybridModel)

```

```

[ ]: # !ls '/content/drive/MyDrive/Major Project/Galaxy Morphology/Data/GalaxyZoo2/
    ↪model/params'

```

```
[ ]: MODEL_PATH = 'D:/OneDrive/Major Project/HybridModel_37Classes/params/
↳00008-test-train/model-47-0.15.hdf5'
OPT_PATH = 'D:/OneDrive/Major Project/HybridModel_37Classes/params/
↳00008-test-train/model-47-0.15.pkl'

[ ]: def load_model_data(model_path, opt_path):
    model = load_model(model_path)
    with open(opt_path, 'rb') as fp:
        d = pickle.load(fp)
        epoch = d['epoch']
        opt = d['opt']
        return epoch, model, opt

epoch, hybridModel, opt = load_model_data(MODEL_PATH, OPT_PATH)
hybridModel.compile(optimizer=tf.keras.optimizers.Adam.from_config(opt),
↳loss='mse', metrics=["accuracy"])
with Logger(os.path.join(run_dir, 'log.txt')):
    train_model(hybridModel, initial_epoch=epoch, max_epochs=100)
```

Epoch 48/100
3103/3103 [=====] - ETA: 0s - loss: 0.1562 - accuracy:
0.3715

Epoch 48: saving model to D:/OneDrive/Major
Project/HybridModel_37Classes/params/00009-test-train\model-48-0.15.hdf5

Epoch 00048: saving optimizer to D:/OneDrive/Major
Project/HybridModel_37Classes/params/00009-test-train\model-48-0.15.pkl
3103/3103 [=====] - 8411s 3s/step - loss: 0.1562 -
accuracy: 0.3715 - val_loss: 0.1484 - val_accuracy: 0.2824 - lr: 7.0973e-21

Epoch 49/100
3103/3103 [=====] - ETA: 0s - loss: 0.1562 - accuracy:
0.3713

Epoch 49: saving model to D:/OneDrive/Major
Project/HybridModel_37Classes/params/00009-test-train\model-49-0.15.hdf5

Epoch 00049: saving optimizer to D:/OneDrive/Major
Project/HybridModel_37Classes/params/00009-test-train\model-49-0.15.pkl
3103/3103 [=====] - 7674s 2s/step - loss: 0.1562 -
accuracy: 0.3713 - val_loss: 0.1484 - val_accuracy: 0.2828 - lr: 7.0973e-21

Epoch 50/100
3103/3103 [=====] - ETA: 0s - loss: 0.1562 - accuracy:
0.3712

Epoch 50: saving model to D:/OneDrive/Major
Project/HybridModel_37Classes/params/00009-test-train\model-50-0.15.hdf5

Epoch 00050: saving optimizer to D:/OneDrive/Major
Project/HybridModel_37Classes/params/00009-test-train\model-50-0.15.pkl

3103/3103 [=====] - 7740s 2s/step - loss: 0.1562 - accuracy: 0.3712 - val_loss: 0.1484 - val_accuracy: 0.2925 - lr: 7.0973e-21
Epoch 51/100
3103/3103 [=====] - ETA: 0s - loss: 0.1562 - accuracy: 0.3714
Epoch 51: saving model to D:/OneDrive/Major
Project/HybridModel_37Classes/params/00009-test-train\model-51-0.15.hdf5

Epoch 00051: saving optimizer to D:/OneDrive/Major
Project/HybridModel_37Classes/params/00009-test-train\model-51-0.15.pkl
3103/3103 [=====] - 5644s 2s/step - loss: 0.1562 - accuracy: 0.3714 - val_loss: 0.1484 - val_accuracy: 0.2822 - lr: 7.0973e-21
Epoch 52/100
3103/3103 [=====] - ETA: 0s - loss: 0.1562 - accuracy: 0.3716
Epoch 52: saving model to D:/OneDrive/Major
Project/HybridModel_37Classes/params/00009-test-train\model-52-0.15.hdf5

Epoch 00052: saving optimizer to D:/OneDrive/Major
Project/HybridModel_37Classes/params/00009-test-train\model-52-0.15.pkl
3103/3103 [=====] - 3197s 1s/step - loss: 0.1562 - accuracy: 0.3716 - val_loss: 0.1484 - val_accuracy: 0.2848 - lr: 7.0973e-21
Epoch 53/100
3103/3103 [=====] - ETA: 0s - loss: 0.1562 - accuracy: 0.3700
Epoch 53: saving model to D:/OneDrive/Major
Project/HybridModel_37Classes/params/00009-test-train\model-53-0.15.hdf5

Epoch 00053: saving optimizer to D:/OneDrive/Major
Project/HybridModel_37Classes/params/00009-test-train\model-53-0.15.pkl
3103/3103 [=====] - 6856s 2s/step - loss: 0.1562 - accuracy: 0.3700 - val_loss: 0.1484 - val_accuracy: 0.2854 - lr: 7.0973e-21
Epoch 54/100
3103/3103 [=====] - ETA: 0s - loss: 0.1562 - accuracy: 0.3704
Epoch 54: saving model to D:/OneDrive/Major
Project/HybridModel_37Classes/params/00009-test-train\model-54-0.15.hdf5

Epoch 00054: saving optimizer to D:/OneDrive/Major
Project/HybridModel_37Classes/params/00009-test-train\model-54-0.15.pkl
3103/3103 [=====] - 9877s 3s/step - loss: 0.1562 - accuracy: 0.3704 - val_loss: 0.1485 - val_accuracy: 0.2866 - lr: 7.0973e-21
Epoch 55/100
3103/3103 [=====] - ETA: 0s - loss: 0.1562 - accuracy: 0.3688
Epoch 55: saving model to D:/OneDrive/Major
Project/HybridModel_37Classes/params/00009-test-train\model-55-0.15.hdf5

Epoch 00055: saving optimizer to D:/OneDrive/Major
Project/HybridModel_37Classes/params/00009-test-train\model-55-0.15.pkl
3103/3103 [=====] - 10755s 3s/step - loss: 0.1562 -
accuracy: 0.3688 - val_loss: 0.1484 - val_accuracy: 0.2866 - lr: 5.3230e-21
Epoch 56/100
3103/3103 [=====] - ETA: 0s - loss: 0.1562 - accuracy:
0.3704
Epoch 56: saving model to D:/OneDrive/Major
Project/HybridModel_37Classes/params/00009-test-train\model-56-0.15.hdf5

Epoch 00056: saving optimizer to D:/OneDrive/Major
Project/HybridModel_37Classes/params/00009-test-train\model-56-0.15.pkl
3103/3103 [=====] - 5671s 2s/step - loss: 0.1562 -
accuracy: 0.3704 - val_loss: 0.1484 - val_accuracy: 0.2816 - lr: 5.3230e-21
Epoch 57/100
3103/3103 [=====] - ETA: 0s - loss: 0.1562 - accuracy:
0.3705
Epoch 57: saving model to D:/OneDrive/Major
Project/HybridModel_37Classes/params/00009-test-train\model-57-0.15.hdf5

Epoch 00057: saving optimizer to D:/OneDrive/Major
Project/HybridModel_37Classes/params/00009-test-train\model-57-0.15.pkl
3103/3103 [=====] - 5590s 2s/step - loss: 0.1562 -
accuracy: 0.3705 - val_loss: 0.1484 - val_accuracy: 0.2871 - lr: 5.3230e-21
Epoch 58/100
3103/3103 [=====] - ETA: 0s - loss: 0.1562 - accuracy:
0.3712
Epoch 58: saving model to D:/OneDrive/Major
Project/HybridModel_37Classes/params/00009-test-train\model-58-0.15.hdf5

Epoch 00058: saving optimizer to D:/OneDrive/Major
Project/HybridModel_37Classes/params/00009-test-train\model-58-0.15.pkl
3103/3103 [=====] - 6075s 2s/step - loss: 0.1562 -
accuracy: 0.3712 - val_loss: 0.1484 - val_accuracy: 0.2923 - lr: 5.3230e-21
Epoch 59/100
3103/3103 [=====] - ETA: 0s - loss: 0.1562 - accuracy:
0.3699
Epoch 59: saving model to D:/OneDrive/Major
Project/HybridModel_37Classes/params/00009-test-train\model-59-0.15.hdf5

Epoch 00059: saving optimizer to D:/OneDrive/Major
Project/HybridModel_37Classes/params/00009-test-train\model-59-0.15.pkl
3103/3103 [=====] - 3734s 1s/step - loss: 0.1562 -
accuracy: 0.3699 - val_loss: 0.1484 - val_accuracy: 0.2846 - lr: 5.3230e-21
Epoch 60/100
3103/3103 [=====] - ETA: 0s - loss: 0.1562 - accuracy:
0.3710
Epoch 60: saving model to D:/OneDrive/Major

Project/HybridModel_37Classes/params/00009-test-train\model-60-0.15.hdf5

Epoch 00060: saving optimizer to D:/OneDrive/Major

Project/HybridModel_37Classes/params/00009-test-train\model-60-0.15.pkl

3103/3103 [=====] - 3590s 1s/step - loss: 0.1562 -
accuracy: 0.3710 - val_loss: 0.1484 - val_accuracy: 0.2838 - lr: 5.3230e-21

Epoch 61/100

3103/3103 [=====] - ETA: 0s - loss: 0.1562 - accuracy:
0.3708

Epoch 61: saving model to D:/OneDrive/Major

Project/HybridModel_37Classes/params/00009-test-train\model-61-0.15.hdf5

Epoch 00061: saving optimizer to D:/OneDrive/Major

Project/HybridModel_37Classes/params/00009-test-train\model-61-0.15.pkl

3103/3103 [=====] - 3609s 1s/step - loss: 0.1562 -
accuracy: 0.3708 - val_loss: 0.1484 - val_accuracy: 0.2777 - lr: 5.3230e-21

Epoch 62/100

3103/3103 [=====] - ETA: 0s - loss: 0.1562 - accuracy:
0.3704

Epoch 62: saving model to D:/OneDrive/Major

Project/HybridModel_37Classes/params/00009-test-train\model-62-0.15.hdf5

Epoch 00062: saving optimizer to D:/OneDrive/Major

Project/HybridModel_37Classes/params/00009-test-train\model-62-0.15.pkl

3103/3103 [=====] - 3307s 1s/step - loss: 0.1562 -
accuracy: 0.3704 - val_loss: 0.1484 - val_accuracy: 0.2889 - lr: 5.3230e-21

Epoch 63/100

3103/3103 [=====] - ETA: 0s - loss: 0.1562 - accuracy:
0.3711

Epoch 63: saving model to D:/OneDrive/Major

Project/HybridModel_37Classes/params/00009-test-train\model-63-0.15.hdf5

Epoch 00063: saving optimizer to D:/OneDrive/Major

Project/HybridModel_37Classes/params/00009-test-train\model-63-0.15.pkl

3103/3103 [=====] - 3535s 1s/step - loss: 0.1562 -
accuracy: 0.3711 - val_loss: 0.1484 - val_accuracy: 0.2875 - lr: 5.3230e-21

Epoch 64/100

3103/3103 [=====] - ETA: 0s - loss: 0.1562 - accuracy:
0.3695

Epoch 64: saving model to D:/OneDrive/Major

Project/HybridModel_37Classes/params/00009-test-train\model-64-0.15.hdf5

Epoch 00064: saving optimizer to D:/OneDrive/Major

Project/HybridModel_37Classes/params/00009-test-train\model-64-0.15.pkl

3103/3103 [=====] - 3905s 1s/step - loss: 0.1562 -
accuracy: 0.3695 - val_loss: 0.1484 - val_accuracy: 0.2909 - lr: 3.9922e-21

Epoch 65/100

3103/3103 [=====] - ETA: 0s - loss: 0.1562 - accuracy:

0.3722
Epoch 65: saving model to D:/OneDrive/Major
Project/HybridModel_37Classes/params/00009-test-train\model-65-0.15.hdf5

Epoch 00065: saving optimizer to D:/OneDrive/Major
Project/HybridModel_37Classes/params/00009-test-train\model-65-0.15.pkl
3103/3103 [=====] - 9059s 3s/step - loss: 0.1562 -
accuracy: 0.3722 - val_loss: 0.1484 - val_accuracy: 0.2842 - lr: 3.9922e-21
Epoch 66/100
3103/3103 [=====] - ETA: 0s - loss: 0.1562 - accuracy:
0.3712
Epoch 66: saving model to D:/OneDrive/Major
Project/HybridModel_37Classes/params/00009-test-train\model-66-0.15.hdf5

Epoch 00066: saving optimizer to D:/OneDrive/Major
Project/HybridModel_37Classes/params/00009-test-train\model-66-0.15.pkl
3103/3103 [=====] - 5990s 2s/step - loss: 0.1562 -
accuracy: 0.3712 - val_loss: 0.1484 - val_accuracy: 0.2714 - lr: 3.9922e-21
Epoch 67/100
3103/3103 [=====] - ETA: 0s - loss: 0.1562 - accuracy:
0.3706
Epoch 67: saving model to D:/OneDrive/Major
Project/HybridModel_37Classes/params/00009-test-train\model-67-0.15.hdf5

Epoch 00067: saving optimizer to D:/OneDrive/Major
Project/HybridModel_37Classes/params/00009-test-train\model-67-0.15.pkl
3103/3103 [=====] - 3509s 1s/step - loss: 0.1562 -
accuracy: 0.3706 - val_loss: 0.1484 - val_accuracy: 0.2824 - lr: 3.9922e-21
Epoch 68/100
3103/3103 [=====] - ETA: 0s - loss: 0.1562 - accuracy:
0.3710
Epoch 68: saving model to D:/OneDrive/Major
Project/HybridModel_37Classes/params/00009-test-train\model-68-0.15.hdf5

Epoch 00068: saving optimizer to D:/OneDrive/Major
Project/HybridModel_37Classes/params/00009-test-train\model-68-0.15.pkl
3103/3103 [=====] - 3513s 1s/step - loss: 0.1562 -
accuracy: 0.3710 - val_loss: 0.1484 - val_accuracy: 0.2834 - lr: 3.9922e-21
Epoch 69/100
3103/3103 [=====] - ETA: 0s - loss: 0.1562 - accuracy:
0.3723
Epoch 69: saving model to D:/OneDrive/Major
Project/HybridModel_37Classes/params/00009-test-train\model-69-0.15.hdf5

Epoch 00069: saving optimizer to D:/OneDrive/Major
Project/HybridModel_37Classes/params/00009-test-train\model-69-0.15.pkl
3103/3103 [=====] - 3778s 1s/step - loss: 0.1562 -
accuracy: 0.3723 - val_loss: 0.1484 - val_accuracy: 0.2913 - lr: 3.9922e-21

Epoch 70/100
3103/3103 [=====] - ETA: 0s - loss: 0.1562 - accuracy: 0.3707
Epoch 70: saving model to D:/OneDrive/Major
Project/HybridModel_37Classes/params/00009-test-train\model-70-0.15.hdf5

Epoch 00070: saving optimizer to D:/OneDrive/Major
Project/HybridModel_37Classes/params/00009-test-train\model-70-0.15.pkl
3103/3103 [=====] - 4028s 1s/step - loss: 0.1562 - accuracy: 0.3707 - val_loss: 0.1485 - val_accuracy: 0.2814 - lr: 3.9922e-21

Epoch 71/100
3103/3103 [=====] - ETA: 0s - loss: 0.1562 - accuracy: 0.3712
Epoch 71: saving model to D:/OneDrive/Major
Project/HybridModel_37Classes/params/00009-test-train\model-71-0.15.hdf5

Epoch 00071: saving optimizer to D:/OneDrive/Major
Project/HybridModel_37Classes/params/00009-test-train\model-71-0.15.pkl
3103/3103 [=====] - 4033s 1s/step - loss: 0.1562 - accuracy: 0.3712 - val_loss: 0.1484 - val_accuracy: 0.2885 - lr: 3.9922e-21

Epoch 72/100
3103/3103 [=====] - ETA: 0s - loss: 0.1562 - accuracy: 0.3697
Epoch 72: saving model to D:/OneDrive/Major
Project/HybridModel_37Classes/params/00009-test-train\model-72-0.15.hdf5

Epoch 00072: saving optimizer to D:/OneDrive/Major
Project/HybridModel_37Classes/params/00009-test-train\model-72-0.15.pkl
3103/3103 [=====] - 4178s 1s/step - loss: 0.1562 - accuracy: 0.3697 - val_loss: 0.1484 - val_accuracy: 0.2834 - lr: 3.9922e-21

Epoch 73/100
72/3103 [...] - ETA: 1:32:28 - loss: 0.1567 - accuracy: 0.3700