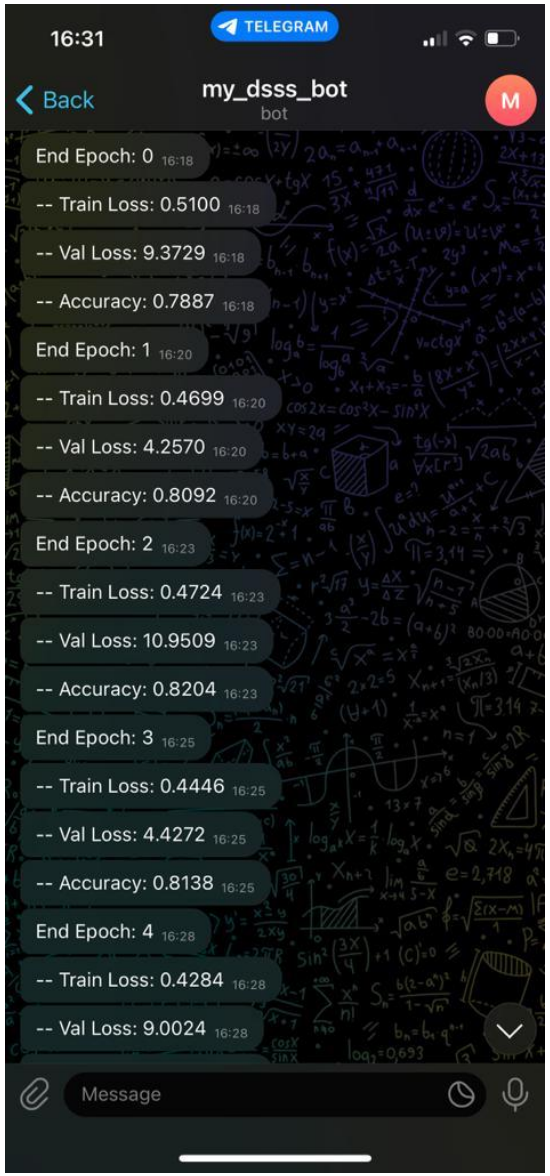

```
def dataloader():
datagen = ImageDataGenerator(rescale=1./255)
data_directory = '/content/drive/MyDrive/DSSS_9/train'
train_data = datagen.flow_from_directory(directory=data_directory,
                                         target_size=(256, 256),
                                         validation_split=0.8,
                                         class_mode='categorical',
                                         batch_size=32)

val_datagen = ImageDataGenerator(rescale=1./255)
val_data = val_datagen.flow_from_directory(directory=data_directory,
                                           target_size=(256, 256),
                                           validation_split=0.1,
                                           class_mode='categorical',
                                           batch_size=32)

test_datagen = ImageDataGenerator(rescale=1./255)
test_data = test_datagen.flow_from_directory(directory=data_directory,
                                             target_size=(256, 256),
                                             validation_split=0.1,
                                             class_mode='categorical',
                                             batch_size=32)
```

```
for i in range(1000):
    urllib.request.urlretrieve(urls_dogs[i],path + '\{0}_{1}.jpg'.format(keyword1 ,i))
    image1 = Image.open(path + '\{0}_{1}.jpg'.format(keyword1 ,i))
    image1 = image1.resize((256, 256), Image. ANTIALIAS)
    image1 = image1.save(path + '\{0}_{1}.jpg'.format(keyword1 ,i))

    urllib.request.urlretrieve(urls_cats[i],path + '\{0}_{1}.jpg'.format(keyword2 ,i))
    image2 = Image.open(path + '\{0}_{1}.jpg'.format(keyword2, i))
    image2 = image2.resize((256, 256), Image. ANTIALIAS)
    image2 = image2.save(path + '\{0}_{1}.jpg'.format(keyword2, i))

    urllib.request.urlretrieve(urls_apes[i],path + '\{0}_{1}.jpg'.format(keyword3 ,i))
    image3 = Image.open(path + '\{0}_{1}.jpg'.format(keyword3, i))
    image3 = image3.resize((256, 256), Image. ANTIALIAS)
    image3 = image3.save(path + '\{0}_{1}.jpg'.format(keyword3, i))
```

```
class BotCallback(tf.keras.callbacks.Callback):
    def __init__(self,personal_token):
        self.personal_token = personal_token
        self.ping_url = 'https://api.telegram.org/bot'+str(self.personal_token)+'/getUpdates'
        self.response = requests.get(self.ping_url).json()
        self.chat_id = self.response['result'][0]['message']['chat']['id']

    def send_message(self,message):
        self.ping_url = 'https://api.telegram.org/bot'+str(self.personal_token)+'/sendMessage?'+\
                        'chat_id='+str(self.chat_id)+\
                        '&parse_mode=Markdown'+\
                        '&text='+message
        self.response = requests.get(self.ping_url)
        return self.ping_url

    def on_epoch_end(self, epoch, epoch_logs = {}):
        # TODO
        self.send_message("End Epoch: {}".format(epoch))
        self.send_message("-- Train Loss: {0:.4f}".format(epoch_logs.get('loss')))
        self.send_message("-- Val Loss: {0:.4f}".format(epoch_logs.get('val_loss')))
        self.send_message("-- Accuracy: {0:.4f}".format(epoch_logs.get('accuracy')))
        pass

model.fit(train_data, epochs=10, validation_data=val_data,
          callbacks=[BotCallback('5854608172:AAFcTPnDUk0ATkaJV3lXpxg1WQcGqhfC6y4')])
```



NAME: ABHIK SARKAR
MATRIKELNUMMER: 23149662
IDM Id : lo88xide