ॐ

VIT – CHENNAI CAMPUS

BCSE332P – DEEP LEARNING LAB

21BAI1364 – VITTA VISHNU DATTA

DEEP LEARNING LAB ASSESSMENT – 6

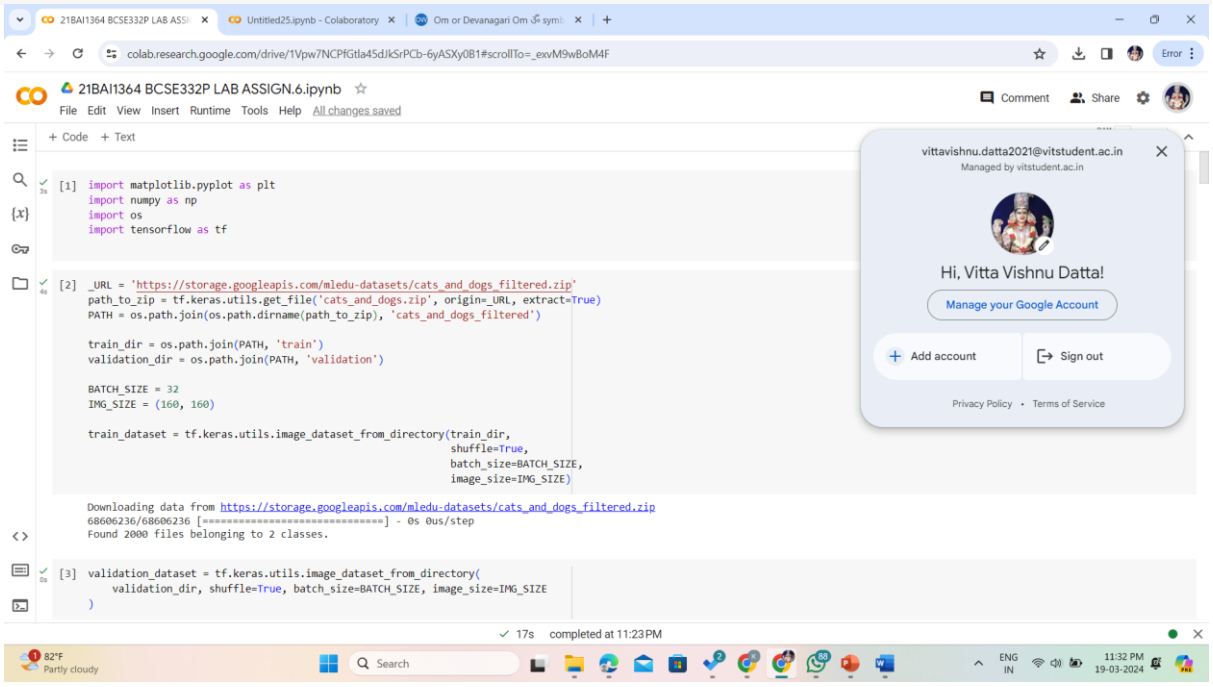
1. Simple Transfer Learning with Keras

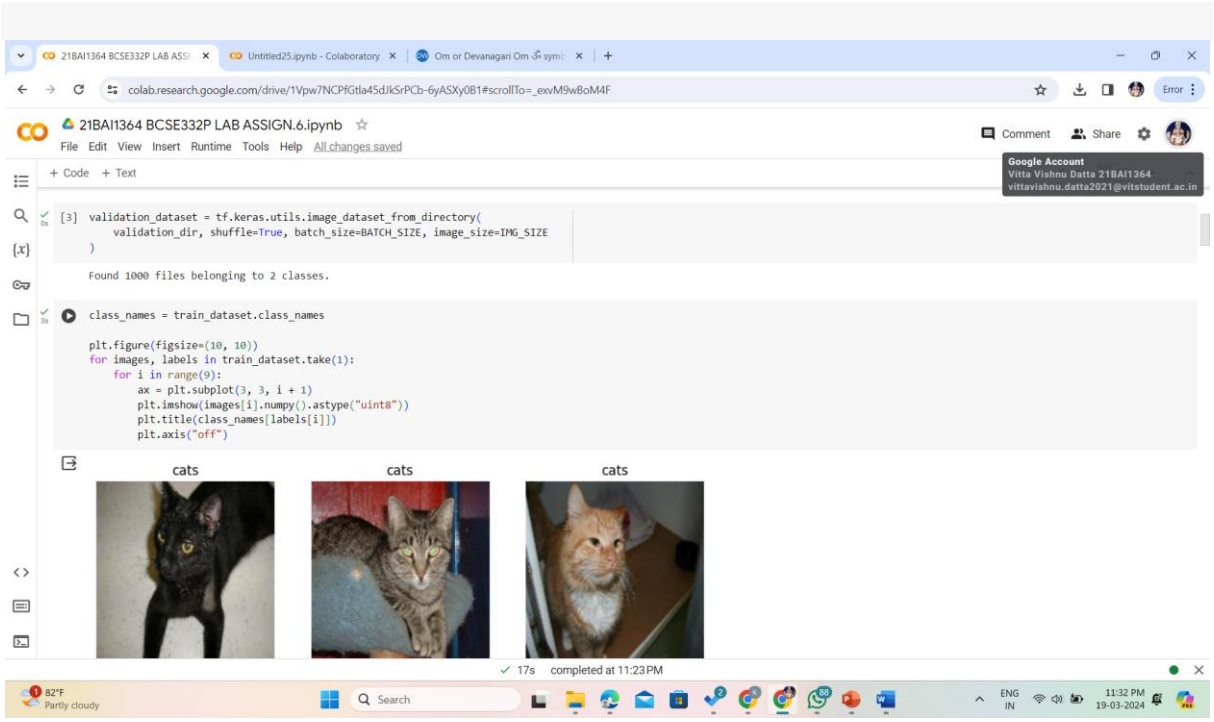
```
import matplotlib.pyplot as plt
import numpy as np
import os
import tensorflow as tf

_URL = 'https://storage.googleapis.com/mledu-
datasets/cats_and_dogs_filtered.zip'
path_to_zip = tf.keras.utils.get_file('cats_and_dogs.zip', origin=_URL,
extract=True)
PATH = os.path.join(os.path.dirname(path_to_zip),
'cats_and_dogs_filtered')

train_dir = os.path.join(PATH, 'train')
validation_dir = os.path.join(PATH, 'validation')

BATCH_SIZE = 32
IMG_SIZE = (160, 160)

train_dataset = tf.keras.utils.image_dataset_from_directory(train_dir,
                                                            shuffle=Tru
e,
                                                            batch_size=
BATCH_SIZE,
                                                            image_size=
IMG_SIZE)
validation_dataset = tf.keras.utils.image_dataset_from_directory(
    validation_dir, shuffle=True, batch_size=BATCH_SIZE,
image_size=IMG_SIZE
)
class_names = train_dataset.class_names

plt.figure(figsize=(10, 10))
for images, labels in train_dataset.take(1):
    for i in range(9):
```

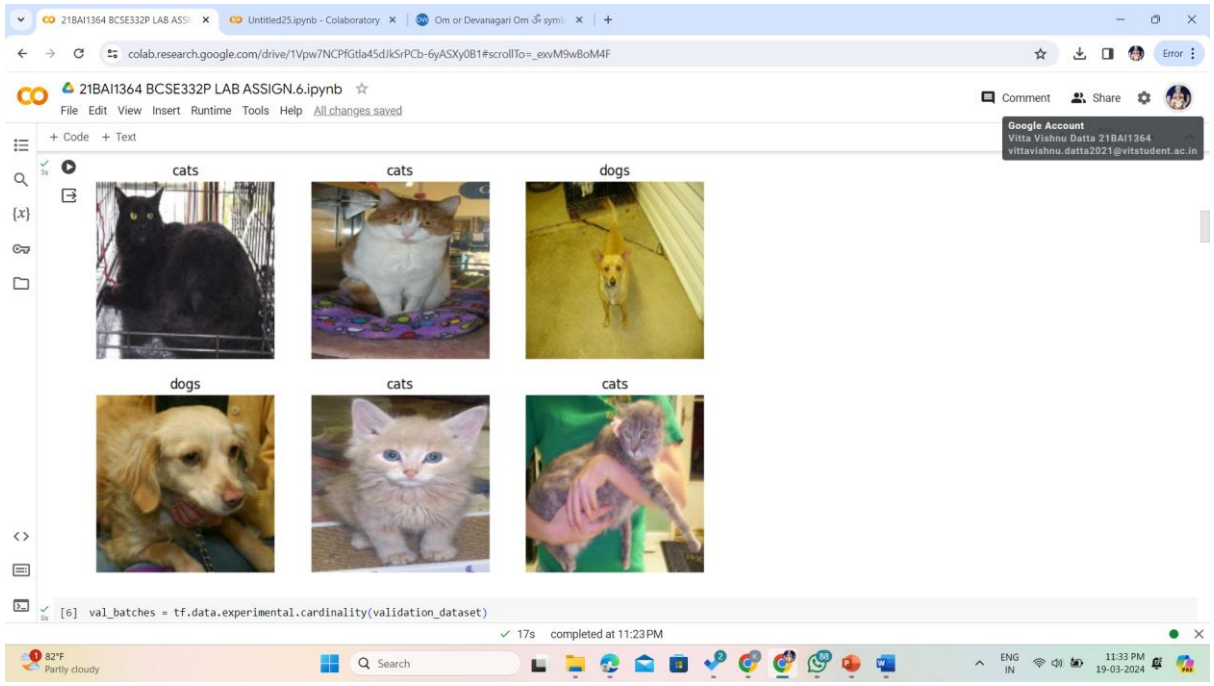
```
        ax = plt.subplot(3, 3, i + 1)
        plt.imshow(images[i].numpy().astype("uint8"))
        plt.title(class_names[labels[i]])
        plt.axis("off")
val_batches = tf.data.experimental.cardinality(validation_dataset)
test_dataset = validation_dataset.take(val_batches // 5)
validation_dataset = validation_dataset.skip(val_batches // 5)
print(
    "Number of validation batches: %d"
    % tf.data.experimental.cardinality(validation_dataset)
)
print("Number of test batches: %d" %
tf.data.experimental.cardinality(test_dataset))
AUTOTUNE = tf.data.AUTOTUNE

train_dataset = train_dataset.prefetch(buffer_size=AUTOTUNE)
validation_dataset = validation_dataset.prefetch(buffer_size=AUTOTUNE)
test_dataset = test_dataset.prefetch(buffer_size=AUTOTUNE)
data_augmentation = tf.keras.Sequential(
    [
        tf.keras.layers.RandomFlip("horizontal"),
        tf.keras.layers.RandomRotation(0.2),
    ]
)
for image, _ in train_dataset.take(1):
    plt.figure(figsize=(10, 10))
    first_image = image[0]
    for i in range(9):
        ax = plt.subplot(3, 3, i + 1)
        augmented_image = data_augmentation(tf.expand_dims(first_image,
0))
        plt.imshow(augmented_image[0] / 255)
        plt.axis("off")
preprocess_input = tf.keras.applications.mobilenet_v2.preprocess_input
rescale = tf.keras.layers.Rescaling(1.0 / 127.5, offset=-1)
# Create the base model from the pre-trained model MobileNet V2
IMG_SHAPE = IMG_SIZE + (3,)
base_model = tf.keras.applications.MobileNetV2(
    input_shape=IMG_SHAPE, include_top=False, weights="imagenet"
)
image_batch, label_batch = next(iter(train_dataset))
feature_batch = base_model(image_batch)
print(feature_batch.shape)
base_model.trainable = False
# Let's take a look at the base model architecture
base_model.summary()
global_average_layer = tf.keras.layers.GlobalAveragePooling2D()
```

```
feature_batch_average = global_average_layer(feature_batch)
print(feature_batch_average.shape)
prediction_layer = tf.keras.layers.Dense(1)
prediction_batch = prediction_layer(feature_batch_average)
print(prediction_batch.shape)
inputs = tf.keras.Input(shape=(160, 160, 3))
x = data_augmentation(inputs)
x = preprocess_input(x)
x = base_model(x, training=False)
x = global_average_layer(x)
x = tf.keras.layers.Dropout(0.2)(x)
outputs = prediction_layer(x)
model = tf.keras.Model(inputs, outputs)
base_learning_rate = 0.0001
model.compile(
    optimizer=tf.keras.optimizers.Adam(learning_rate=base_learning_rate
),
    loss=tf.keras.losses.BinaryCrossentropy(from_logits=True),
    metrics=["accuracy"],
)
len(model.trainable_variables)
initial_epochs = 10

loss0, accuracy0 = model.evaluate(validation_dataset)
print("initial loss: {:.2f}".format(loss0))
print("initial accuracy: {:.2f}".format(accuracy0))
history = model.fit(
    train_dataset, epochs=initial_epochs,
validation_data=validation_dataset
)
base_model.trainable = True
model.compile(
    loss=tf.keras.losses.BinaryCrossentropy(from_logits=True),
    optimizer=tf.keras.optimizers.RMSprop(learning_rate=base_learning_r
ate / 10),
    metrics=["accuracy"],
)
len(model.trainable_variables)
fine_tune_epochs = 10
total_epochs = initial_epochs + fine_tune_epochs

history_fine = model.fit(
    train_dataset,
    epochs=total_epochs,
    initial_epoch=history.epoch[-1],
    validation_data=validation_dataset,
)
```

```
# Retrieve a batch of images from the test set
image_batch, label_batch = test_dataset.as_numpy_iterator().next()
predictions = model.predict_on_batch(image_batch).flatten()

# Apply a sigmoid since our model returns logits
predictions = tf.nn.sigmoid(predictions)
predictions = tf.where(predictions < 0.5, 0, 1)

print("Predictions:\n", predictions.numpy())
print("Labels:\n", label_batch)

plt.figure(figsize=(10, 10))
for i in range(9):
    ax = plt.subplot(3, 3, i + 1)
    plt.imshow(image_batch[i].astype("uint8"))
    plt.title(class_names[predictions[i]])
    plt.axis("off")
# Save the model
model_name = "cat-dog-tuned"
model.save(model_name)
```




21BAI1364 BCSE332P LAB ASSIGN.6.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment Share
Google Account
Vitta Vishnu Datta 21BAI1364
vittavishnu.datta2021@vitstudent.ac.in
+ Code + Text
[3] validation_dataset = tf.keras.utils.image_dataset_from_directory(
validation_dir, shuffle=True, batch_size=BATCH_SIZE, image_size=IMG_SIZE
)
Found 1000 files belonging to 2 classes.
class_names = train_dataset.class_names
plt.figure(figsize=(10, 10))
for images, labels in train_dataset.take(1):
for i in range(9):
ax = plt.subplot(3, 3, i + 1)
plt.imshow(images[i].numpy().astype("uint8"))
plt.title(class_names[labels[i]])
plt.axis("off")
cats
cats
cats
17s completed at 11:23PM
82°F Partly cloudy
Search
ENG IN
11:32 PM 19-03-2024


21BAI1364 BCSE332P LAB ASSIGN.6.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment Share
Google Account
Vitta Vishnu Datta 21BAI1364
vittavishnu.datta2021@vitstudent.ac.in
+ Code + Text
cats
cats
dogs
dogs
cats
cats
[6] val_batches = tf.data.experimental.cardinality(validation_dataset)
17s completed at 11:23PM
82°F Partly cloudy
Search
ENG IN
11:33 PM 19-03-2024

21BAI1364 BCSE332P LAB ASSIGN.6.ipynb

File Edit View Insert Runtime Tools Help All changes saved

Comment Share

Google Account
Vitta Vishnu Datta 21BAI1364
vittavishnu.datta2021@vitstudent.ac.in

+ Code + Text

```
[6] val_batches = tf.data.experimental.cardinality(validation_dataset)
    test_dataset = validation_dataset.take(val_batches // 5)
    validation_dataset = validation_dataset.skip(val_batches // 5)
```

```
[7] print(
        "Number of validation batches: %d"
        % tf.data.experimental.cardinality(validation_dataset)
    )
    print("Number of test batches: %d" % tf.data.experimental.cardinality(test_dataset))
```

```
Number of validation batches: 26
Number of test batches: 6
```

```
AUTOTUNE = tf.data.AUTOTUNE

train_dataset = train_dataset.prefetch(buffer_size=AUTOTUNE)
validation_dataset = validation_dataset.prefetch(buffer_size=AUTOTUNE)
test_dataset = test_dataset.prefetch(buffer_size=AUTOTUNE)
```

```
[9] data_augmentation = tf.keras.Sequential(
        [
            tf.keras.layers.RandomFlip("horizontal"),
            tf.keras.layers.RandomRotation(0.2),
        ]
    )
```

```
[18] for image, _ in train_dataset.take(1):
         plt.figure(figsize=(10, 10))
```

✓ 17s completed at 11:23 PM

82°F Partly cloudy | Search | ENG IN | 11:33 PM 19-03-2024

21BAI1364 BCSE332P LAB ASSIGN.6.ipynb

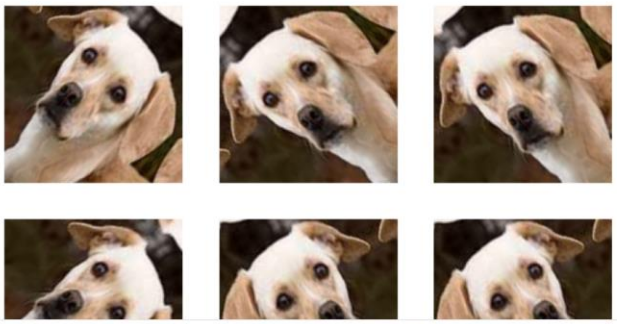
File Edit View Insert Runtime Tools Help All changes saved

Comment Share

Google Account
Vitta Vishnu Datta 21BAI1364
vittavishnu.datta2021@vitstudent.ac.in

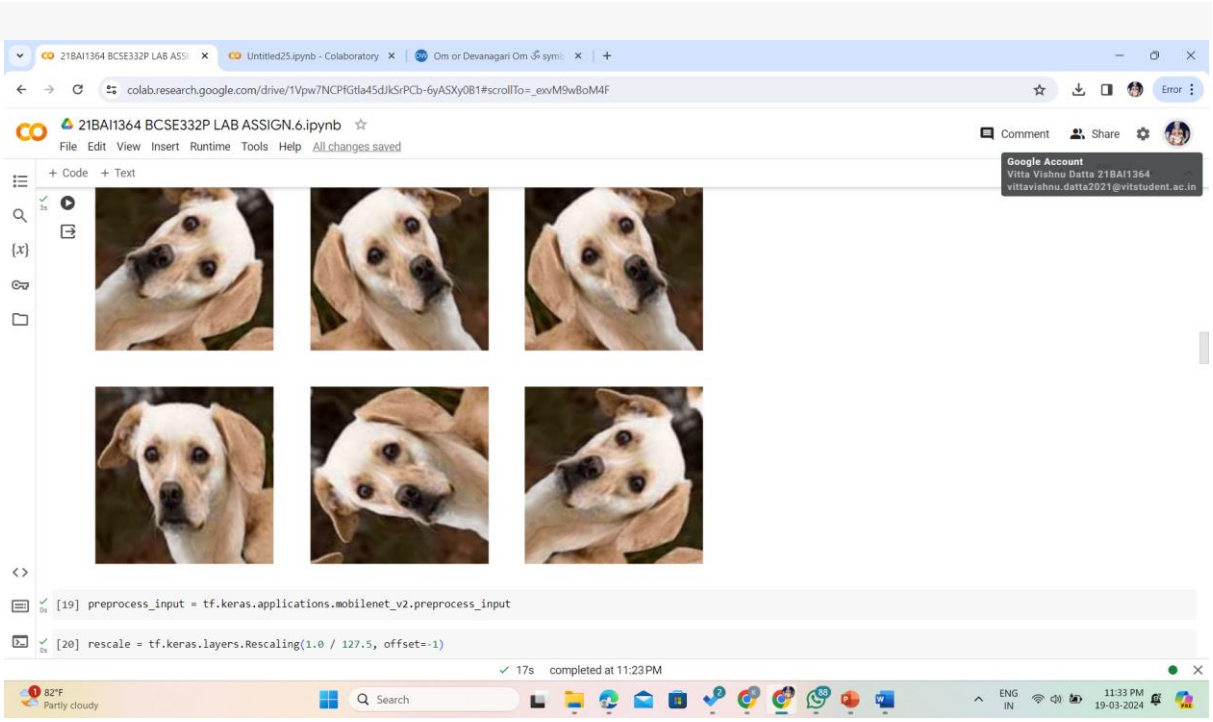
+ Code + Text

```
[18] for image, _ in train_dataset.take(1):
         plt.figure(figsize=(10, 10))
         first_image = image[0]
         for i in range(9):
             ax = plt.subplot(3, 3, i + 1)
             augmented_image = data_augmentation(tf.expand_dims(first_image, 0))
             plt.imshow(augmented_image[0] / 255)
             plt.axis("off")
```

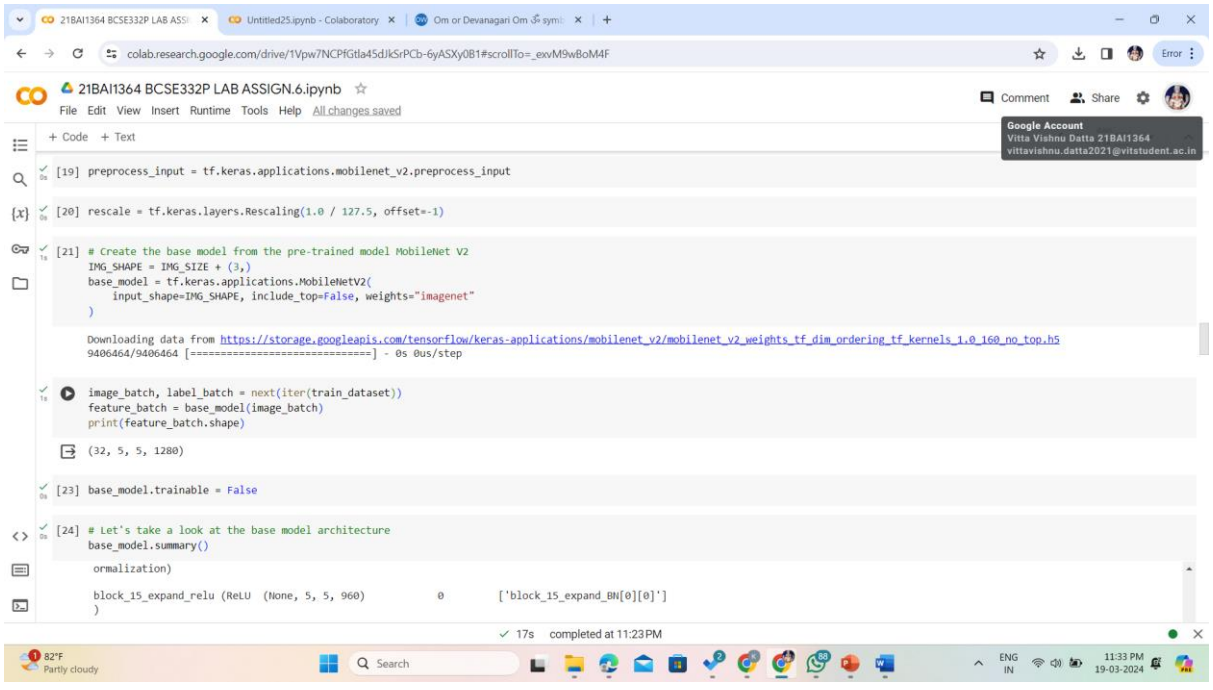


✓ 17s completed at 11:23 PM

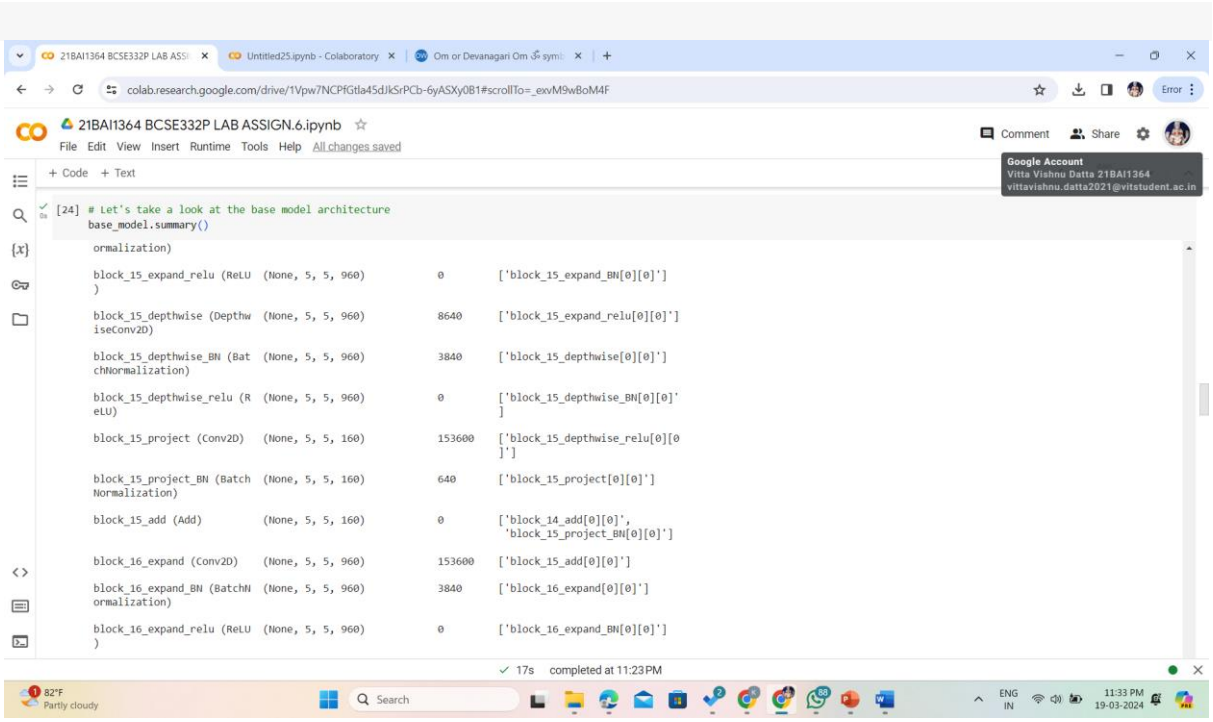
82°F Partly cloudy | Search | ENG IN | 11:33 PM 19-03-2024

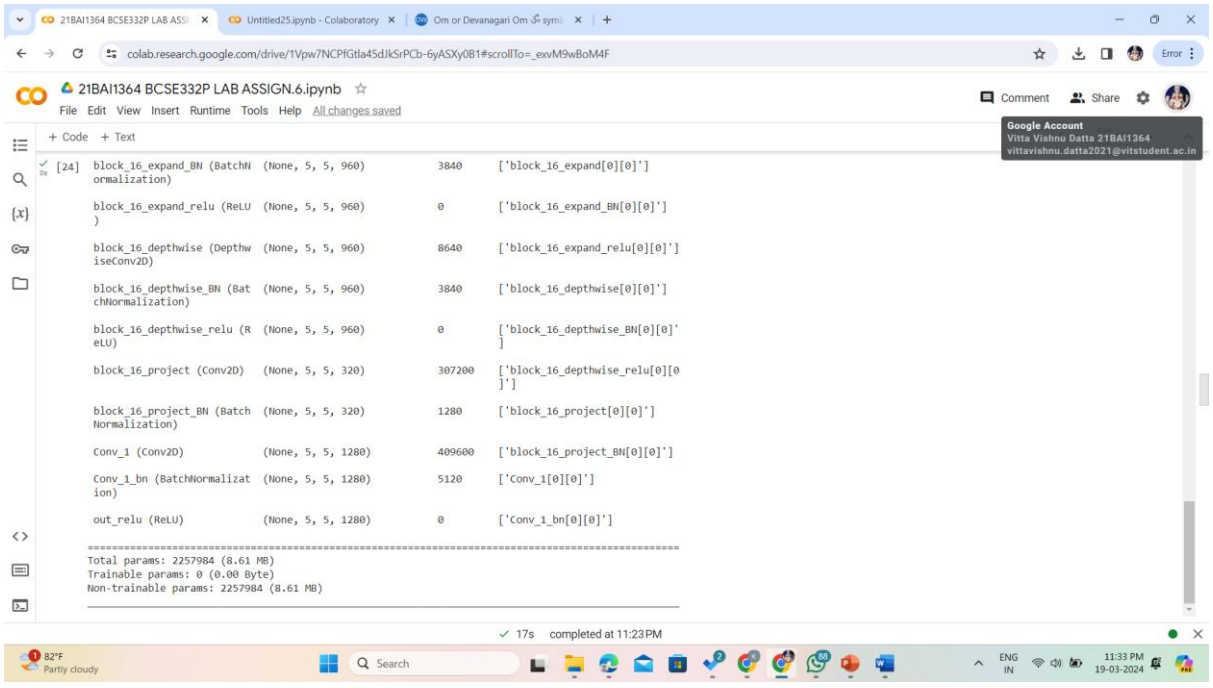


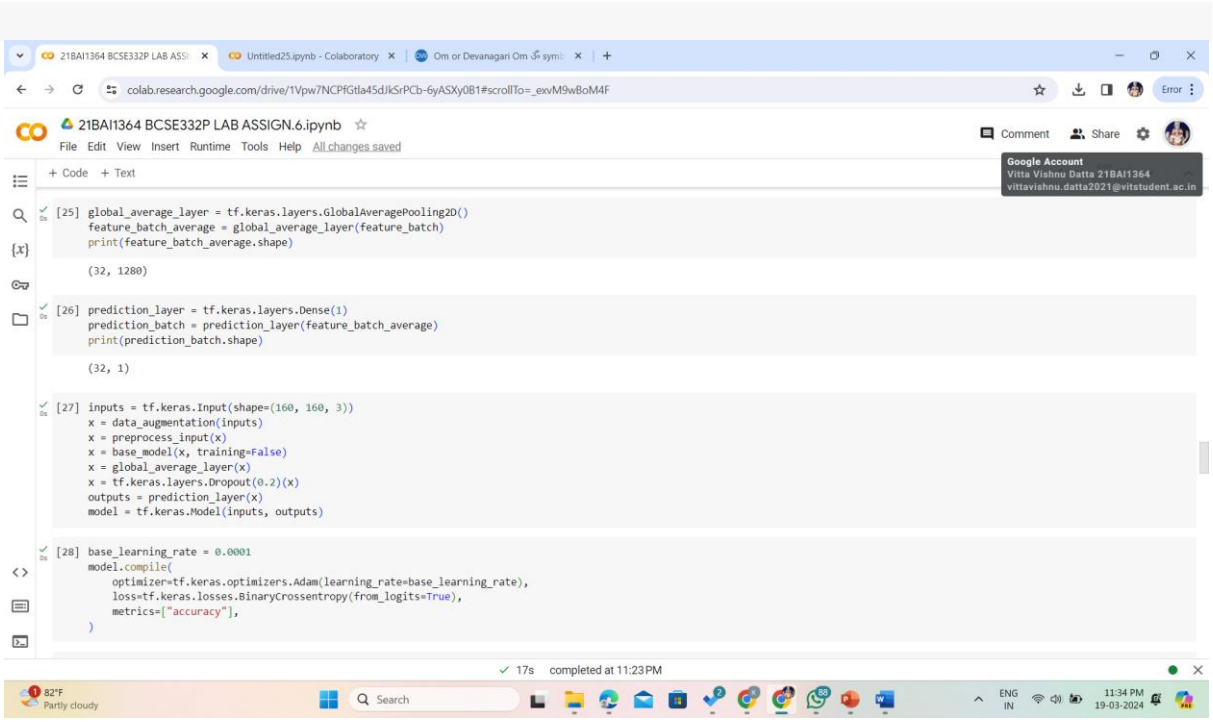
```
[19] preprocess_input = tf.keras.applications.mobilenet_v2.preprocess_input

[20] rescale = tf.keras.layers.Rescaling(1.0 / 127.5, offset=-1)
```

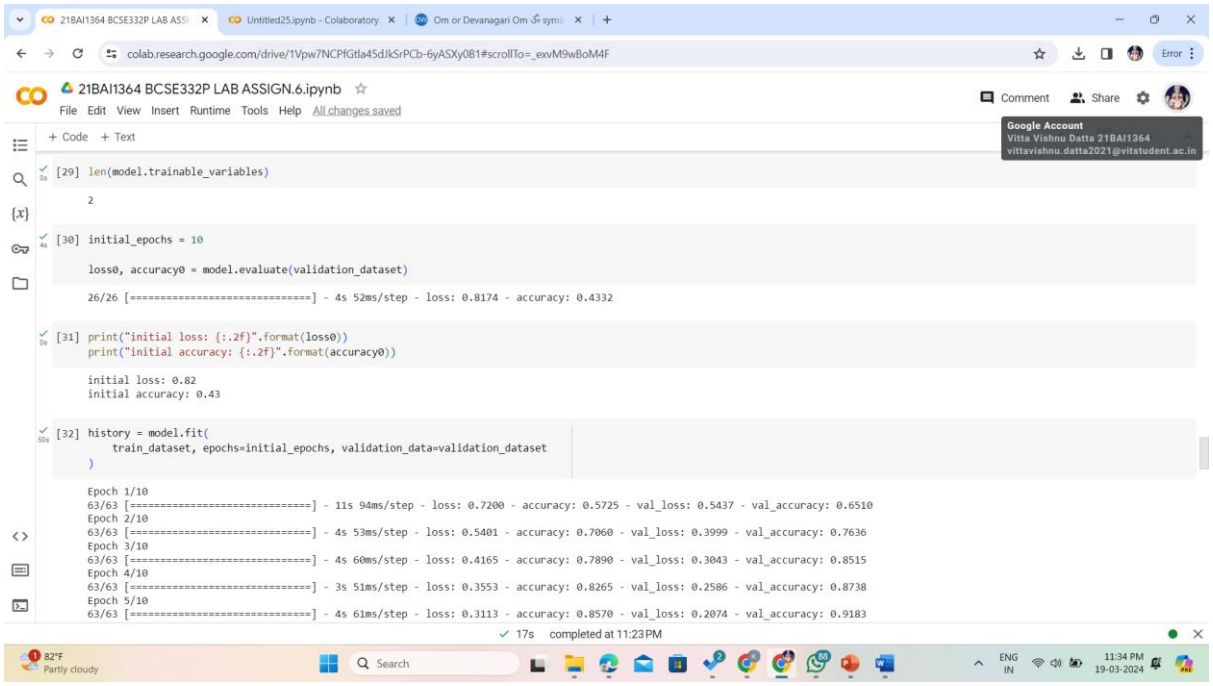


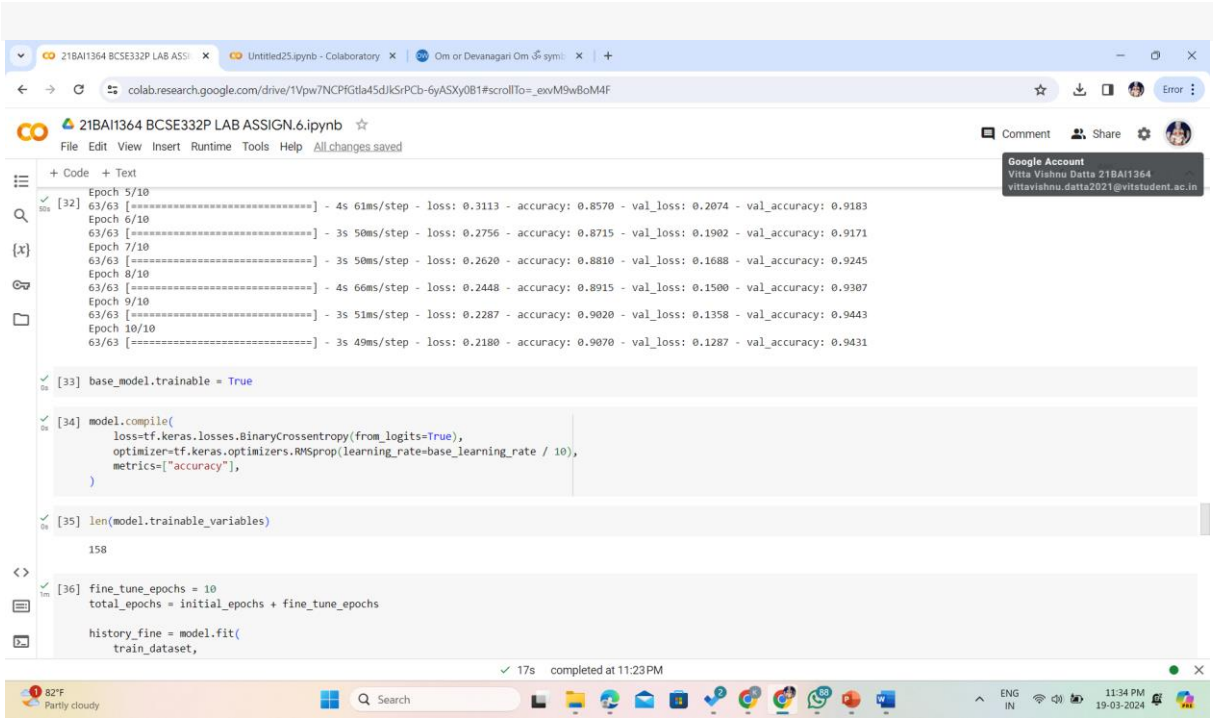
```
[21] # Create the base model from the pre-trained model MobileNet V2
     IMG_SHAPE = IMG_SIZE + (3,)
     base_model = tf.keras.applications.MobileNetV2(
         input_shape=IMG_SHAPE, include_top=False, weights="imagenet"
     )

     Downloading data from https://storage.googleapis.com/tensorflow/keras-applications/mobilenet_v2/mobilenet_v2_weights_tf_dim_ordering_tf_kernels_1.0_160_no_top.h5
     9406464/9406464 [==============================] - 0s 0us/step

     image_batch, label_batch = next(iter(train_dataset))
     feature_batch = base_model(image_batch)
     print(feature_batch.shape)

     (32, 5, 5, 1280)

[23] base_model.trainable = False

[24] # Let's take a look at the base model architecture
     base_model.summary()

     ormalization)

     block_15_expand_relu (ReLU  (None, 5, 5, 960)          0         ['block_15_expand_BN[0][0]']
     )
```

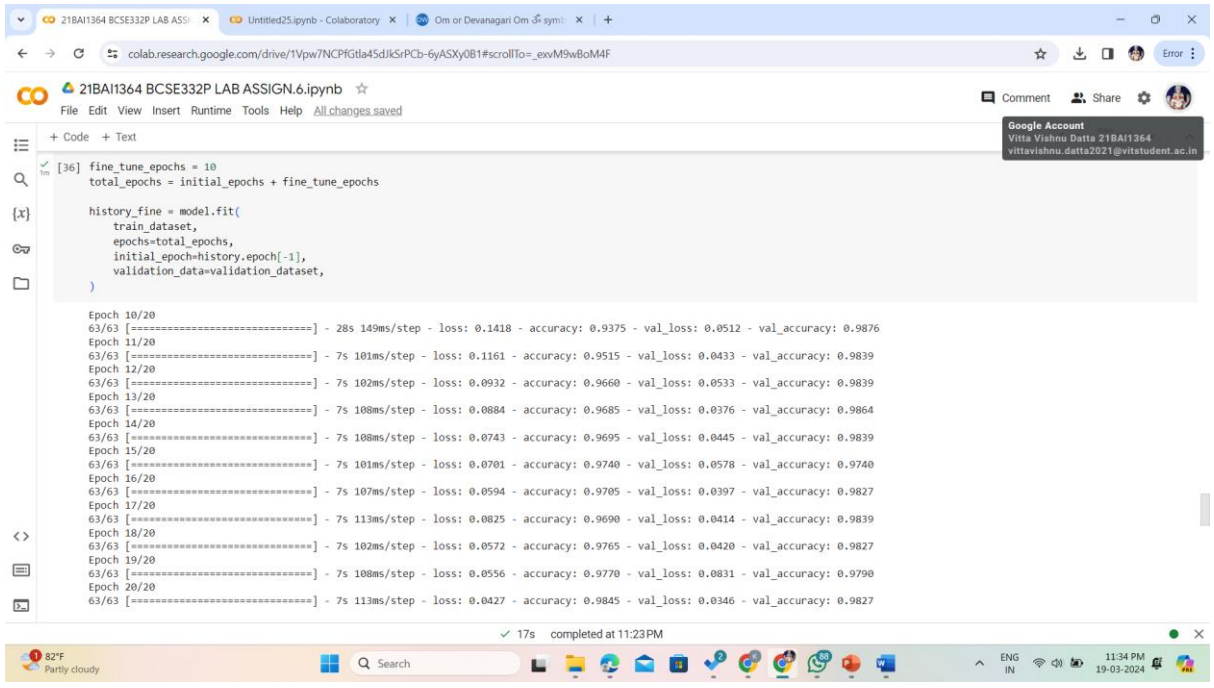

21BAI1364 BCSE332P LAB ASSIGN.6.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment Share
Google Account
Vitta Vishnu Datta 21BAI1364
vittavishnu.datta2021@vitstudent.ac.in
+ Code + Text
[24] # Let's take a look at the base model architecture
base_model.summary()
ormalization)
block_15_expand_relu (ReLU) (None, 5, 5, 960) 0 ['block_15_expand_BN[0][0]']
block_15_depthwise (DepthwiseConv2D) (None, 5, 5, 960) 8640 ['block_15_expand_relu[0][0]']
block_15_depthwise_BN (BatchNormalization) (None, 5, 5, 960) 3840 ['block_15_depthwise[0][0]']
block_15_depthwise_relu (ReLU) (None, 5, 5, 960) 0 ['block_15_depthwise_BN[0][0]']
block_15_project (Conv2D) (None, 5, 5, 160) 153600 ['block_15_depthwise_relu[0][0]']
block_15_project_BN (BatchNormalization) (None, 5, 5, 160) 640 ['block_15_project[0][0]']
block_15_add (Add) (None, 5, 5, 160) 0 ['block_14_add[0][0]', 'block_15_project_BN[0][0]']
block_16_expand (Conv2D) (None, 5, 5, 960) 153600 ['block_15_add[0][0]']
block_16_expand_BN (BatchNormalization) (None, 5, 5, 960) 3840 ['block_16_expand[0][0]']
block_16_expand_relu (ReLU) (None, 5, 5, 960) 0 ['block_16_expand_BN[0][0]']
17s completed at 11:23PM
82°F Partly cloudy
Search
ENG IN
11:33 PM 19-03-2024

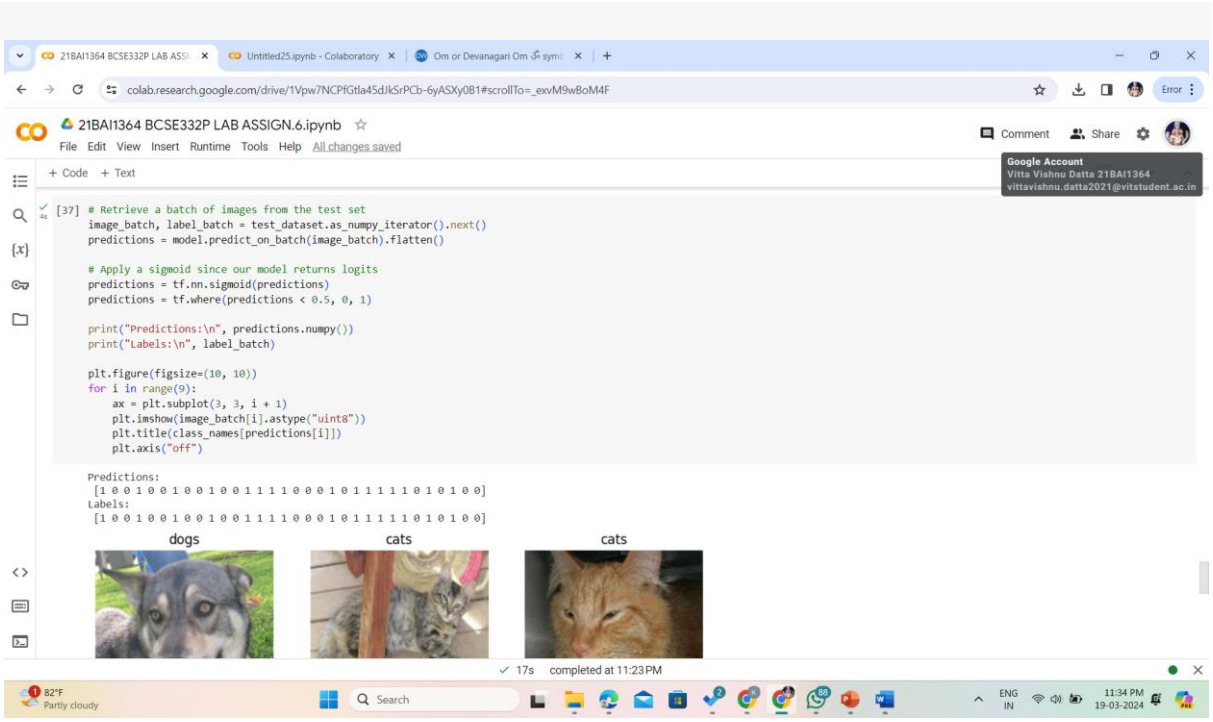

21BAI1364 BCSE332P LAB ASSIGN.6.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment Share
Google Account
Vitta Vishnu Datta 21BAI1364
vittavishnu.datta2021@vitstudent.ac.in
+ Code + Text
[24] block_16_expand_BN (BatchNormalization) (None, 5, 5, 960) 3840 ['block_16_expand[0][0]']
block_16_expand_relu (ReLU) (None, 5, 5, 960) 0 ['block_16_expand_BN[0][0]']
block_16_depthwise (DepthwiseConv2D) (None, 5, 5, 960) 8640 ['block_16_expand_relu[0][0]']
block_16_depthwise_BN (BatchNormalization) (None, 5, 5, 960) 3840 ['block_16_depthwise[0][0]']
block_16_depthwise_relu (ReLU) (None, 5, 5, 960) 0 ['block_16_depthwise_BN[0][0]']
block_16_project (Conv2D) (None, 5, 5, 320) 307200 ['block_16_depthwise_relu[0][0]']
block_16_project_BN (BatchNormalization) (None, 5, 5, 320) 1280 ['block_16_project[0][0]']
Conv_1 (Conv2D) (None, 5, 5, 1280) 409600 ['block_16_project_BN[0][0]']
Conv_1_bn (BatchNormalization) (None, 5, 5, 1280) 5120 ['Conv_1[0][0]']
out_relu (ReLU) (None, 5, 5, 1280) 0 ['Conv_1_bn[0][0]']
Total params: 2257984 (8.61 MB)
Trainable params: 0 (0.00 Byte)
Non-trainable params: 2257984 (8.61 MB)
17s completed at 11:23PM
82°F Partly cloudy
Search
ENG IN
11:33 PM 19-03-2024

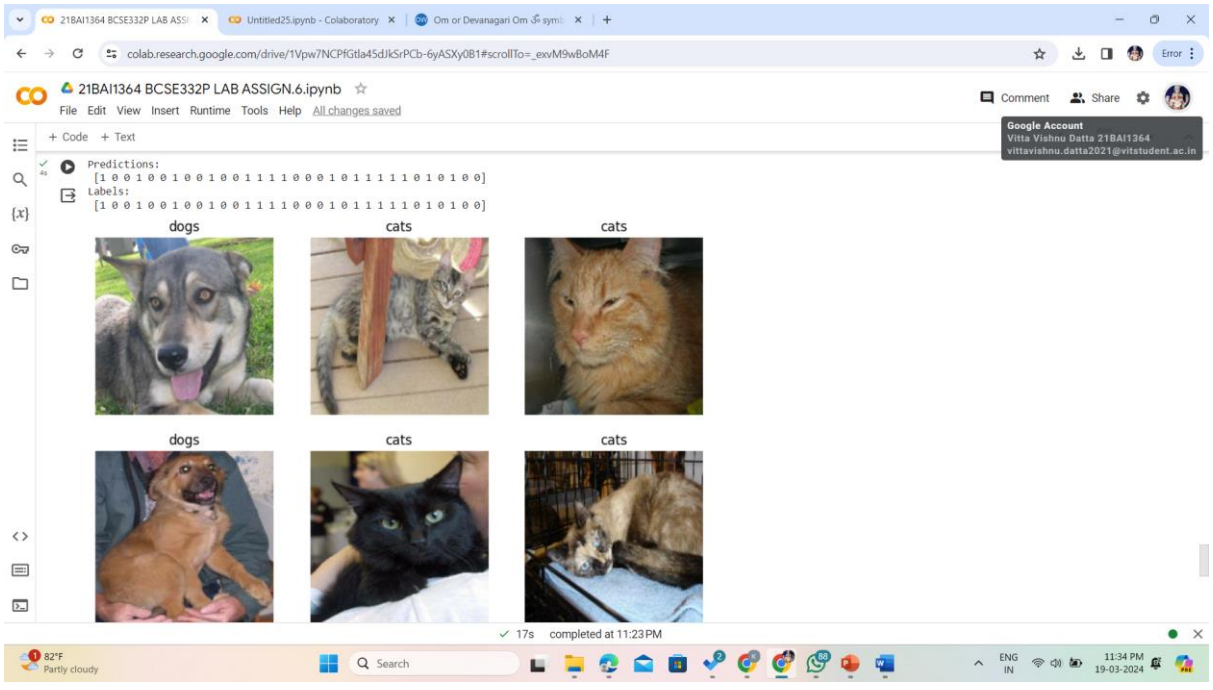

21BAI1364 BCSE332P LAB ASSIGN.6.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment Share
Google Account
Vitta Vishnu Datta 21BAI1364
vittavishnu.datta2021@vitstudent.ac.in
+ Code + Text
[25] global_average_layer = tf.keras.layers.GlobalAveragePooling2D()
feature_batch_average = global_average_layer(feature_batch)
print(feature_batch_average.shape)
(32, 1280)
[26] prediction_layer = tf.keras.layers.Dense(1)
prediction_batch = prediction_layer(feature_batch_average)
print(prediction_batch.shape)
(32, 1)
[27] inputs = tf.keras.Input(shape=(160, 160, 3))
x = data_augmentation(inputs)
x = preprocess_input(x)
x = base_model(x, training=False)
x = global_average_layer(x)
x = tf.keras.layers.Dropout(0.2)(x)
outputs = prediction_layer(x)
model = tf.keras.Model(inputs, outputs)
[28] base_learning_rate = 0.0001
model.compile(
optimizer=tf.keras.optimizers.Adam(learning_rate=base_learning_rate),
loss=tf.keras.losses.BinaryCrossentropy(from_logits=True),
metrics=["accuracy"],
)
17s completed at 11:23PM
82°F Partly cloudy
Search
ENG IN
11:34 PM 19-03-2024

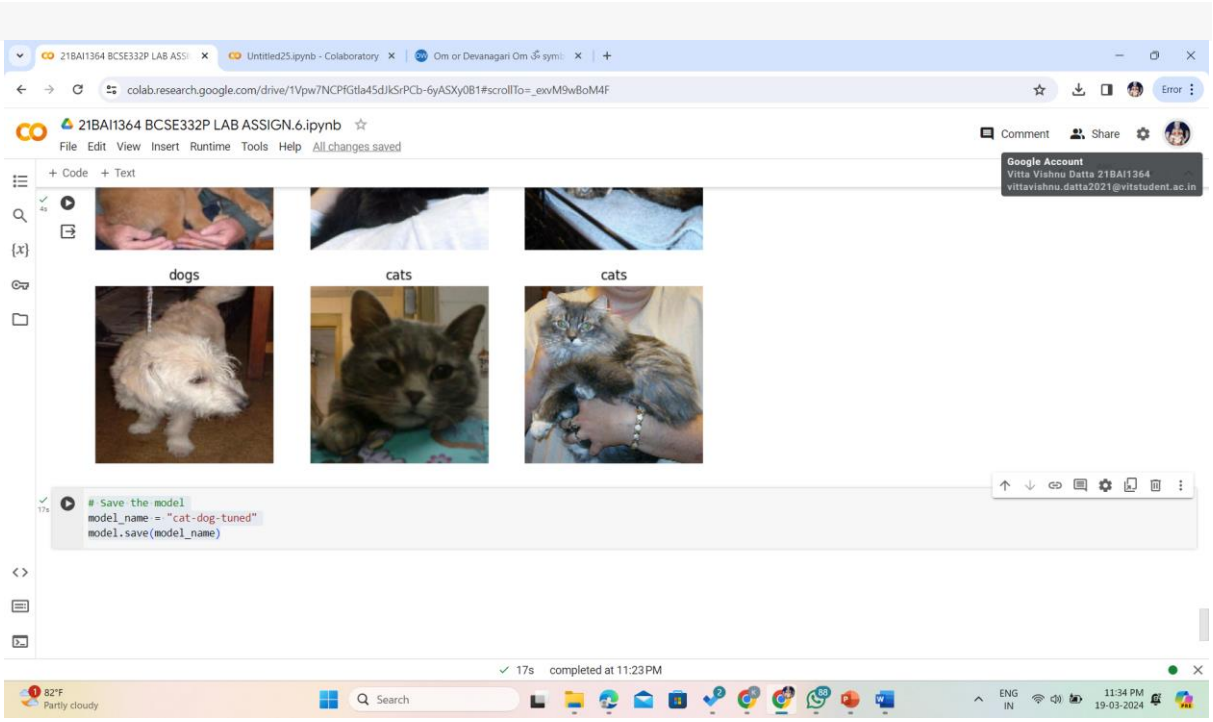

21BAI1364 BCSE332P LAB ASSIGN.6.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment Share
Google Account
Vitta Vishnu Datta 21BAI1364
vittavishnu.datta2021@vitstudent.ac.in
+ Code + Text
[29] len(model.trainable_variables)
2
[30] initial_epochs = 10
loss0, accuracy0 = model.evaluate(validation_dataset)
26/26 [==============================] - 4s 52ms/step - loss: 0.8174 - accuracy: 0.4332
[31] print("initial loss: {:.2f}".format(loss0))
print("initial accuracy: {:.2f}".format(accuracy0))
initial loss: 0.82
initial accuracy: 0.43
[32] history = model.fit(
train_dataset, epochs=initial_epochs, validation_data=validation_dataset
)
Epoch 1/10
63/63 [==============================] - 11s 94ms/step - loss: 0.7200 - accuracy: 0.5725 - val_loss: 0.5437 - val_accuracy: 0.6510
Epoch 2/10
63/63 [==============================] - 4s 53ms/step - loss: 0.5401 - accuracy: 0.7060 - val_loss: 0.3999 - val_accuracy: 0.7636
Epoch 3/10
63/63 [==============================] - 4s 60ms/step - loss: 0.4165 - accuracy: 0.7890 - val_loss: 0.3043 - val_accuracy: 0.8515
Epoch 4/10
63/63 [==============================] - 3s 51ms/step - loss: 0.3553 - accuracy: 0.8265 - val_loss: 0.2586 - val_accuracy: 0.8738
Epoch 5/10
63/63 [==============================] - 4s 61ms/step - loss: 0.3113 - accuracy: 0.8570 - val_loss: 0.2074 - val_accuracy: 0.9183
17s completed at 11:23PM
82°F Partly cloudy
Search
ENG IN
11:34 PM 19-03-2024


21BAI1364 BCSE332P LAB ASSIGN.6.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment
Share
Google Account
Vitta Vishnu Datta 21BAI1364
vittavishnu.datta2021@vitstudent.ac.in
+ Code + Text
Epoch 5/10
63/63 [==========] - 4s 61ms/step - loss: 0.3113 - accuracy: 0.8570 - val_loss: 0.2074 - val_accuracy: 0.9183
Epoch 6/10
63/63 [==========] - 3s 50ms/step - loss: 0.2756 - accuracy: 0.8715 - val_loss: 0.1902 - val_accuracy: 0.9171
Epoch 7/10
63/63 [==========] - 3s 50ms/step - loss: 0.2620 - accuracy: 0.8810 - val_loss: 0.1688 - val_accuracy: 0.9245
Epoch 8/10
63/63 [==========] - 4s 66ms/step - loss: 0.2448 - accuracy: 0.8915 - val_loss: 0.1500 - val_accuracy: 0.9307
Epoch 9/10
63/63 [==========] - 3s 51ms/step - loss: 0.2287 - accuracy: 0.9020 - val_loss: 0.1358 - val_accuracy: 0.9443
Epoch 10/10
63/63 [==========] - 3s 49ms/step - loss: 0.2180 - accuracy: 0.9070 - val_loss: 0.1287 - val_accuracy: 0.9431
[33] base_model.trainable = True
[34] model.compile(
loss=tf.keras.losses.BinaryCrossentropy(from_logits=True),
optimizer=tf.keras.optimizers.RMSprop(learning_rate=base_learning_rate / 10),
metrics=["accuracy"],
)
[35] len(model.trainable_variables)
158
[36] fine_tune_epochs = 10
total_epochs = initial_epochs + fine_tune_epochs
history_fine = model.fit(
train_dataset,
17s completed at 11:23PM
82°F Partly cloudy
Search
ENG IN
11:34 PM 19-03-2024


21BAI1364 BCSE332P LAB ASSIGN.6.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment
Share
Google Account
Vitta Vishnu Datta 21BAI1364
vittavishnu.datta2021@vitstudent.ac.in
+ Code + Text
[36] fine_tune_epochs = 10
total_epochs = initial_epochs + fine_tune_epochs
history_fine = model.fit(
train_dataset,
epochs=total_epochs,
initial_epoch=history.epoch[-1],
validation_data=validation_dataset,
)
Epoch 10/20
63/63 [==========] - 28s 149ms/step - loss: 0.1418 - accuracy: 0.9375 - val_loss: 0.0512 - val_accuracy: 0.9876
Epoch 11/20
63/63 [==========] - 7s 101ms/step - loss: 0.1161 - accuracy: 0.9515 - val_loss: 0.0433 - val_accuracy: 0.9839
Epoch 12/20
63/63 [==========] - 7s 102ms/step - loss: 0.0932 - accuracy: 0.9660 - val_loss: 0.0533 - val_accuracy: 0.9839
Epoch 13/20
63/63 [==========] - 7s 108ms/step - loss: 0.0884 - accuracy: 0.9685 - val_loss: 0.0376 - val_accuracy: 0.9864
Epoch 14/20
63/63 [==========] - 7s 108ms/step - loss: 0.0743 - accuracy: 0.9695 - val_loss: 0.0445 - val_accuracy: 0.9839
Epoch 15/20
63/63 [==========] - 7s 101ms/step - loss: 0.0701 - accuracy: 0.9740 - val_loss: 0.0578 - val_accuracy: 0.9740
Epoch 16/20
63/63 [==========] - 7s 107ms/step - loss: 0.0594 - accuracy: 0.9705 - val_loss: 0.0397 - val_accuracy: 0.9827
Epoch 17/20
63/63 [==========] - 7s 113ms/step - loss: 0.0825 - accuracy: 0.9690 - val_loss: 0.0414 - val_accuracy: 0.9839
Epoch 18/20
63/63 [==========] - 7s 102ms/step - loss: 0.0572 - accuracy: 0.9765 - val_loss: 0.0420 - val_accuracy: 0.9827
Epoch 19/20
63/63 [==========] - 7s 108ms/step - loss: 0.0556 - accuracy: 0.9770 - val_loss: 0.0831 - val_accuracy: 0.9790
Epoch 20/20
63/63 [==========] - 7s 113ms/step - loss: 0.0427 - accuracy: 0.9845 - val_loss: 0.0346 - val_accuracy: 0.9827
17s completed at 11:23PM
82°F Partly cloudy
Search
ENG IN
11:34 PM 19-03-2024


21BAI1364 BCSE332P LAB ASSIGN.6.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment Share
Google Account
Vitta Vishnu Datta 21BAI1364
vittavishnu.datta2021@vitstudent.ac.in
+ Code + Text
[37] # Retrieve a batch of images from the test set
image_batch, label_batch = test_dataset.as_numpy_iterator().next()
predictions = model.predict_on_batch(image_batch).flatten()
# Apply a sigmoid since our model returns logits
predictions = tf.nn.sigmoid(predictions)
predictions = tf.where(predictions < 0.5, 0, 1)
print("Predictions:\n", predictions.numpy())
print("Labels:\n", label_batch)
plt.figure(figsize=(10, 10))
for i in range(9):
ax = plt.subplot(3, 3, i + 1)
plt.imshow(image_batch[i].astype("uint8"))
plt.title(class_names[predictions[i]])
plt.axis("off")
Predictions:
[1 0 0 1 0 0 1 0 0 1 0 0 1 1 1 1 0 0 0 1 0 1 1 1 1 1 0 1 0 1 0 0]
Labels:
[1 0 0 1 0 0 1 0 0 1 0 0 1 1 1 1 0 0 0 1 0 1 1 1 1 1 0 1 0 1 0 0]
dogs
cats
cats
17s completed at 11:23 PM
82°F Partly cloudy
Search
ENG IN
11:34 PM 19-03-2024


21BAI1364 BCSE332P LAB ASSIGN.6.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment Share
Google Account
Vitta Vishnu Datta 21BAI1364
vittavishnu.datta2021@vitstudent.ac.in
+ Code + Text
Predictions:
[1 0 0 1 0 0 1 0 0 1 0 0 1 1 1 1 0 0 0 1 0 1 1 1 1 1 0 1 0 1 0 0]
Labels:
[1 0 0 1 0 0 1 0 0 1 0 0 1 1 1 1 0 0 0 1 0 1 1 1 1 1 0 1 0 1 0 0]
dogs
cats
cats
dogs
cats
cats
17s completed at 11:23 PM
82°F Partly cloudy
Search
ENG IN
11:34 PM 19-03-2024



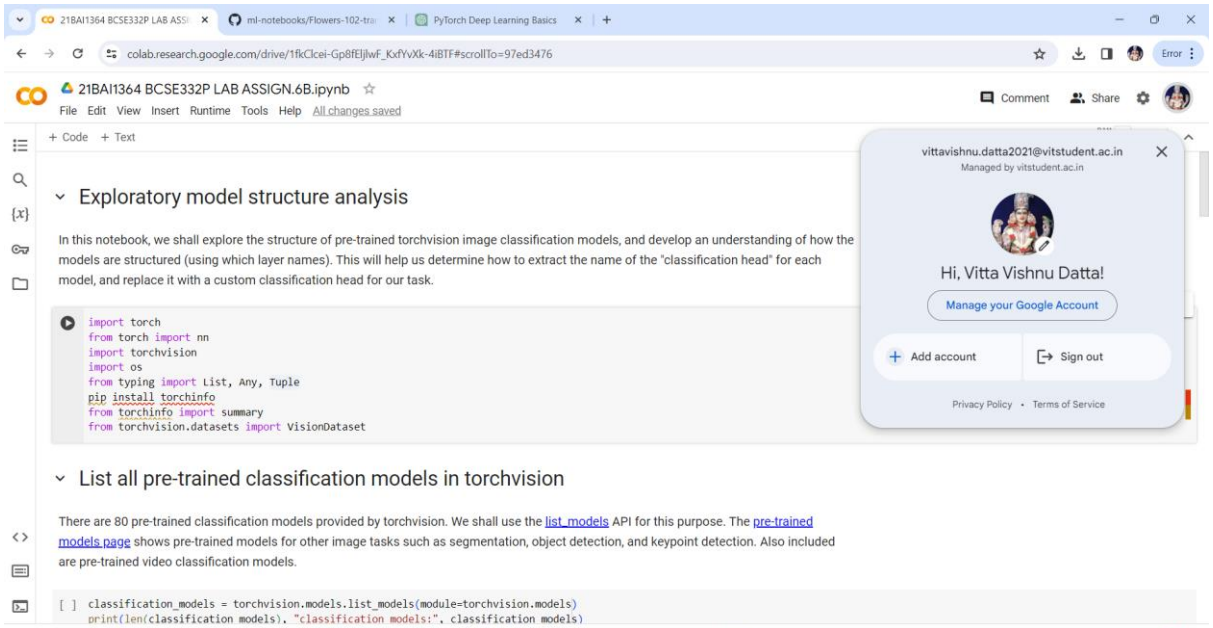
Exploratory model structure analysis

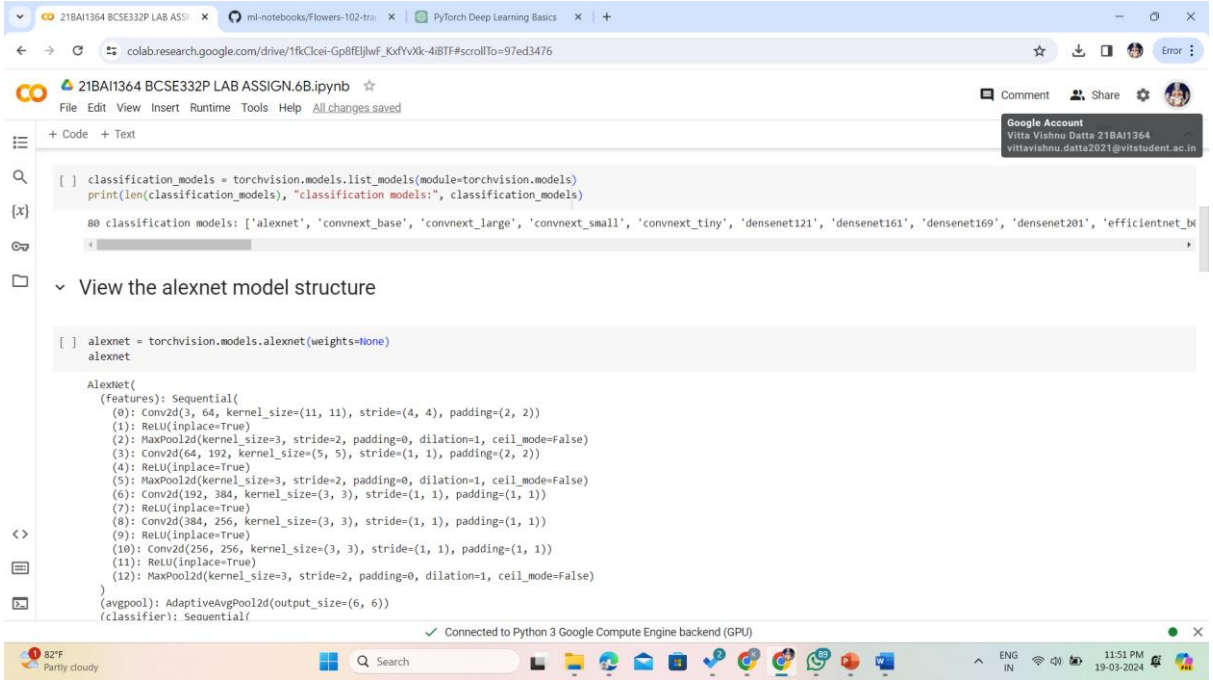
```
import torch
from torch import nn
import torchvision
import os
from typing import List, Any, Tuple
pip install torchinfo
from torchinfo import summary
from torchvision.datasets import VisionDataset
```

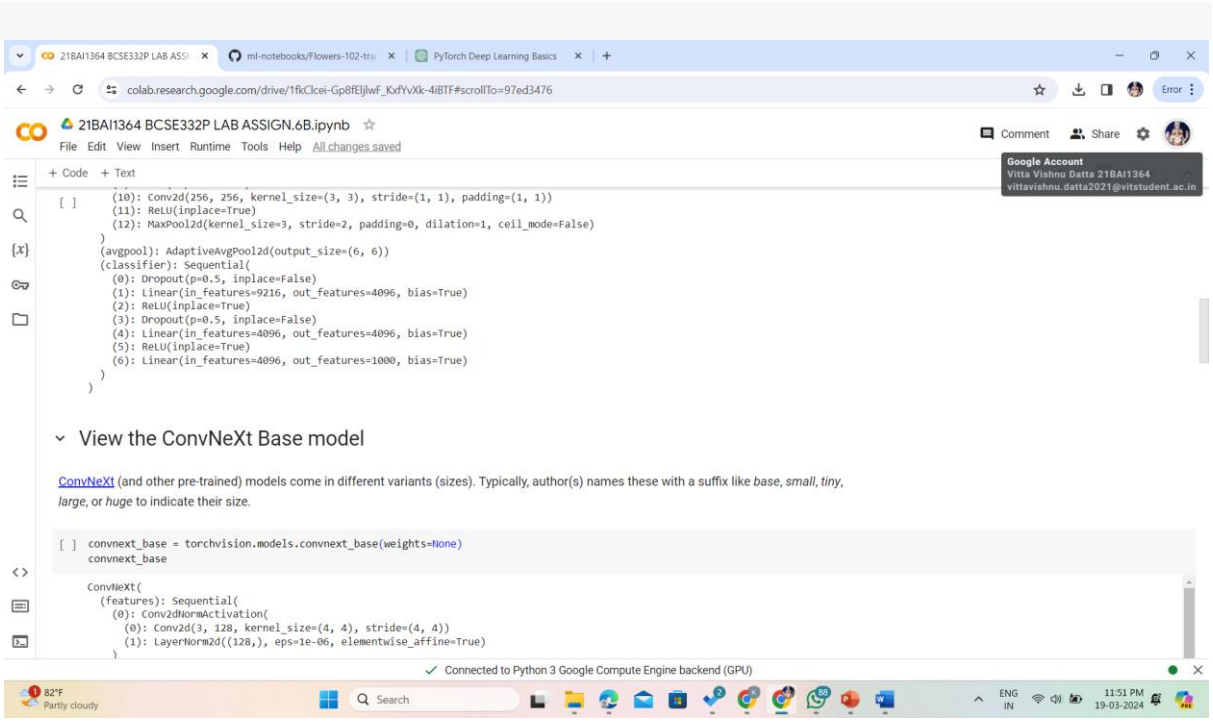
```
classification_models =
torchvision.models.list_models(module=torchvision.models)
print(len(classification_models), "classification models:",
classification_models)
```

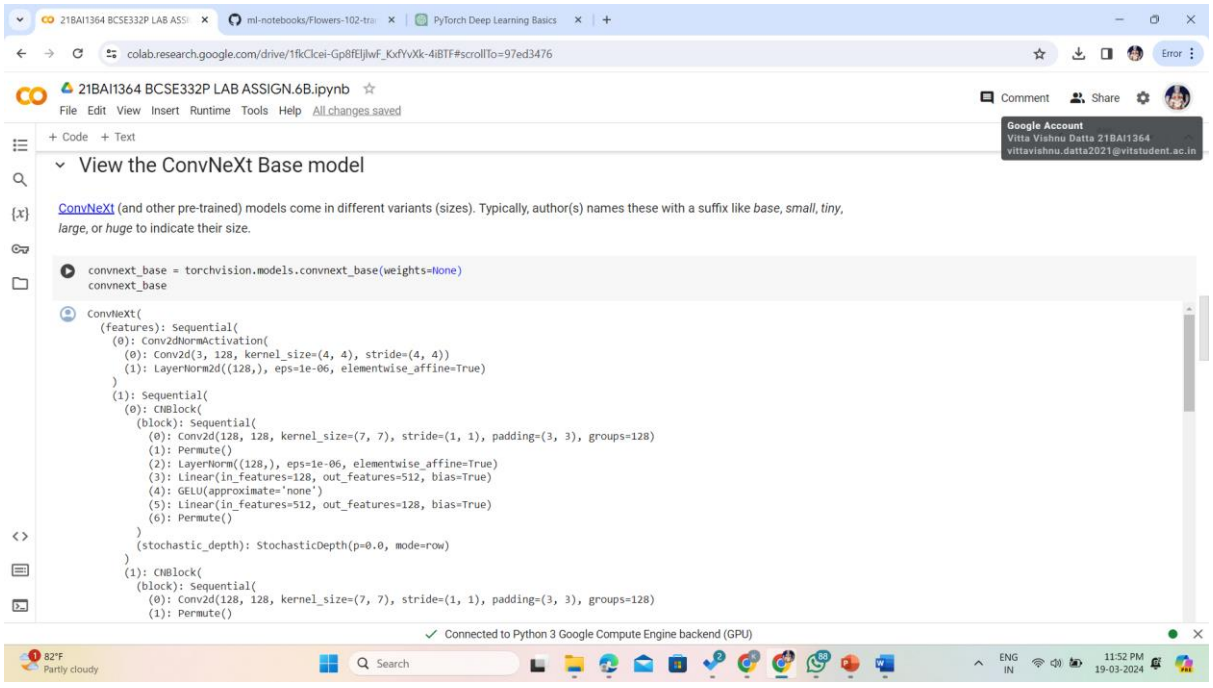
```
alexnet = torchvision.models.alexnet(weights=None)
alexnet
convnext_base = torchvision.models.convnext_base(weights=None)
convnext_base
dict(alexnet.named_children()).keys()
layer_count = dict()

for model_name in classification_models:
    m = getattr(torchvision.models, model_name)(weights=None)
    layer_names = dict(m.named_children()).keys()
    for ln in layer_names:
```

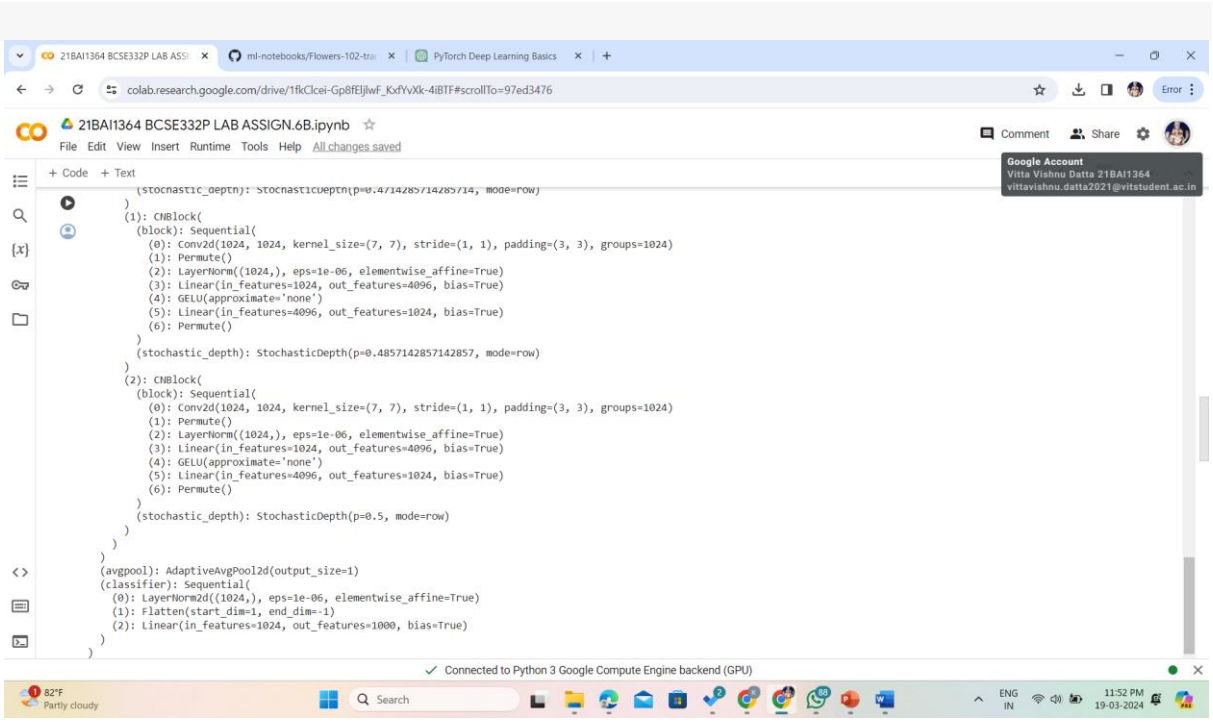
```
            layer_count[ln] = layer_count.get(ln, 0) + 1
        # end for
    # end for

print(layer_count)
vgg16 = torchvision.models.vgg16_bn(weights=None)
resnet50 = torchvision.models.resnet50(weights=None)
resnet152 = torchvision.models.resnet152(weights=None)
print("vgg16\n", vgg16.classifier)
print("resnet50\n", resnet50.fc)
print("resnet152\n", resnet152.fc)
```

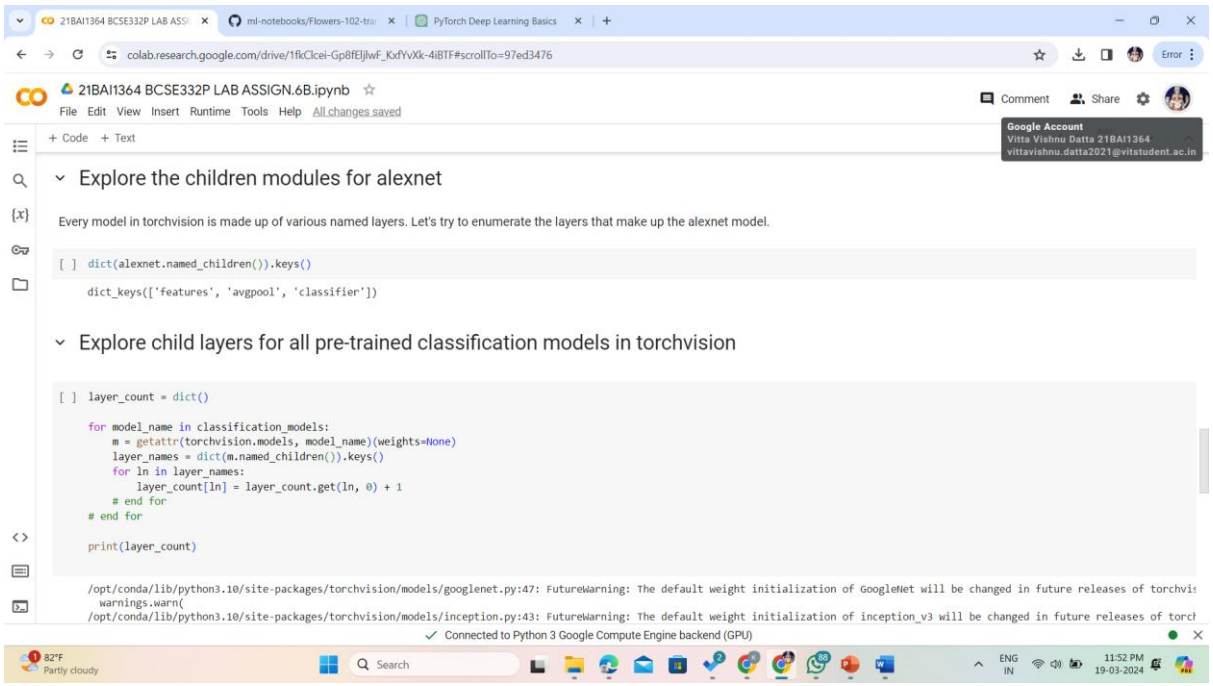


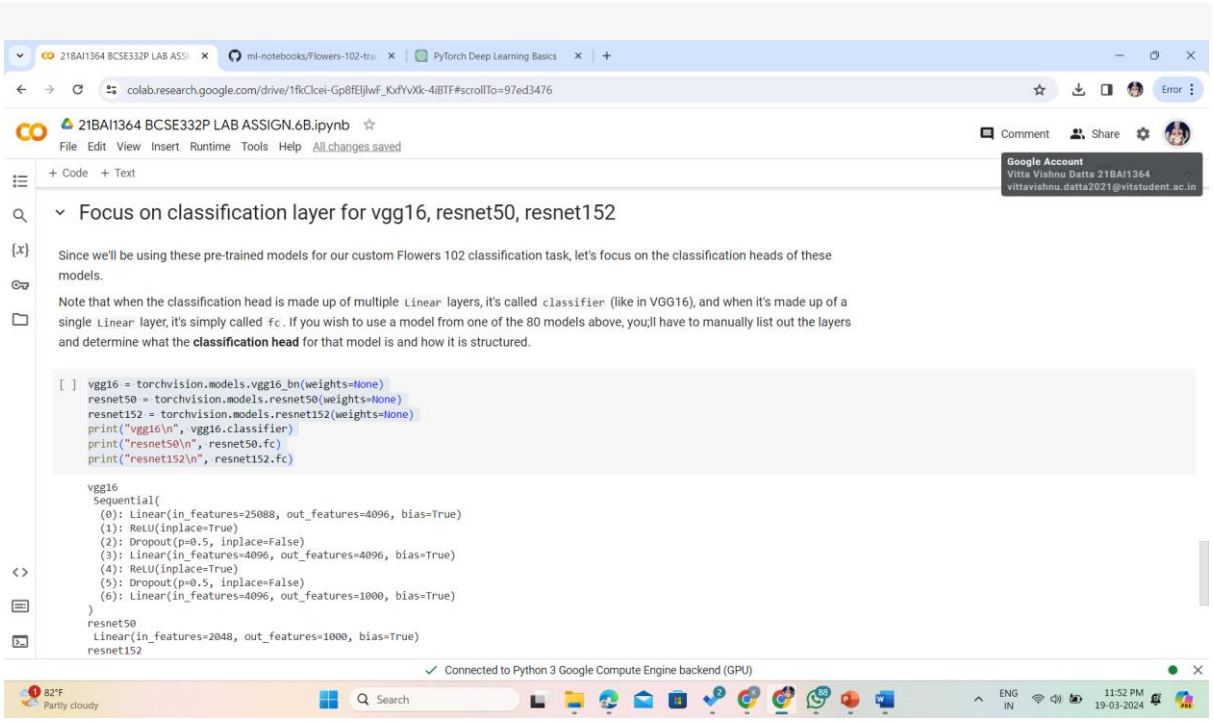


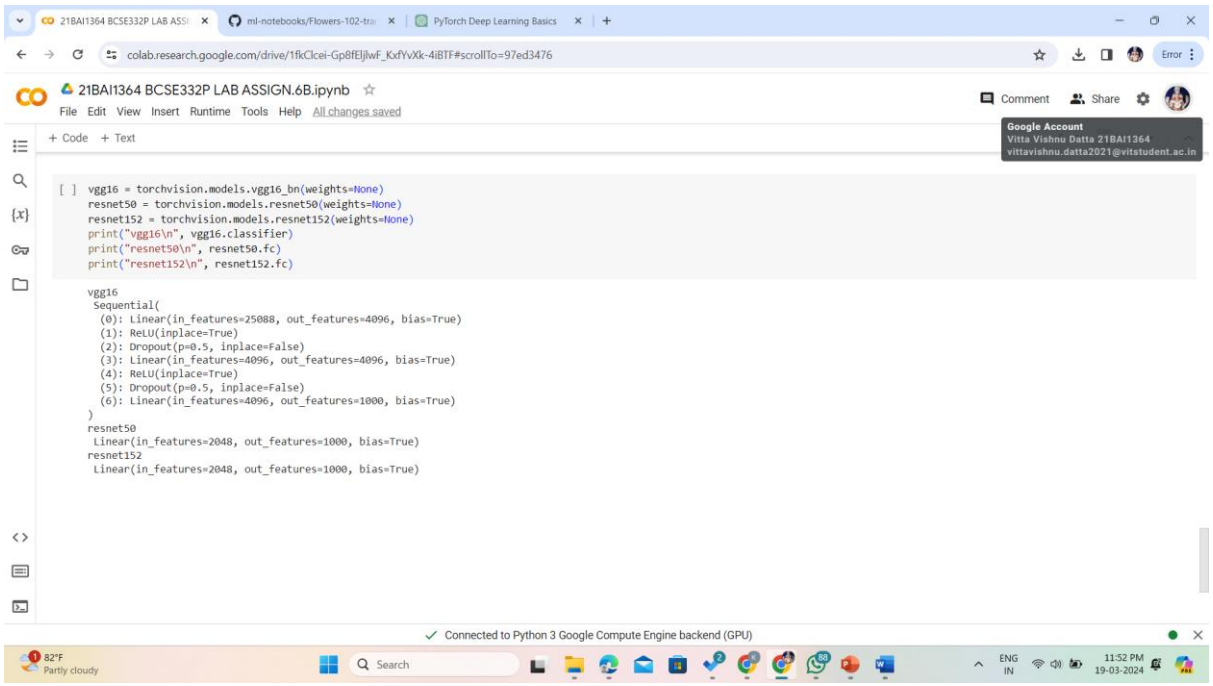

21BAI1364 BCSE332P LAB ASSIGN.6B.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment Share
Google Account
Vitta Vishnu Datta 21BAI1364
vittavishnu.datta2021@vitstudent.ac.in
+ Code + Text
(10): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
(11): ReLU(inplace=True)
(12): MaxPool2d(kernel_size=3, stride=2, padding=0, dilation=1, ceil_mode=False)
(avgpool): AdaptiveAvgPool2d(output_size=(6, 6))
(classifier): Sequential(
(0): Dropout(p=0.5, inplace=False)
(1): Linear(in_features=9216, out_features=4096, bias=True)
(2): ReLU(inplace=True)
(3): Dropout(p=0.5, inplace=False)
(4): Linear(in_features=4096, out_features=4096, bias=True)
(5): ReLU(inplace=True)
(6): Linear(in_features=4096, out_features=1000, bias=True)
View the ConvNeXt Base model
ConvNeXt (and other pre-trained) models come in different variants (sizes). Typically, author(s) names these with a suffix like base, small, tiny, large, or huge to indicate their size.
convnext_base = torchvision.models.convnext_base(weights=None)
convnext_base
ConvNeXt(
(features): Sequential(
(0): Conv2dNormActivation(
(0): Conv2d(3, 128, kernel_size=(4, 4), stride=(4, 4))
(1): LayerNorm2d((128,), eps=1e-06, elementwise_affine=True)
Connected to Python 3 Google Compute Engine backend (GPU)
82°F Partly cloudy
Search
ENG IN
11:51 PM 19-03-2024


21BAI1364 BCSE332P LAB ASSIGN.6B.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment Share
Google Account
Vitta Vishnu Datta 21BAI1364
vittavishnu.datta2021@vitstudent.ac.in
+ Code + Text
View the ConvNeXt Base model
ConvNeXt (and other pre-trained) models come in different variants (sizes). Typically, author(s) names these with a suffix like base, small, tiny, large, or huge to indicate their size.
convnext_base = torchvision.models.convnext_base(weights=None)
convnext_base
ConvNeXt(
(features): Sequential(
(0): Conv2dNormActivation(
(0): Conv2d(3, 128, kernel_size=(4, 4), stride=(4, 4))
(1): LayerNorm2d((128,), eps=1e-06, elementwise_affine=True)
(1): Sequential(
(0): CNBlock(
(block): Sequential(
(0): Conv2d(128, 128, kernel_size=(7, 7), stride=(1, 1), padding=(3, 3), groups=128)
(1): Permute()
(2): LayerNorm((128,), eps=1e-06, elementwise_affine=True)
(3): Linear(in_features=128, out_features=512, bias=True)
(4): GELU(approximate='none')
(5): Linear(in_features=512, out_features=128, bias=True)
(6): Permute()
(stochastic_depth): StochasticDepth(p=0.0, mode=row)
(1): CNBlock(
(block): Sequential(
(0): Conv2d(128, 128, kernel_size=(7, 7), stride=(1, 1), padding=(3, 3), groups=128)
(1): Permute()
Connected to Python 3 Google Compute Engine backend (GPU)
82°F Partly cloudy
Search
ENG IN
11:52 PM 19-03-2024


21BAI1364 BCSE332P LAB ASSIGN.6B.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment
Share
Google Account
Vitta Vishnu Datta 21BAI1364
vittavishnu.datta2021@vitstudent.ac.in
+ Code + Text
(stochastic_depth): StochasticDepth(p=0.4714285714285714, mode=row)
)
(1): CNBlock(
(block): Sequential(
(0): Conv2d(1024, 1024, kernel_size=(7, 7), stride=(1, 1), padding=(3, 3), groups=1024)
(1): Permute()
(2): LayerNorm((1024,), eps=1e-06, elementwise_affine=True)
(3): Linear(in_features=1024, out_features=4096, bias=True)
(4): GELU(approximate='none')
(5): Linear(in_features=4096, out_features=1024, bias=True)
(6): Permute()
)
(stochastic_depth): StochasticDepth(p=0.4857142857142857, mode=row)
)
(2): CNBlock(
(block): Sequential(
(0): Conv2d(1024, 1024, kernel_size=(7, 7), stride=(1, 1), padding=(3, 3), groups=1024)
(1): Permute()
(2): LayerNorm((1024,), eps=1e-06, elementwise_affine=True)
(3): Linear(in_features=1024, out_features=4096, bias=True)
(4): GELU(approximate='none')
(5): Linear(in_features=4096, out_features=1024, bias=True)
(6): Permute()
)
(stochastic_depth): StochasticDepth(p=0.5, mode=row)
)
)
)
(avgpool): AdaptiveAvgPool2d(output_size=1)
(classifier): Sequential(
(0): LayerNorm2d((1024,), eps=1e-06, elementwise_affine=True)
(1): Flatten(start_dim=1, end_dim=-1)
(2): Linear(in_features=1024, out_features=1000, bias=True)
)
)
Connected to Python 3 Google Compute Engine backend (GPU)
82°F Partly cloudy
Search
ENG IN
11:52 PM 19-03-2024


21BAI1364 BCSE332P LAB ASSIGN.6B.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment
Share
Google Account
Vitta Vishnu Datta 21BAI1364
vittavishnu.datta2021@vitstudent.ac.in
+ Code + Text
Explore the children modules for alexnet
Every model in torchvision is made up of various named layers. Let's try to enumerate the layers that make up the alexnet model.
[ ] dict(alexnet.named_children()).keys()
dict_keys(['features', 'avgpool', 'classifier'])
Explore child layers for all pre-trained classification models in torchvision
[ ] layer_count = dict()
for model_name in classification_models:
m = getattr(torchvision.models, model_name)(weights=None)
layer_names = dict(m.named_children()).keys()
for ln in layer_names:
layer_count[ln] = layer_count.get(ln, 0) + 1
# end for
# end for
print(layer_count)
/opt/conda/lib/python3.10/site-packages/torchvision/models/googlenet.py:47: FutureWarning: The default weight initialization of GoogleNet will be changed in future releases of torchvis
warnings.warn(
/opt/conda/lib/python3.10/site-packages/torchvision/models/inception.py:43: FutureWarning: The default weight initialization of inception_v3 will be changed in future releases of torch
Connected to Python 3 Google Compute Engine backend (GPU)
82°F Partly cloudy
Search
ENG IN
11:52 PM 19-03-2024


21BAI1364 BCSE332P LAB ASSIGN.6B.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment
Share
Google Account
Vitta Vishnu Datta 21BAI1364
vittavishnu.datta2021@vitstudent.ac.in
+ Code + Text
Focus on classification layer for vgg16, resnet50, resnet152
Since we'll be using these pre-trained models for our custom Flowers 102 classification task, let's focus on the classification heads of these models.
Note that when the classification head is made up of multiple Linear layers, it's called classifier (like in VGG16), and when it's made up of a single Linear layer, it's simply called fc. If you wish to use a model from one of the 80 models above, you;ll have to manually list out the layers and determine what the classification head for that model is and how it is structured.
[ ] vgg16 = torchvision.models.vgg16_bn(weights=None)
resnet50 = torchvision.models.resnet50(weights=None)
resnet152 = torchvision.models.resnet152(weights=None)
print("vgg16\n", vgg16.classifier)
print("resnet50\n", resnet50.fc)
print("resnet152\n", resnet152.fc)
vgg16
Sequential(
(0): Linear(in_features=25088, out_features=4096, bias=True)
(1): ReLU(inplace=True)
(2): Dropout(p=0.5, inplace=False)
(3): Linear(in_features=4096, out_features=4096, bias=True)
(4): ReLU(inplace=True)
(5): Dropout(p=0.5, inplace=False)
(6): Linear(in_features=4096, out_features=1000, bias=True)
)
resnet50
Linear(in_features=2048, out_features=1000, bias=True)
resnet152
Connected to Python 3 Google Compute Engine backend (GPU)
82°F
Partly cloudy
Search
ENG
IN
11:52 PM
19-03-2024

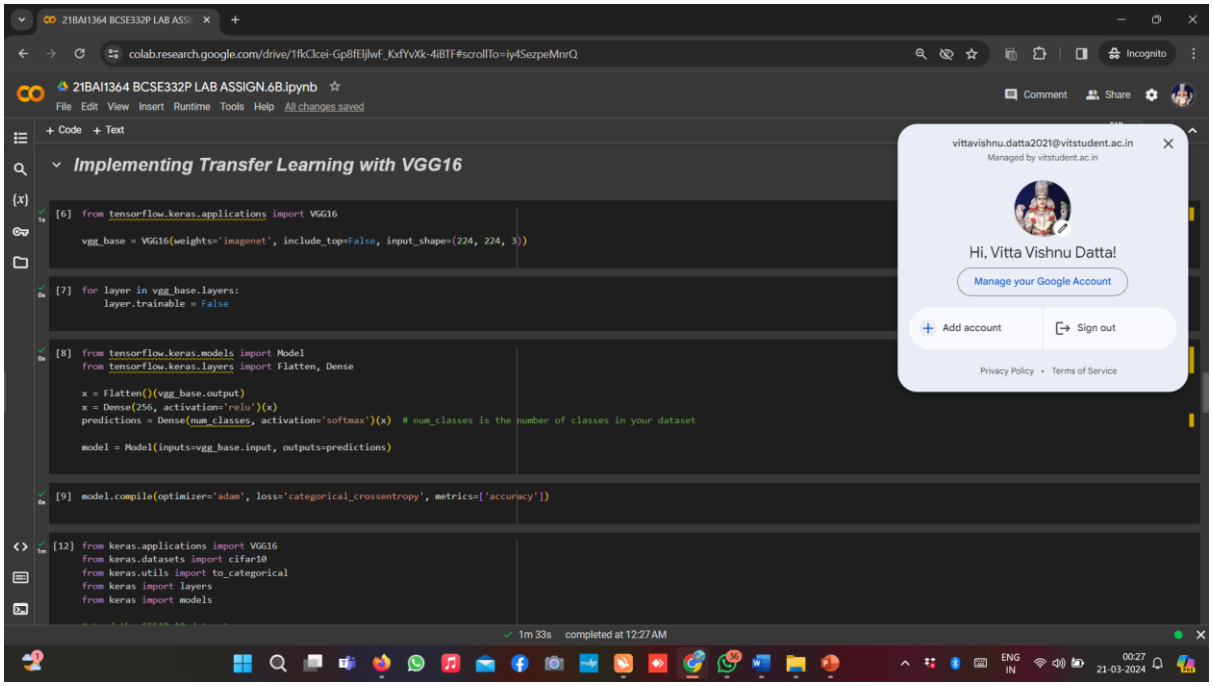

21BAI1364 BCSE332P LAB ASSIGN.6B.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment
Share
Google Account
Vitta Vishnu Datta 21BAI1364
vittavishnu.datta2021@vitstudent.ac.in
+ Code + Text
[ ] vgg16 = torchvision.models.vgg16_bn(weights=None)
resnet50 = torchvision.models.resnet50(weights=None)
resnet152 = torchvision.models.resnet152(weights=None)
print("vgg16\n", vgg16.classifier)
print("resnet50\n", resnet50.fc)
print("resnet152\n", resnet152.fc)
vgg16
Sequential(
(0): Linear(in_features=25088, out_features=4096, bias=True)
(1): ReLU(inplace=True)
(2): Dropout(p=0.5, inplace=False)
(3): Linear(in_features=4096, out_features=4096, bias=True)
(4): ReLU(inplace=True)
(5): Dropout(p=0.5, inplace=False)
(6): Linear(in_features=4096, out_features=1000, bias=True)
)
resnet50
Linear(in_features=2048, out_features=1000, bias=True)
resnet152
Linear(in_features=2048, out_features=1000, bias=True)
Connected to Python 3 Google Compute Engine backend (GPU)
82°F
Partly cloudy
Search
ENG
IN
11:52 PM
19-03-2024

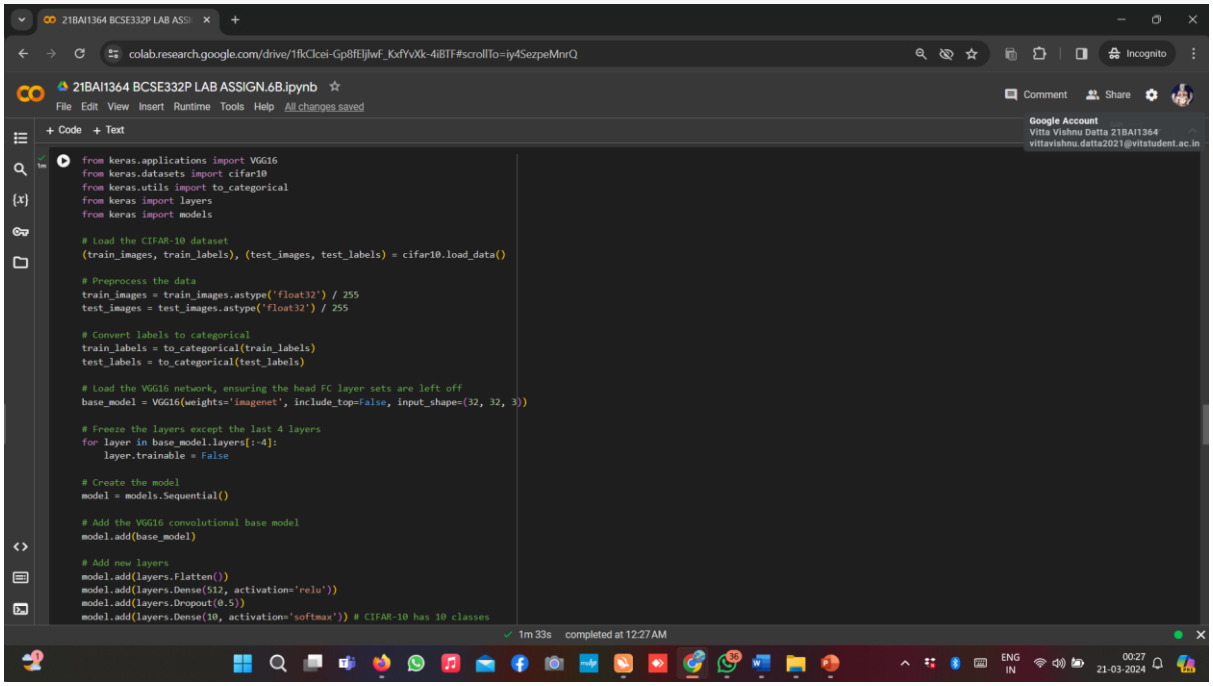
Implement transfer learning with vgg16, resnet50, resnet152

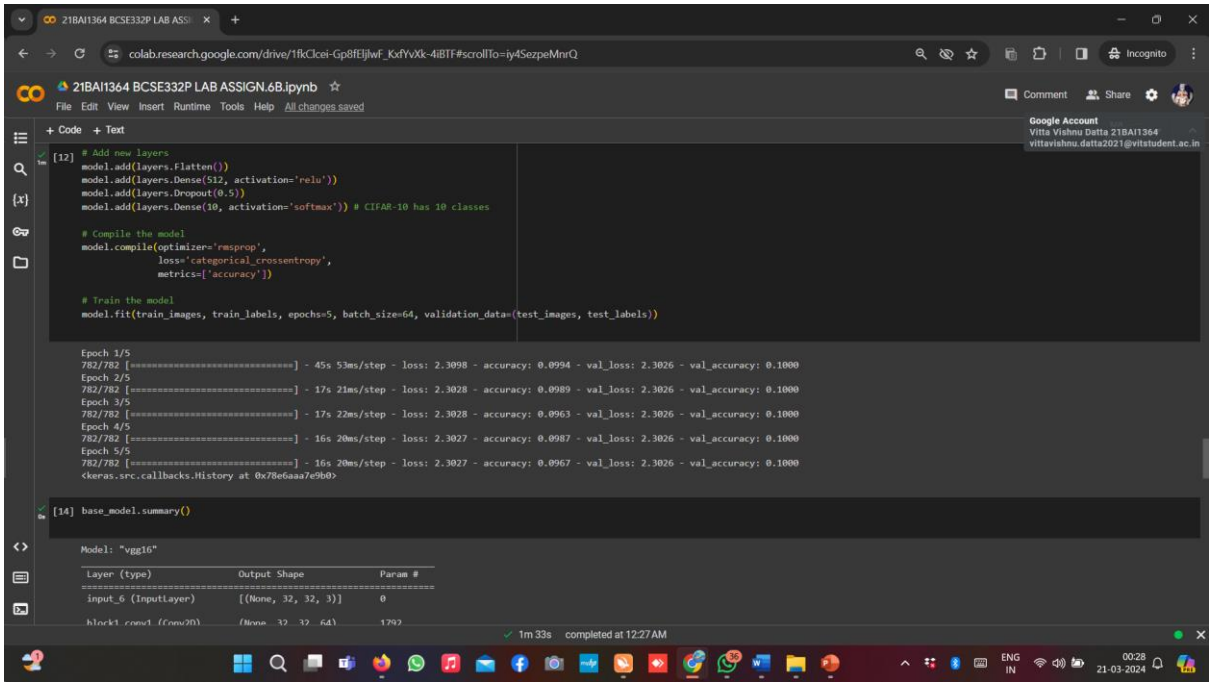
Implementing Transfer Learning with VGG16

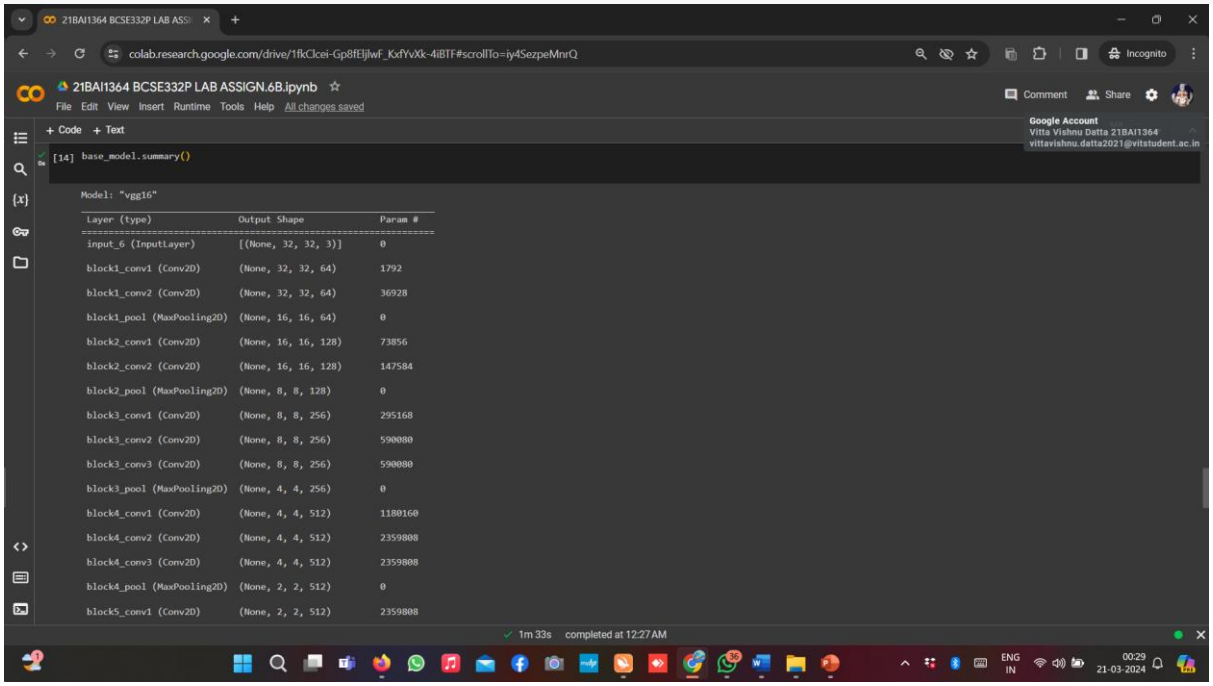
```
from tensorflow.keras.applications import VGG16

vgg_base = VGG16(weights='imagenet', include_top=False,
input_shape=(224, 224, 3))

for layer in vgg_base.layers:
    layer.trainable = False
from tensorflow.keras.models import Model
from tensorflow.keras.layers import Flatten, Dense

x = Flatten()(vgg_base.output)
x = Dense(256, activation='relu')(x)
predictions = Dense(num_classes, activation='softmax')(x)  #
num_classes is the number of classes in your dataset

model = Model(inputs=vgg_base.input, outputs=predictions)
model.compile(optimizer='adam', loss='categorical_crossentropy',
metrics=['accuracy'])
from keras.applications import VGG16
from keras.datasets import cifar10
from keras.utils import to_categorical
from keras import layers
from keras import models

# Load the CIFAR-10 dataset
(train_images, train_labels), (test_images, test_labels) =
cifar10.load_data()

# Preprocess the data
train_images = train_images.astype('float32') / 255
test_images = test_images.astype('float32') / 255

# Convert labels to categorical
train_labels = to_categorical(train_labels)
test_labels = to_categorical(test_labels)

# Load the VGG16 network, ensuring the head FC layer sets are left off
base_model = VGG16(weights='imagenet', include_top=False,
input_shape=(32, 32, 3))

# Freeze the layers except the last 4 layers
for layer in base_model.layers[:-4]:
```

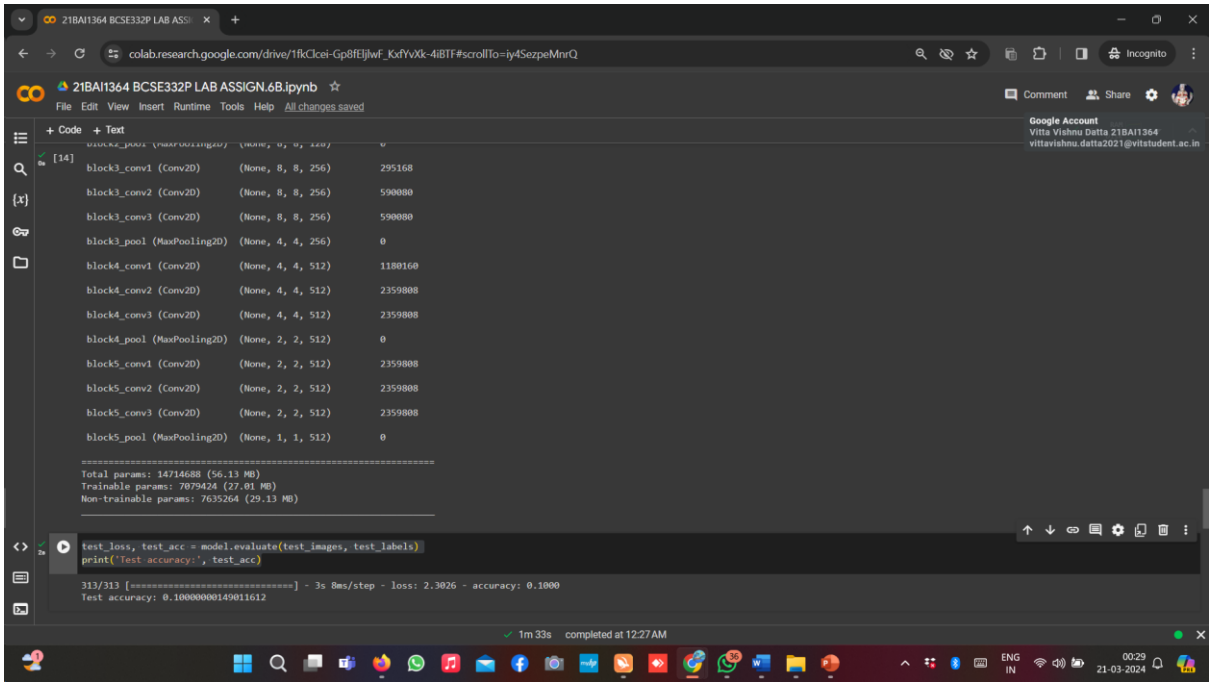
```
    layer.trainable = False

# Create the model
model = models.Sequential()

# Add the VGG16 convolutional base model
model.add(base_model)

# Add new layers
model.add(layers.Flatten())
model.add(layers.Dense(512, activation='relu'))
model.add(layers.Dropout(0.5))
model.add(layers.Dense(10, activation='softmax')) # CIFAR-10 has 10
classes

# Compile the model
model.compile(optimizer='rmsprop',
              loss='categorical_crossentropy',
              metrics=['accuracy'])

# Train the model
model.fit(train_images, train_labels, epochs=5, batch_size=64,
validation_data=(test_images, test_labels))

base_model.summary()

test_loss, test_acc = model.evaluate(test_images, test_labels)
print('Test accuracy:', test_acc)
```




21BAI1364 BCSE332P LAB ASSIGN.6B.ipynb
File Edit View Insert Runtime Tools Help All changes saved
+ Code + Text
from keras.applications import VGG16
from keras.datasets import cifar10
from keras.utils import to_categorical
from keras import layers
from keras import models
# Load the CIFAR-10 dataset
(train_images, train_labels), (test_images, test_labels) = cifar10.load_data()
# Preprocess the data
train_images = train_images.astype('float32') / 255
test_images = test_images.astype('float32') / 255
# Convert labels to categorical
train_labels = to_categorical(train_labels)
test_labels = to_categorical(test_labels)
# Load the VGG16 network, ensuring the head FC layer sets are left off
base_model = VGG16(weights='imagenet', include_top=False, input_shape=(32, 32, 3))
# Freeze the layers except the last 4 layers
for layer in base_model.layers[:-4]:
layer.trainable = False
# Create the model
model = models.Sequential()
# Add the VGG16 convolutional base model
model.add(base_model)
# Add new layers
model.add(layers.Flatten())
model.add(layers.Dense(512, activation='relu'))
model.add(layers.Dropout(0.5))
model.add(layers.Dense(10, activation='softmax')) # CIFAR-10 has 10 classes
1m 33s completed at 12:27 AM


21BAI1364 BCSE332P LAB ASSIGN.6B.ipynb
File Edit View Insert Runtime Tools Help All changes saved
+ Code + Text
[12] # Add new layers
model.add(layers.Flatten())
model.add(layers.Dense(512, activation='relu'))
model.add(layers.Dropout(0.5))
model.add(layers.Dense(10, activation='softmax')) # CIFAR-10 has 10 classes
Compile the model
model.compile(optimizer='rmsprop',
loss='categorical_crossentropy',
metrics=['accuracy'])
# Train the model
model.fit(train_images, train_labels, epochs=5, batch_size=64, validation_data=(test_images, test_labels))
Epoch 1/5
782/782 [==========] - 45s 53ms/step - loss: 2.3098 - accuracy: 0.0994 - val_loss: 2.3026 - val_accuracy: 0.1000
Epoch 2/5
782/782 [==========] - 17s 21ms/step - loss: 2.3028 - accuracy: 0.0989 - val_loss: 2.3026 - val_accuracy: 0.1000
Epoch 3/5
782/782 [==========] - 17s 22ms/step - loss: 2.3028 - accuracy: 0.0963 - val_loss: 2.3026 - val_accuracy: 0.1000
Epoch 4/5
782/782 [==========] - 16s 20ms/step - loss: 2.3027 - accuracy: 0.0987 - val_loss: 2.3026 - val_accuracy: 0.1000
Epoch 5/5
782/782 [==========] - 16s 20ms/step - loss: 2.3027 - accuracy: 0.0967 - val_loss: 2.3026 - val_accuracy: 0.1000
<keras.src.callbacks.History at 0x78e6aaa7e9b0>
[14] base_model.summary()
Model: "vgg16"
Layer (type) Output Shape Param #
input_6 (InputLayer) [(None, 32, 32, 3)] 0
1m 33s completed at 12:27 AM

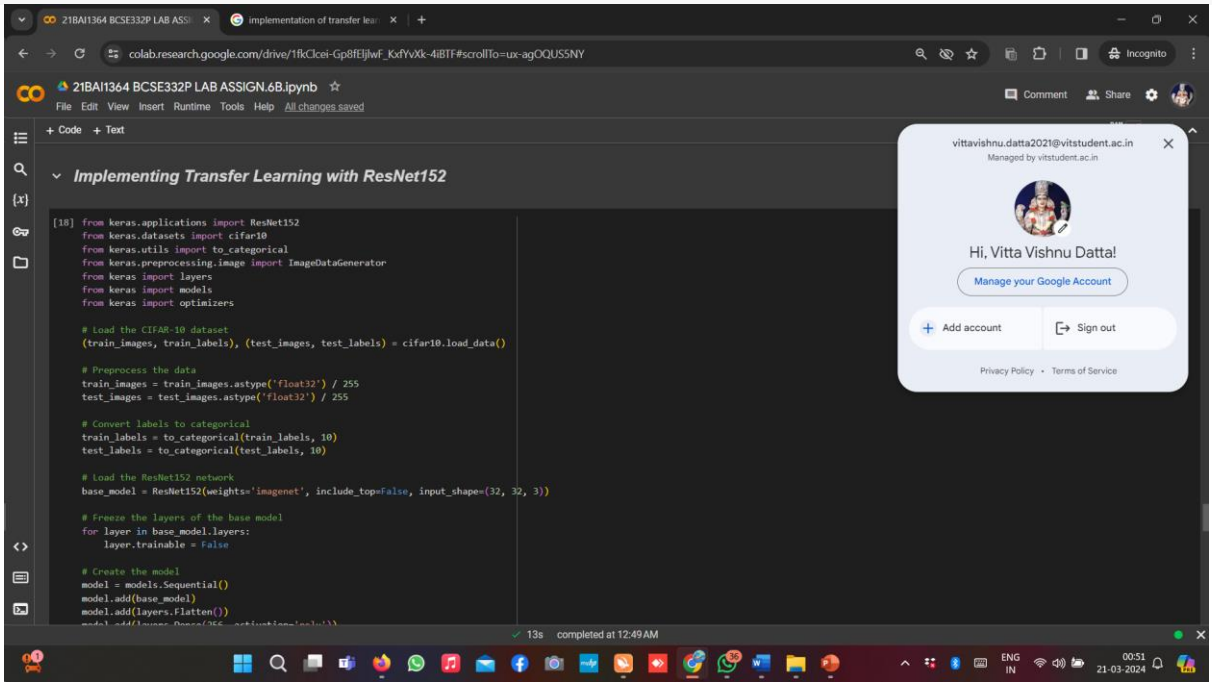


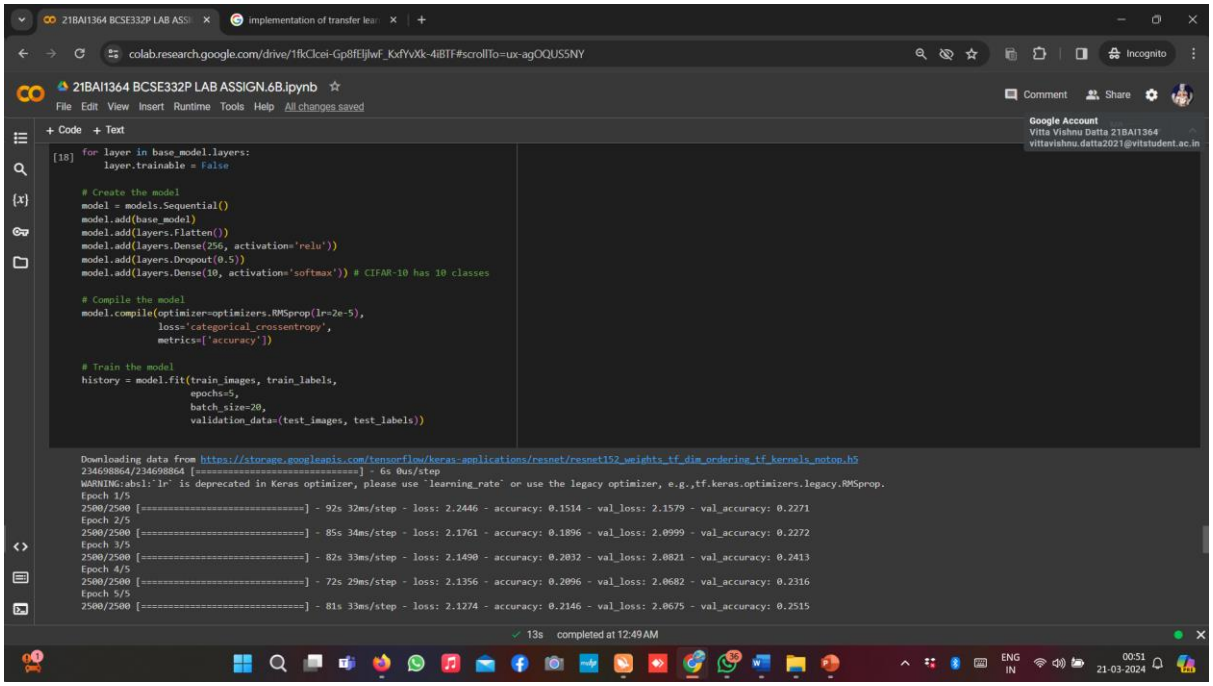


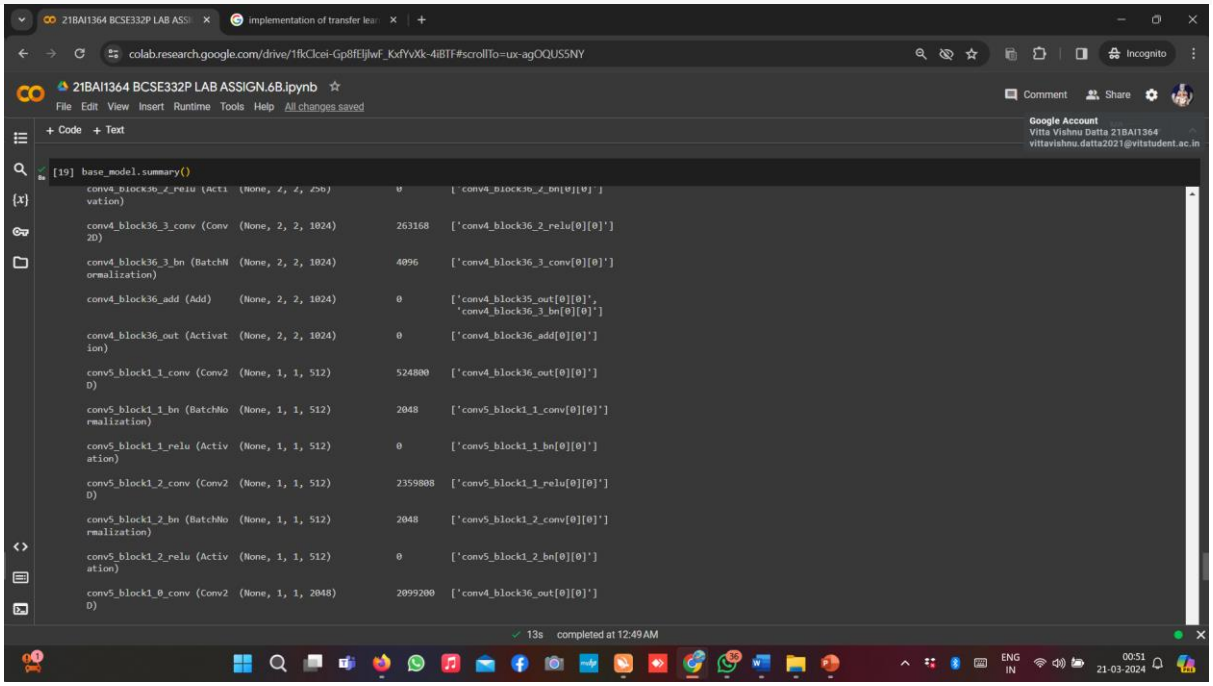
Implementing Transfer Learning with ResNet50

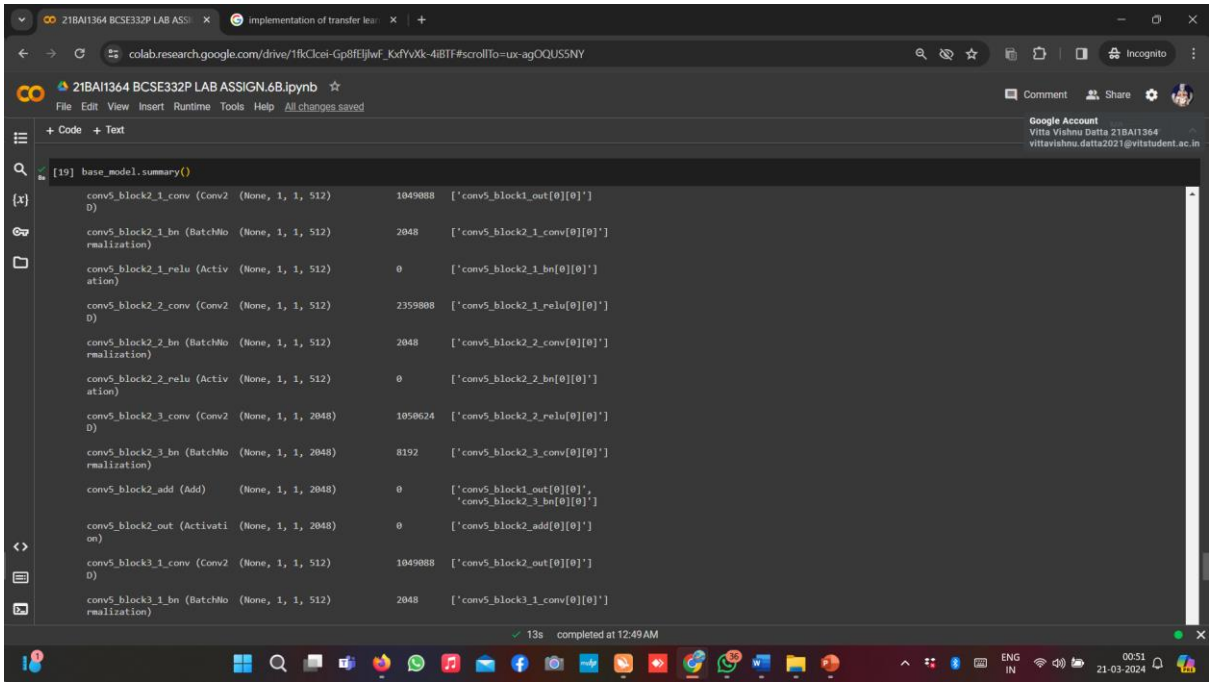
```
from keras.applications import ResNet152
from keras.datasets import cifar10
from keras.utils import to_categorical
from keras.preprocessing.image import ImageDataGenerator
from keras import layers
```

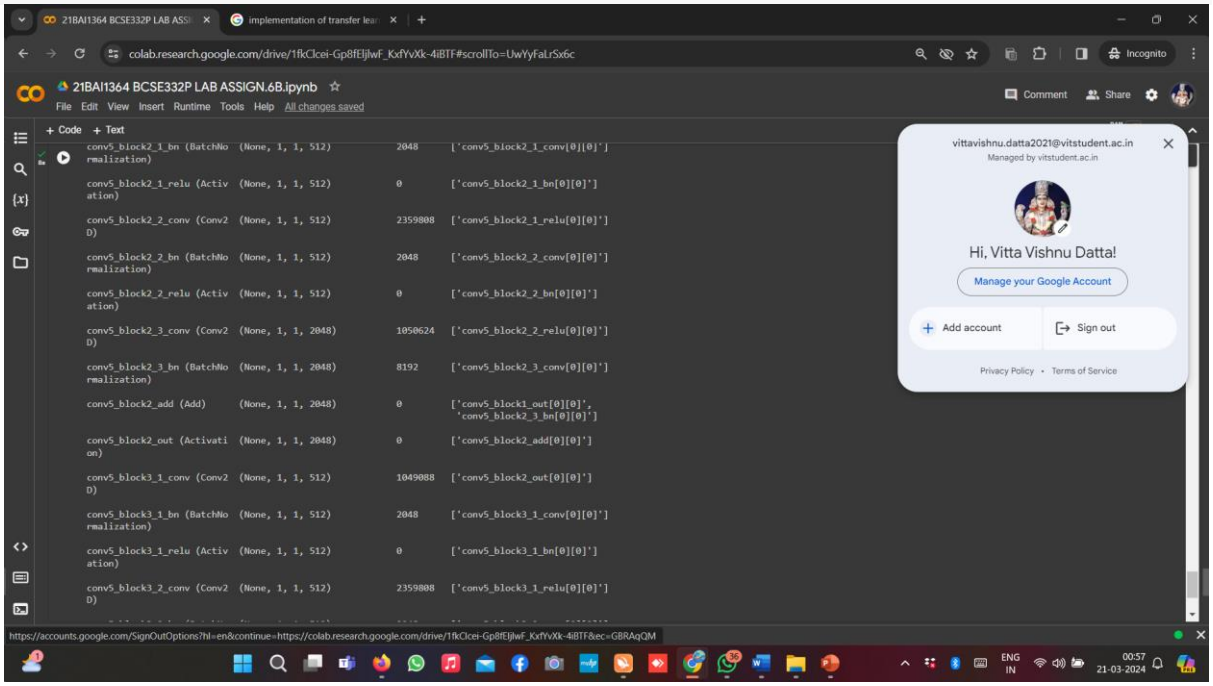
```
from keras import models
from keras import optimizers

# Load the CIFAR-10 dataset
(train_images, train_labels), (test_images, test_labels) =
cifar10.load_data()

# Preprocess the data
train_images = train_images.astype('float32') / 255
test_images = test_images.astype('float32') / 255

# Convert labels to categorical
train_labels = to_categorical(train_labels, 10)
test_labels = to_categorical(test_labels, 10)

# Load the ResNet152 network
base_model = ResNet152(weights='imagenet', include_top=False,
input_shape=(32, 32, 3))

# Freeze the layers of the base model
for layer in base_model.layers:
    layer.trainable = False

# Create the model
model = models.Sequential()
model.add(base_model)
model.add(layers.Flatten())
model.add(layers.Dense(256, activation='relu'))
model.add(layers.Dropout(0.5))
model.add(layers.Dense(10, activation='softmax')) # CIFAR-10 has 10
classes

# Compile the model
model.compile(optimizer=optimizers.RMSprop(lr=2e-5),
              loss='categorical_crossentropy',
              metrics=['accuracy'])

# Train the model
history = model.fit(train_images, train_labels,
                    epochs=5,
                    batch_size=20,
                    validation_data=(test_images, test_labels))

base_model.summary()
test_loss, test_acc = model.evaluate(test_images, test_labels)
print('Test accuracy:', test_acc)
```

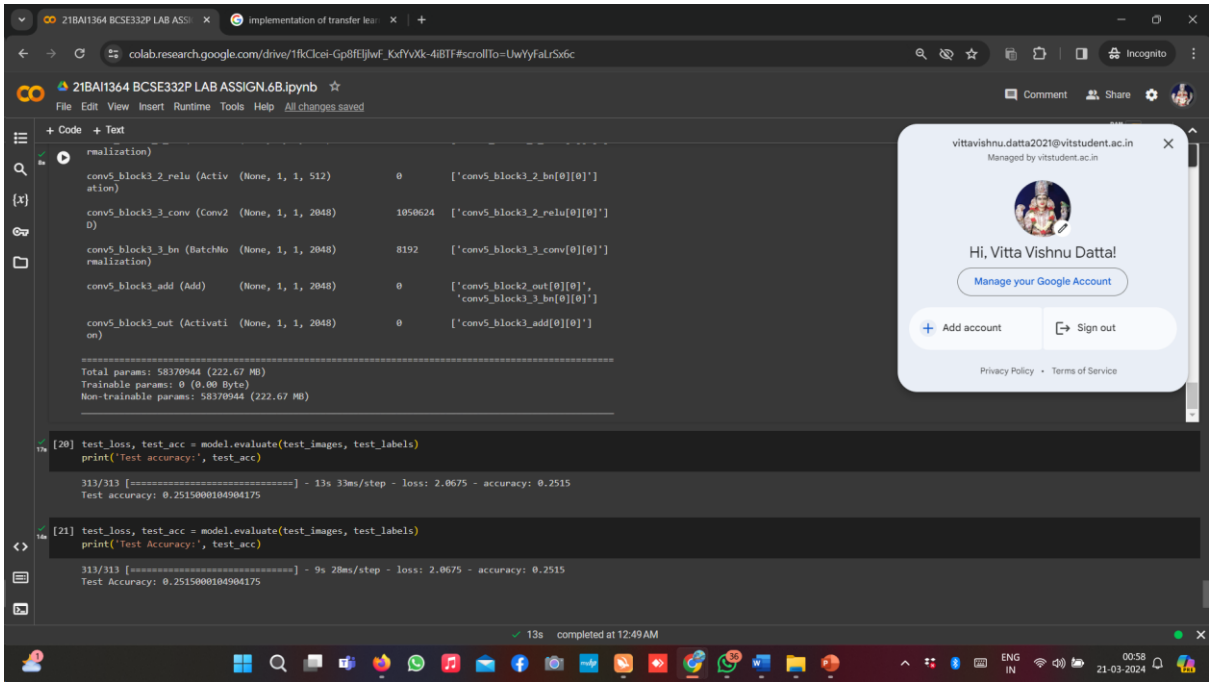

21BAI1364 BCSE332P LAB ASSIGN.6B.ipynb
File Edit View Insert Runtime Tools Help All changes saved
+ Code + Text
Implementing Transfer Learning with ResNet152
[18] from keras.applications import ResNet152
from keras.datasets import cifar10
from keras.utils import to_categorical
from keras.preprocessing.image import ImageDataGenerator
from keras import layers
from keras import models
from keras import optimizers
# Load the CIFAR-10 dataset
(train_images, train_labels), (test_images, test_labels) = cifar10.load_data()
# Preprocess the data
train_images = train_images.astype('float32') / 255
test_images = test_images.astype('float32') / 255
# Convert labels to categorical
train_labels = to_categorical(train_labels, 10)
test_labels = to_categorical(test_labels, 10)
# Load the ResNet152 network
base_model = ResNet152(weights='imagenet', include_top=False, input_shape=(32, 32, 3))
# Freeze the layers of the base model
for layer in base_model.layers:
layer.trainable = False
# Create the model
model = models.Sequential()
model.add(base_model)
model.add(layers.Flatten())
vittavishnu.datta2021@vitstudent.ac.in
Managed by vitstudent.ac.in
Hi, Vitta Vishnu Datta!
Manage your Google Account
Add account
Sign out
Privacy Policy • Terms of Service
13s completed at 12:49 AM


21BAI1364 BCSE332P LAB ASSIGN.6B.ipynb
File Edit View Insert Runtime Tools Help All changes saved
+ Code + Text
[18] for layer in base_model.layers:
layer.trainable = False
# Create the model
model = models.Sequential()
model.add(base_model)
model.add(layers.Flatten())
model.add(layers.Dense(256, activation='relu'))
model.add(layers.Dropout(0.5))
model.add(layers.Dense(10, activation='softmax')) # CIFAR-10 has 10 classes
Compile the model
model.compile(optimizer=optimizers.RMSprop(lr=2e-5),
loss='categorical_crossentropy',
metrics=['accuracy'])
# Train the model
history = model.fit(train_images, train_labels,
epochs=5,
batch_size=20,
validation_data=(test_images, test_labels))
Google Account
Vitta Vishnu Datta 21BAI1364
vittavishnu.datta2021@vitstudent.ac.in
Downloading data from https://storage.googleapis.com/tensorflow/keras-applications/resnet/resnet152_weights_tf_dim_ordering_tf_kernels_notop.h5
234698864/234698864 [==========] - 6s 0us/step
WARNING:absl:`lr` is deprecated in Keras optimizer, please use `learning_rate` or use the legacy optimizer, e.g.,tf.keras.optimizers.legacy.RMSprop.
Epoch 1/5
2500/2500 [==========] - 92s 32ms/step - loss: 2.2446 - accuracy: 0.1514 - val_loss: 2.1579 - val_accuracy: 0.2271
Epoch 2/5
2500/2500 [==========] - 85s 34ms/step - loss: 2.1761 - accuracy: 0.1896 - val_loss: 2.0999 - val_accuracy: 0.2272
Epoch 3/5
2500/2500 [==========] - 82s 33ms/step - loss: 2.1490 - accuracy: 0.2032 - val_loss: 2.0821 - val_accuracy: 0.2413
Epoch 4/5
2500/2500 [==========] - 72s 29ms/step - loss: 2.1356 - accuracy: 0.2096 - val_loss: 2.0682 - val_accuracy: 0.2316
Epoch 5/5
2500/2500 [==========] - 81s 33ms/step - loss: 2.1274 - accuracy: 0.2146 - val_loss: 2.0675 - val_accuracy: 0.2515
13s completed at 12:49 AM


21BAI1364 BCSE332P LAB ASSIGN.6B.ipynb
File Edit View Insert Runtime Tools Help All changes saved
+ Code + Text
[19] base_model.summary()
conv4_block36_3_conv (Conv2D) (None, 2, 2, 1024) 263168 ['conv4_block36_2_relu[0][0]']
conv4_block36_3_bn (BatchNormalization) (None, 2, 2, 1024) 4096 ['conv4_block36_3_conv[0][0]']
conv4_block36_add (Add) (None, 2, 2, 1024) 0 ['conv4_block35_out[0][0]', 'conv4_block36_3_bn[0][0]']
conv4_block36_out (Activation) (None, 2, 2, 1024) 0 ['conv4_block36_add[0][0]']
conv5_block1_1_conv (Conv2D) (None, 1, 1, 512) 524800 ['conv4_block36_out[0][0]']
conv5_block1_1_bn (BatchNormalization) (None, 1, 1, 512) 2048 ['conv5_block1_1_conv[0][0]']
conv5_block1_1_relu (Activation) (None, 1, 1, 512) 0 ['conv5_block1_1_bn[0][0]']
conv5_block1_2_conv (Conv2D) (None, 1, 1, 512) 2359808 ['conv5_block1_1_relu[0][0]']
conv5_block1_2_bn (BatchNormalization) (None, 1, 1, 512) 2048 ['conv5_block1_2_conv[0][0]']
conv5_block1_2_relu (Activation) (None, 1, 1, 512) 0 ['conv5_block1_2_bn[0][0]']
conv5_block1_0_conv (Conv2D) (None, 1, 1, 2048) 2099200 ['conv4_block36_out[0][0]']
13s completed at 12:49 AM
Comment Share
Incognito
Google Account
Vitta Vishnu Datta 21BAI1364
vittavishnu.datta2021@vitstudent.ac.in
ENG IN
00:51
21-03-2024


21BAI1364 BCSE332P LAB ASSIGN.6B.ipynb
File Edit View Insert Runtime Tools Help All changes saved
+ Code + Text
[19] base_model.summary()
conv5_block2_1_conv (Conv2D) (None, 1, 1, 512) 1049088 ['conv5_block1_out[0][0]']
conv5_block2_1_bn (BatchNormalization) (None, 1, 1, 512) 2048 ['conv5_block2_1_conv[0][0]']
conv5_block2_1_relu (Activation) (None, 1, 1, 512) 0 ['conv5_block2_1_bn[0][0]']
conv5_block2_2_conv (Conv2D) (None, 1, 1, 512) 2359808 ['conv5_block2_1_relu[0][0]']
conv5_block2_2_bn (BatchNormalization) (None, 1, 1, 512) 2048 ['conv5_block2_2_conv[0][0]']
conv5_block2_2_relu (Activation) (None, 1, 1, 512) 0 ['conv5_block2_2_bn[0][0]']
conv5_block2_3_conv (Conv2D) (None, 1, 1, 2048) 1050624 ['conv5_block2_2_relu[0][0]']
conv5_block2_3_bn (BatchNormalization) (None, 1, 1, 2048) 8192 ['conv5_block2_3_conv[0][0]']
conv5_block2_add (Add) (None, 1, 1, 2048) 0 ['conv5_block1_out[0][0]', 'conv5_block2_3_bn[0][0]']
conv5_block2_out (Activation) (None, 1, 1, 2048) 0 ['conv5_block2_add[0][0]']
conv5_block3_1_conv (Conv2D) (None, 1, 1, 512) 1049088 ['conv5_block2_out[0][0]']
conv5_block3_1_bn (BatchNormalization) (None, 1, 1, 512) 2048 ['conv5_block3_1_conv[0][0]']
13s completed at 12:49 AM
Comment Share
Incognito
Google Account
Vitta Vishnu Datta 21BAI1364
vittavishnu.datta2021@vitstudent.ac.in
ENG IN
00:51
21-03-2024


21BAI1364 BCSE332P LAB ASSIGN.6B.ipynb
File Edit View Insert Runtime Tools Help All changes saved
+ Code + Text
conv5_block2_1_bn (BatchNo rmalization) (None, 1, 1, 512) 2048 ['conv5_block2_1_conv[0][0]']
conv5_block2_1_relu (Activ ation) (None, 1, 1, 512) 0 ['conv5_block2_1_bn[0][0]']
conv5_block2_2_conv (Conv2 D) (None, 1, 1, 512) 2359808 ['conv5_block2_1_relu[0][0]']
conv5_block2_2_bn (BatchNo rmalization) (None, 1, 1, 512) 2048 ['conv5_block2_2_conv[0][0]']
conv5_block2_2_relu (Activ ation) (None, 1, 1, 512) 0 ['conv5_block2_2_bn[0][0]']
conv5_block2_3_conv (Conv2 D) (None, 1, 1, 2048) 1050624 ['conv5_block2_2_relu[0][0]']
conv5_block2_3_bn (BatchNo rmalization) (None, 1, 1, 2048) 8192 ['conv5_block2_3_conv[0][0]']
conv5_block2_add (Add) (None, 1, 1, 2048) 0 ['conv5_block1_out[0][0]', 'conv5_block2_3_bn[0][0]']
conv5_block2_out (Activati on) (None, 1, 1, 2048) 0 ['conv5_block2_add[0][0]']
conv5_block3_1_conv (Conv2 D) (None, 1, 1, 512) 1049088 ['conv5_block2_out[0][0]']
conv5_block3_1_bn (BatchNo rmalization) (None, 1, 1, 512) 2048 ['conv5_block3_1_conv[0][0]']
conv5_block3_1_relu (Activ ation) (None, 1, 1, 512) 0 ['conv5_block3_1_bn[0][0]']
conv5_block3_2_conv (Conv2 D) (None, 1, 1, 512) 2359808 ['conv5_block3_1_relu[0][0]']
vittavishnu.datta2021@vitstudent.ac.in
Managed by vitstudent.ac.in
Hi, Vitta Vishnu Datta!
Manage your Google Account
Add account
Sign out
Privacy Policy • Terms of Service
https://accounts.google.com/SignOutOptions?hl=en&continue=https://colab.research.google.com/drive/1fkClcei-Gp8fEljlwF_KxfYvXk-4iBTF&ec=GBRAqQM


21BAI1364 BCSE332P LAB ASSIGN.6B.ipynb
File Edit View Insert Runtime Tools Help All changes saved
+ Code + Text
conv5_block3_2_relu (Activ ation) (None, 1, 1, 512) 0 ['conv5_block3_2_bn[0][0]']
conv5_block3_3_conv (Conv2 D) (None, 1, 1, 2048) 1050624 ['conv5_block3_2_relu[0][0]']
conv5_block3_3_bn (BatchNo rmalization) (None, 1, 1, 2048) 8192 ['conv5_block3_3_conv[0][0]']
conv5_block3_add (Add) (None, 1, 1, 2048) 0 ['conv5_block2_out[0][0]', 'conv5_block3_3_bn[0][0]']
conv5_block3_out (Activati on) (None, 1, 1, 2048) 0 ['conv5_block3_add[0][0]']
Total params: 58370944 (222.67 MB)
Trainable params: 0 (0.00 Byte)
Non-trainable params: 58370944 (222.67 MB)
[20] test_loss, test_acc = model.evaluate(test_images, test_labels)
print('Test accuracy:', test_acc)
313/313 [==========] - 13s 33ms/step - loss: 2.0675 - accuracy: 0.2515
Test accuracy: 0.2515000104904175
[21] test_loss, test_acc = model.evaluate(test_images, test_labels)
print('Test Accuracy:', test_acc)
313/313 [==========] - 9s 28ms/step - loss: 2.0675 - accuracy: 0.2515
Test Accuracy: 0.2515000104904175
13s completed at 12:49 AM
vittavishnu.datta2021@vitstudent.ac.in
Managed by vitstudent.ac.in
Hi, Vitta Vishnu Datta!
Manage your Google Account
Add account
Sign out
Privacy Policy • Terms of Service

****************THANK YOU*****************