Giovanni Rodríguez Gutiérrez
S0556233. 9. July 2022. Neural Nets.

1. Color Spaces.

The chosen color space can indeed affect the performance of the model. RGB is the most common color space, but in many situations the model performance increases if a color space (such as YUV, YCbCr, CIE XYZ, among many others), it depends on the application and kind of the images (source of the images or camera used to take them) used as training dataset.

The varying imaging conditions are to be taken into account when choosing a color space, for example, varying lighting conditions can affect the performance greatly, that is why choosing a invariant color space like the normalized rgb space deals with the problem for the variations coming from the light sources in the images.

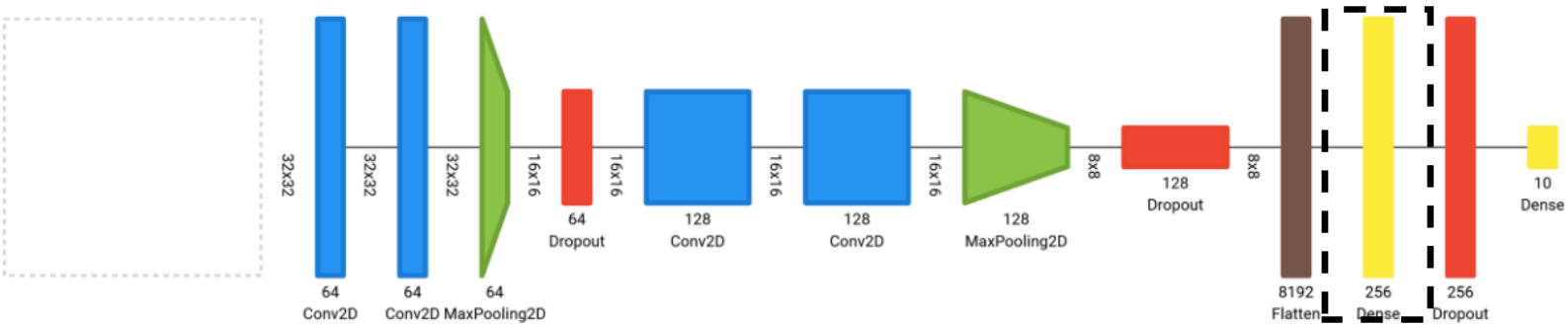
As well how the color space can be decomposed into its base components (i.e RGB: Red, Green, Blue) with each component containing meaningful information for the training model.

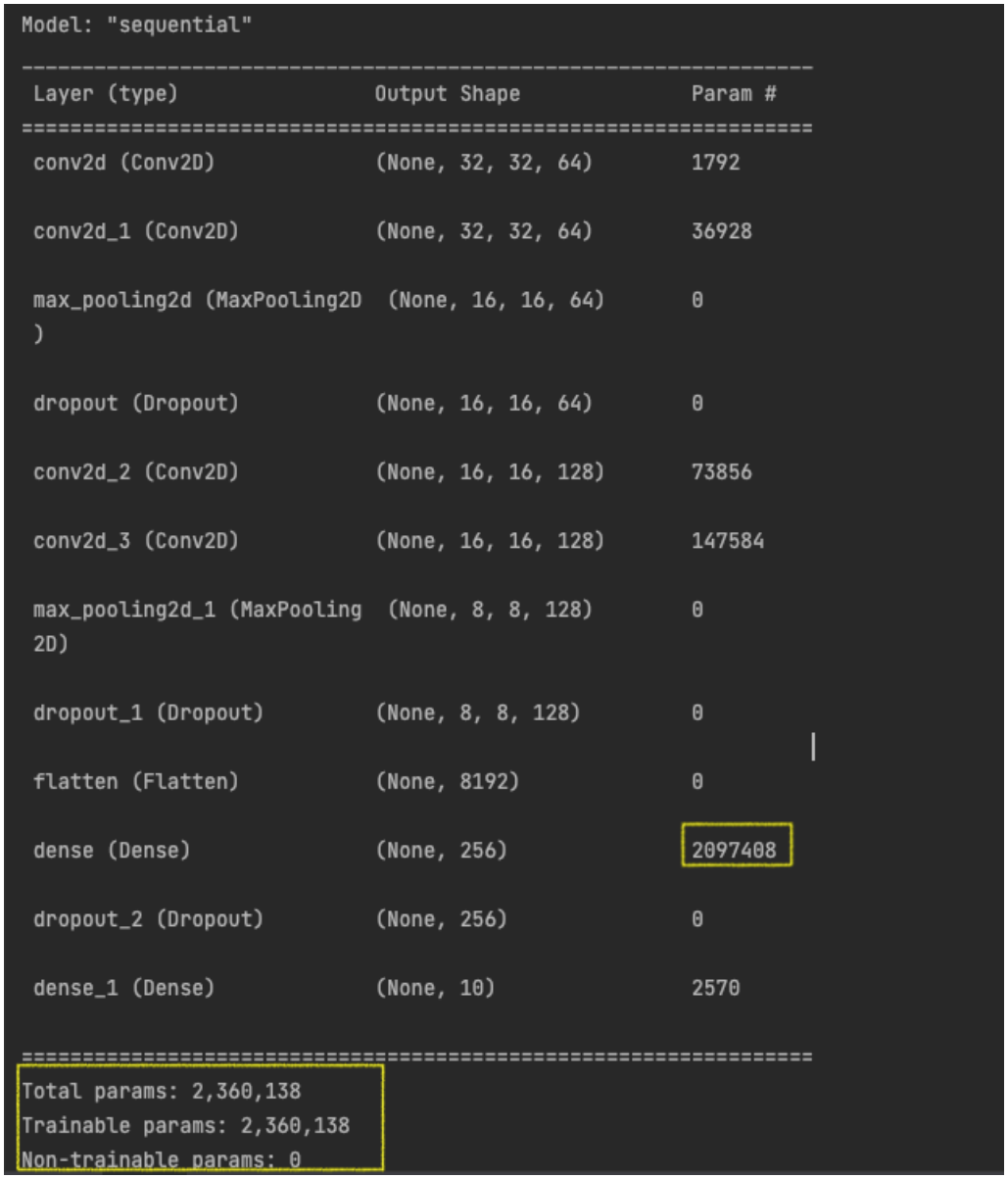
2. Number of Training Params:

2.360.138

3. Layer which contributes the most trainable parameters to the network:

The first dense Layer with 2.097.408 Parameters.



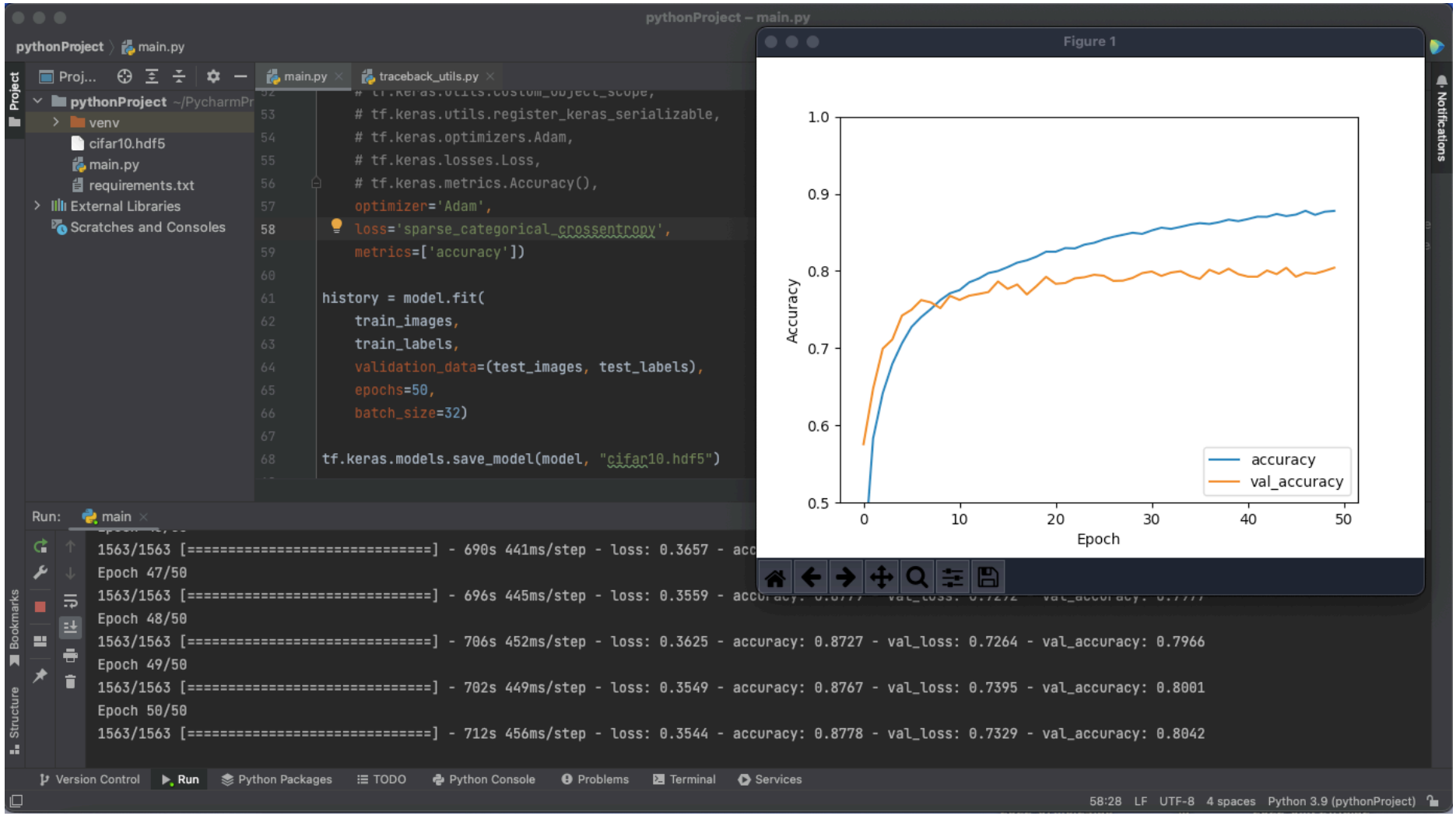


```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d (Conv2D)             (None, 32, 32, 64)        1792

 conv2d_1 (Conv2D)           (None, 32, 32, 64)        36928

 max_pooling2d (MaxPooling2D  (None, 16, 16, 64)       0
 )

 dropout (Dropout)           (None, 16, 16, 64)        0

 conv2d_2 (Conv2D)           (None, 16, 16, 128)       73856

 conv2d_3 (Conv2D)           (None, 16, 16, 128)       147584

 max_pooling2d_1 (MaxPooling  (None, 8, 8, 128)        0
 2D)

 dropout_1 (Dropout)         (None, 8, 8, 128)         0

 flatten (Flatten)           (None, 8192)              0

 dense (Dense)               (None, 256)               2097408

 dropout_2 (Dropout)         (None, 256)               0

 dense_1 (Dense)             (None, 10)                2570

=================================================================
Total params: 2,360,138
Trainable params: 2,360,138
Non-trainable params: 0
```

Model summary.

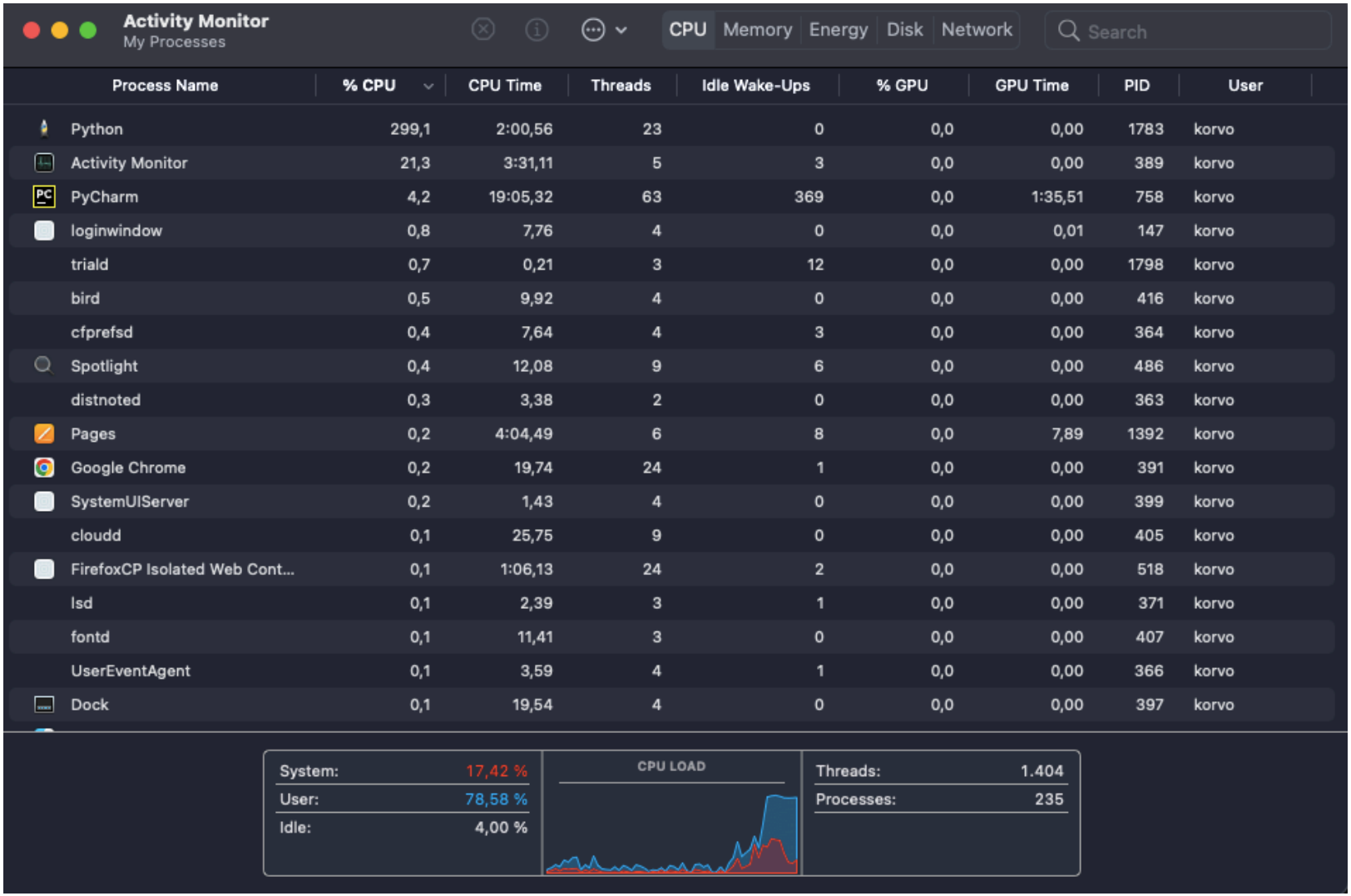
# Model Training:

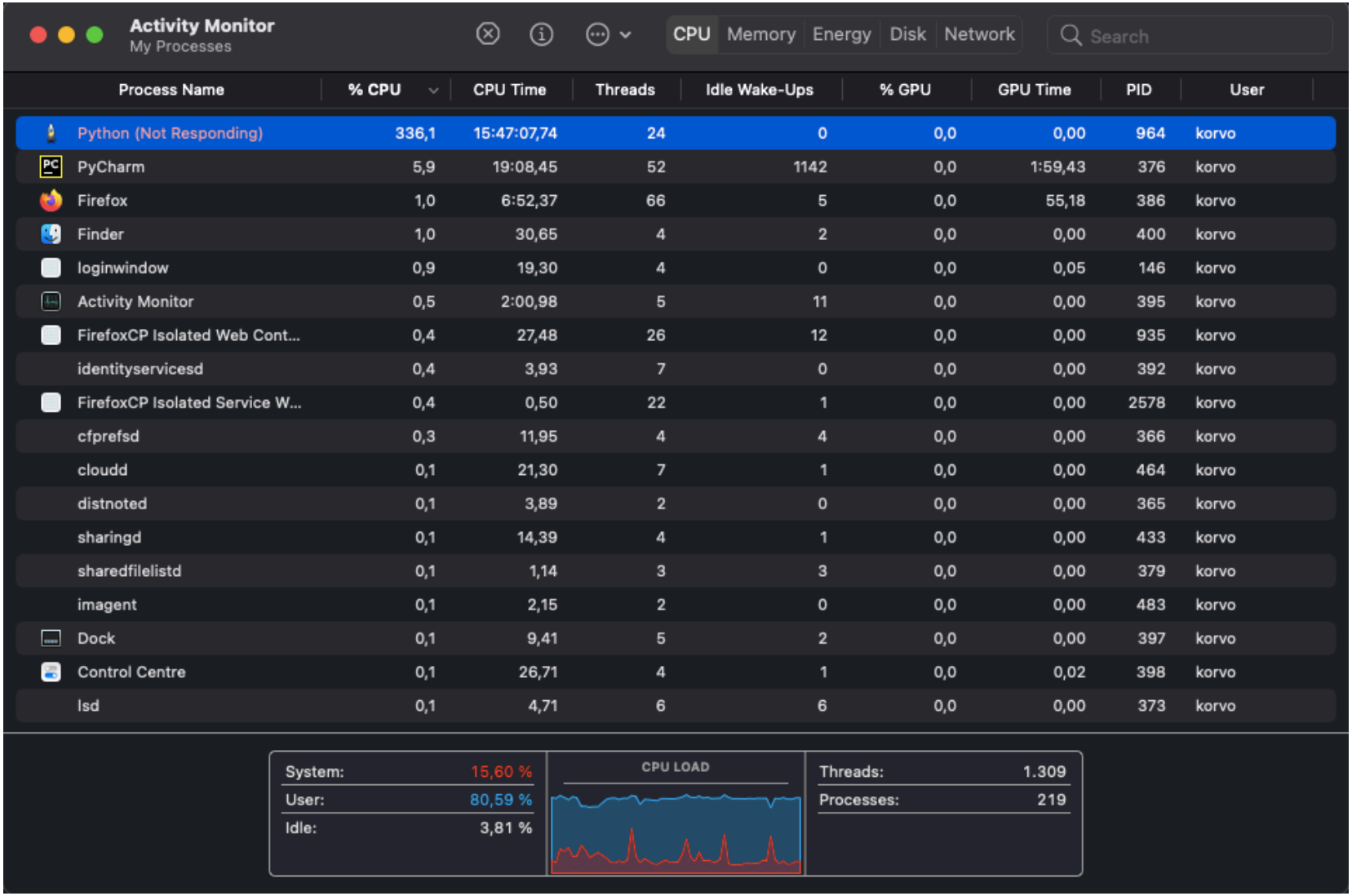
The accuracy obtained was nearly 80% as expected



# CPU Load while training:

The model was trained only on CPU, which took several hours, it is too observe how the CPU Load increases when the training begins and remains high during the whole training, as seen in the screenshots below.

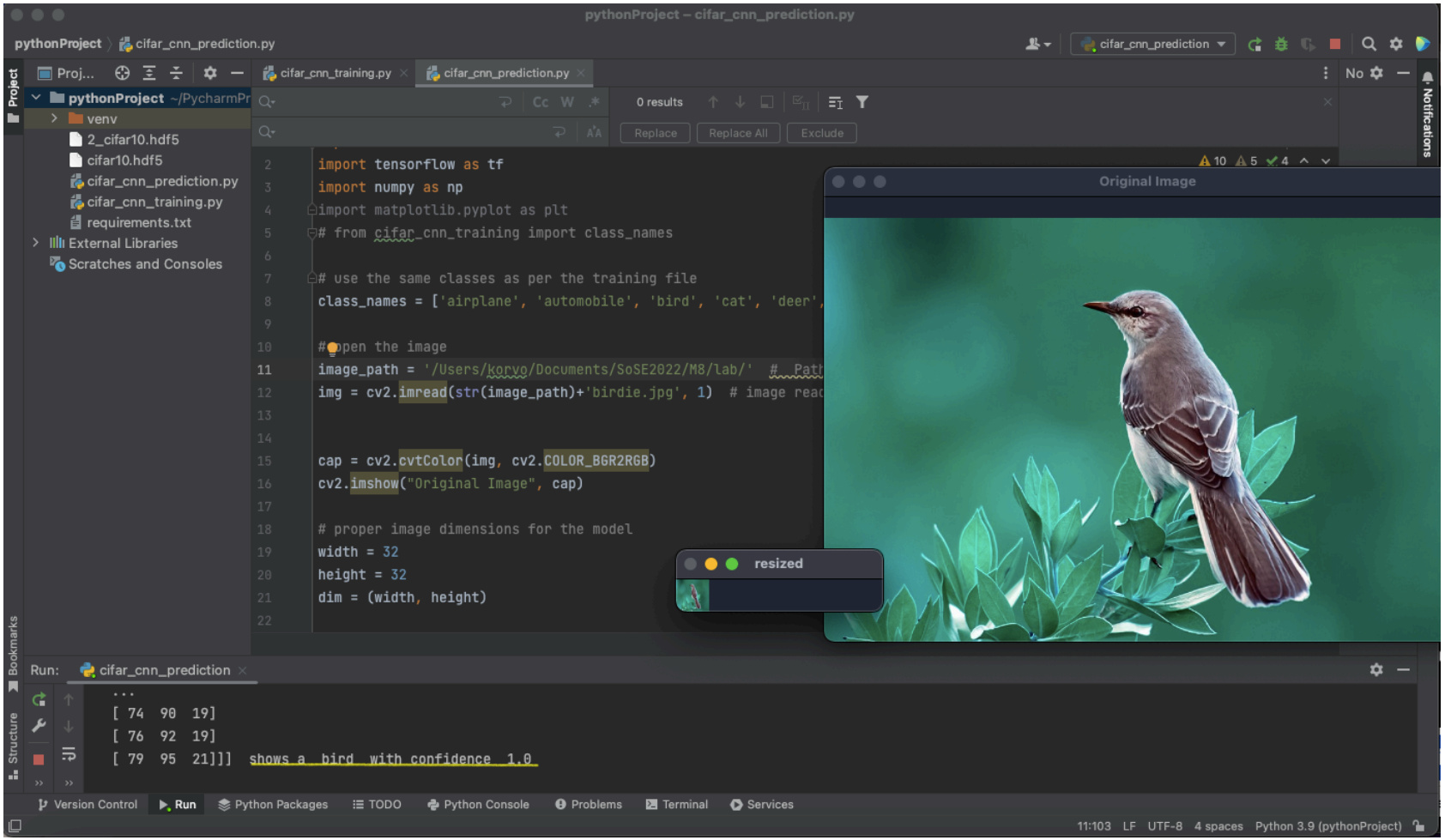


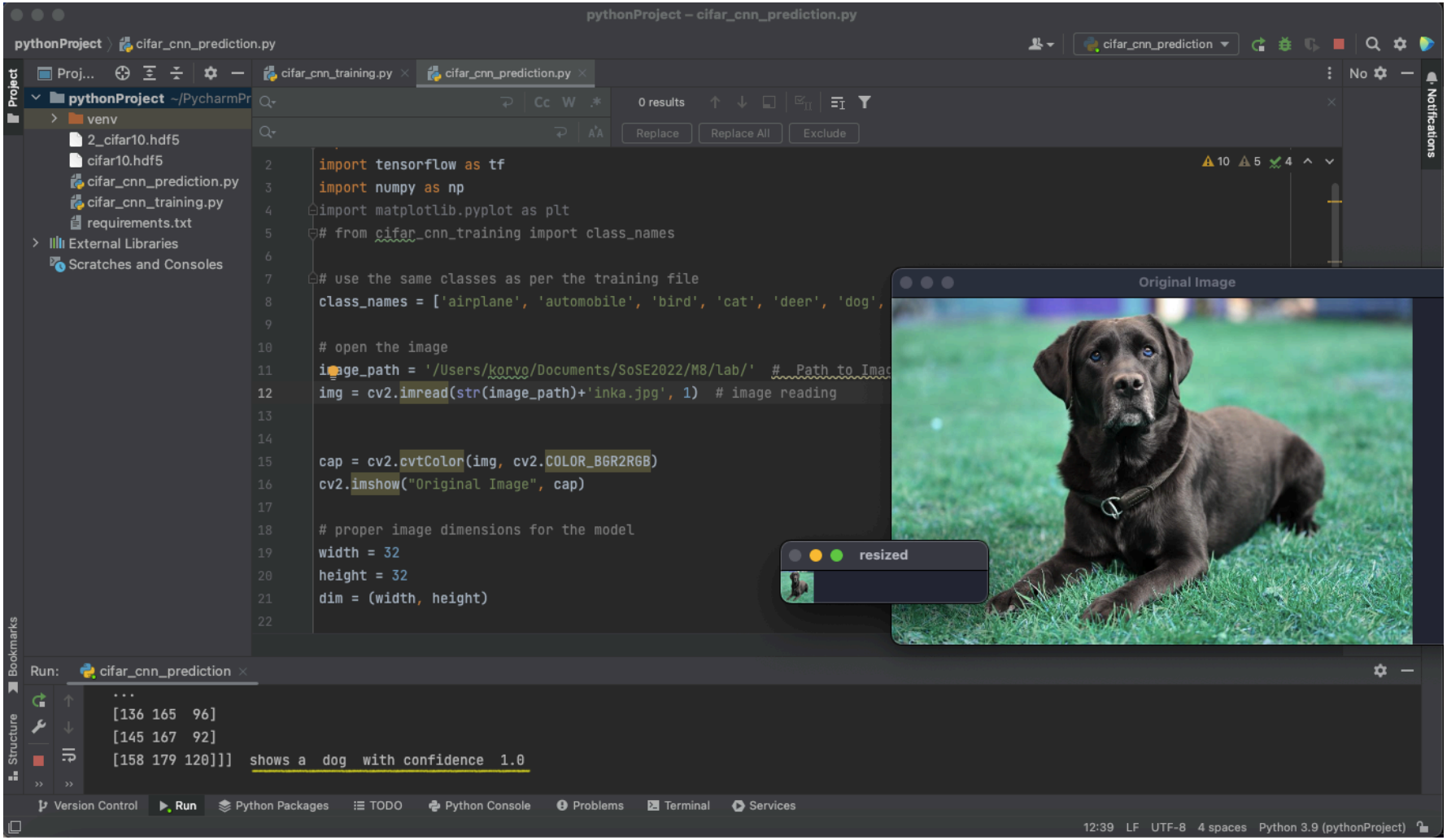


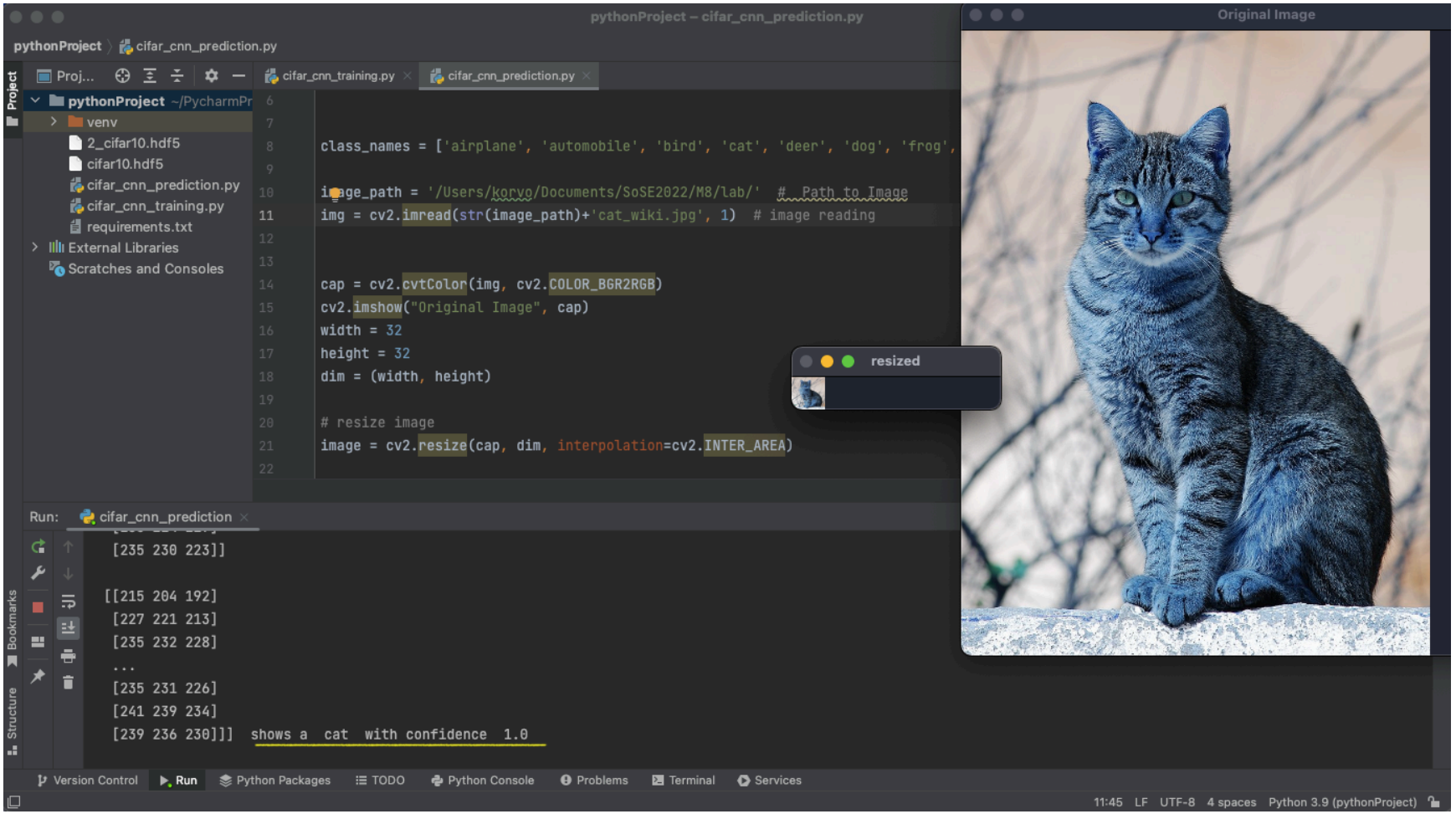

# Predicting images:

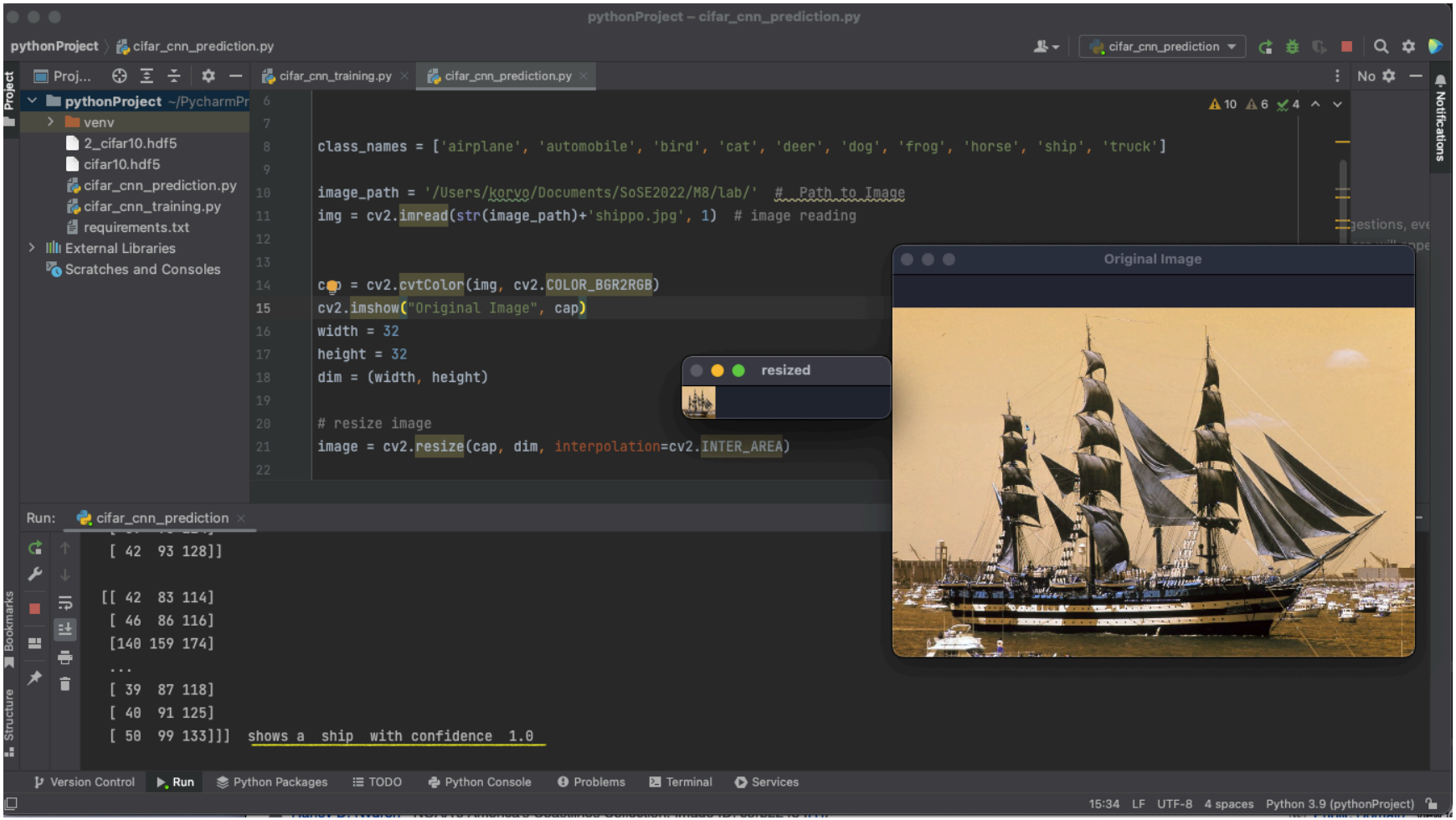
These are the properly predicted images using the trained model:

Class number 3, bird:



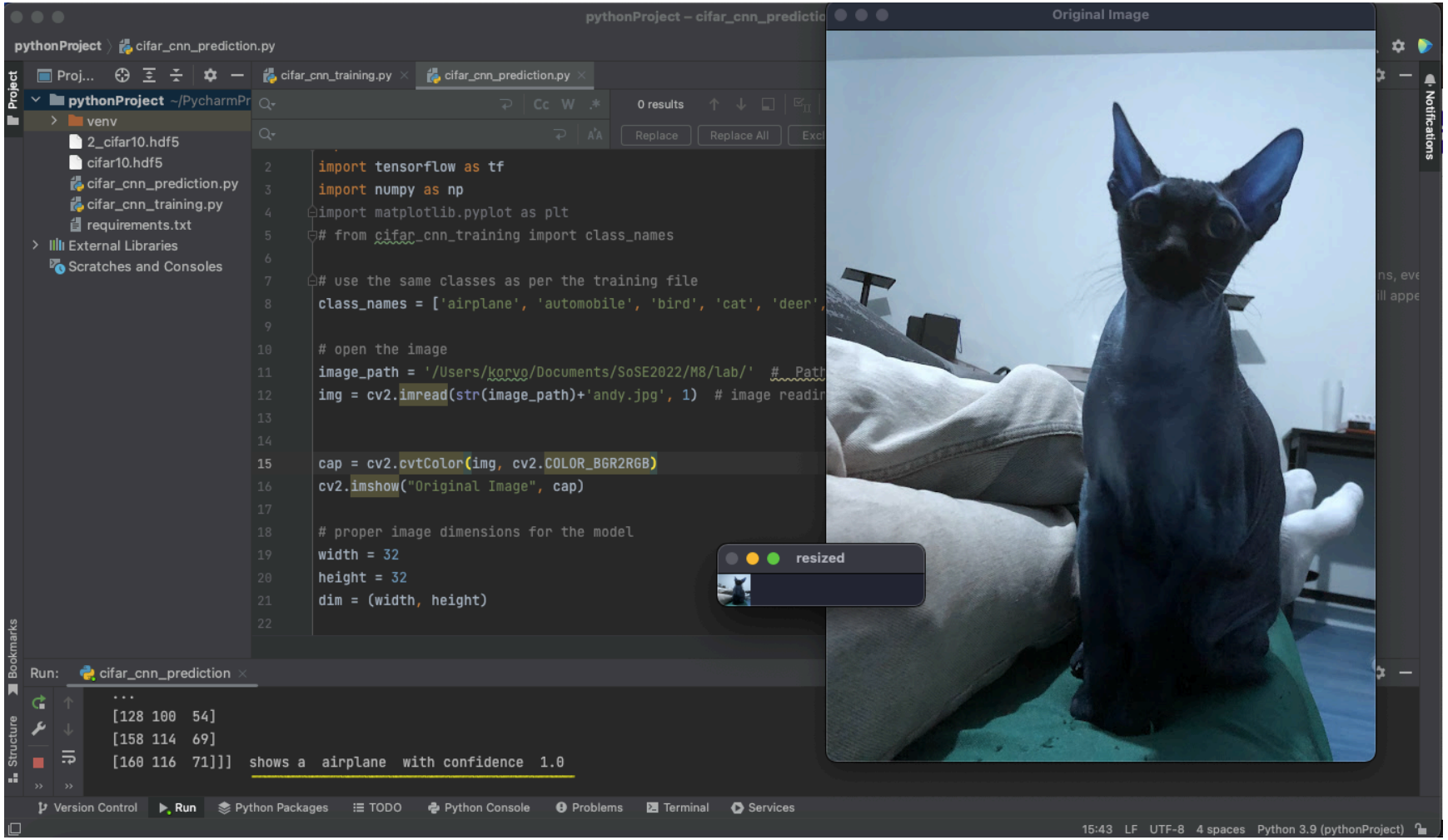

pythonProject – cifar_cnn_prediction.py
pythonProject
cifar_cnn_prediction.py
cifar_cnn_training.py
cifar_cnn_prediction.py
pythonProject ~/PycharmPr
venv
2_cifar10.hdf5
cifar10.hdf5
cifar_cnn_prediction.py
cifar_cnn_training.py
requirements.txt
External Libraries
Scratches and Consoles
0 results
Replace
Replace All
Exclude
import tensorflow as tf
import numpy as np
import matplotlib.pyplot as plt
# from cifar_cnn_training import class_names
# use the same classes as per the training file
class_names = ['airplane', 'automobile', 'bird', 'cat', 'deer', 'dog',
# open the image
image_path = '/Users/korvo/Documents/SoSE2022/M8/lab/'
img = cv2.imread(str(image_path)+'inka.jpg', 1) # image reading
cap = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
cv2.imshow("Original Image", cap)
# proper image dimensions for the model
width = 32
height = 32
dim = (width, height)
Original Image
resized
Run:
cifar_cnn_prediction
...
[136 165 96]
[145 167 92]
[158 179 120]]] shows a dog with confidence 1.0
Version Control
Run
Python Packages
TODO
Python Console
Problems
Terminal
Services
12:39 LF UTF-8 4 spaces Python 3.9 (pythonProject)

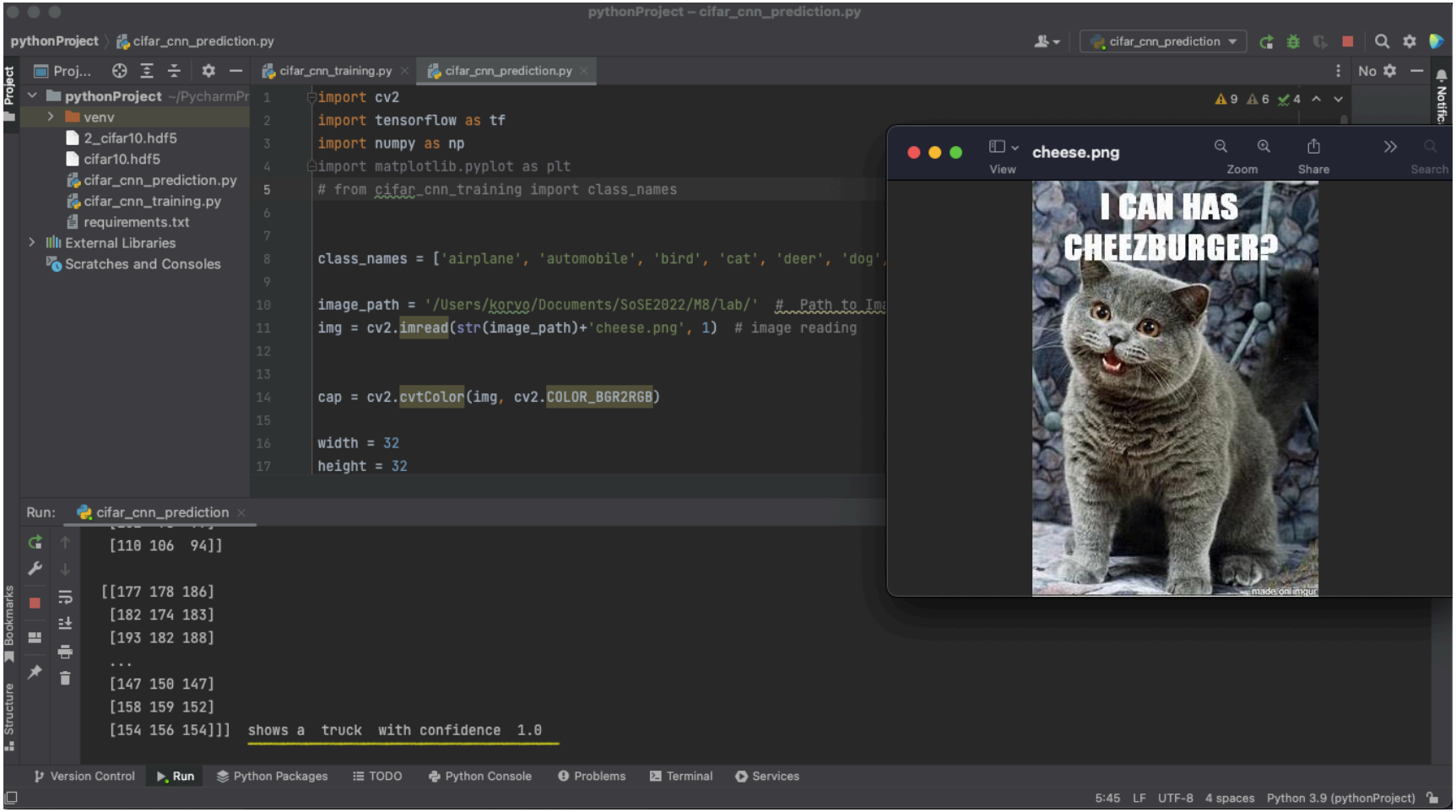

pythonProject – cifar_cnn_prediction.py
pythonProject
cifar_cnn_prediction.py
Proj...
pythonProject ~/PycharmPr
venv
2_cifar10.hdf5
cifar10.hdf5
cifar_cnn_prediction.py
cifar_cnn_training.py
requirements.txt
External Libraries
Scratches and Consoles
cifar_cnn_training.py
cifar_cnn_prediction.py
class_names = ['airplane', 'automobile', 'bird', 'cat', 'deer', 'dog', 'frog',
image_path = '/Users/korvo/Documents/SoSE2022/M8/lab/' # Path to Image
img = cv2.imread(str(image_path)+'cat_wiki.jpg', 1) # image reading
cap = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
cv2.imshow("Original Image", cap)
width = 32
height = 32
dim = (width, height)
# resize image
image = cv2.resize(cap, dim, interpolation=cv2.INTER_AREA)
resized
Original Image
Run:
cifar_cnn_prediction
[235 230 223]]
[[215 204 192]
[227 221 213]
[235 232 228]
...
[235 231 226]
[241 239 234]
[239 236 230]]] shows a cat with confidence 1.0
Version Control
Run
Python Packages
TODO
Python Console
Problems
Terminal
Services
11:45 LF UTF-8 4 spaces Python 3.9 (pythonProject)




But Some others are not being correctly predicted:

Perhaps the scenarios are too complex for this model


pythonProject – cifar_cnn_predictic
pythonProject 〉 cifar_cnn_prediction.py
Original Image
cifar_cnn_training.py
cifar_cnn_prediction.py
pythonProject ~/PycharmPr
venv
2_cifar10.hdf5
cifar10.hdf5
cifar_cnn_prediction.py
cifar_cnn_training.py
requirements.txt
External Libraries
Scratches and Consoles
0 results
Replace
Replace All
import tensorflow as tf
import numpy as np
import matplotlib.pyplot as plt
# from cifar_cnn_training import class_names
# use the same classes as per the training file
class_names = ['airplane', 'automobile', 'bird', 'cat', 'deer',
# open the image
image_path = '/Users/korvo/Documents/SoSE2022/M8/lab/'
img = cv2.imread(str(image_path)+'andy.jpg', 1)
cap = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
cv2.imshow("Original Image", cap)
# proper image dimensions for the model
width = 32
height = 32
dim = (width, height)
resized
Run: cifar_cnn_prediction
...
[128 100 54]
[158 114 69]
[160 116 71]]] shows a airplane with confidence 1.0
Version Control
Run
Python Packages
TODO
Python Console
Problems
Terminal
Services
15:43 LF UTF-8 4 spaces Python 3.9 (pythonProject)


pythonProject – cifar_cnn_prediction.py
pythonProject 〉 cifar_cnn_prediction.py
cifar_cnn_prediction
Proj...
pythonProject ~/PycharmPr
venv
2_cifar10.hdf5
cifar10.hdf5
cifar_cnn_prediction.py
cifar_cnn_training.py
requirements.txt
External Libraries
Scratches and Consoles
cifar_cnn_training.py
cifar_cnn_prediction.py
1 import cv2
2 import tensorflow as tf
3 import numpy as np
4 import matplotlib.pyplot as plt
5 # from cifar_cnn_training import class_names
6
7
8 class_names = ['airplane', 'automobile', 'bird', 'cat', 'deer', 'dog',
9
10 image_path = '/Users/korvo/Documents/SoSE2022/M8/lab/' # Path to Ima
11 img = cv2.imread(str(image_path)+'cheese.png', 1) # image reading
12
13
14 cap = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
15
16 width = 32
17 height = 32
cheese.png
View
Zoom
Share
Search
I CAN HAS CHEEZBURGER?
made on imgur
Run: cifar_cnn_prediction
[110 106 94]]
[[177 178 186]
[182 174 183]
[193 182 188]
...
[147 150 147]
[158 159 152]
[154 156 154]]] shows a truck with confidence 1.0
Version Control
Run
Python Packages
TODO
Python Console
Problems
Terminal
Services
5:45 LF UTF-8 4 spaces Python 3.9 (pythonProject)