**Faculty of Computers and Artificial Intelligence**

**Cairo University**

**Supervised Learning**

**AI322**

**Assignment 3**

**(CNN Report)**

|  |  |
| --- | --- |
| **Name** | **ID** |
| Loai Gamal Mohamed | 20180206 |
| Ahmed Kadry Abd El-Shafy | 20180018 |

**Prof. Khaled El-Sayed**

# 0-The code:First implementation of the code.

# Simple CNN for the MNIST Dataset

import keras.optimizers

from keras.datasets import mnist

from keras.models import Sequential  # Allows to build the architecture for neural network

from keras.layers import Dense

from keras.layers import Dropout

from keras.layers import Flatten

from keras.layers.convolutional import Conv2D

from keras.layers.convolutional import MaxPooling2D

from keras.utils import np\_utils

# load data

(X\_train, y\_train), (X\_test, y\_test) = mnist.load\_data()

# reshape to be [samples][width][height][channels]

X\_train = X\_train.reshape(60000, 28, 28, 1)  # it makes images in grayscale

X\_test = X\_test.reshape(10000, 28, 28, 1)

# one hot encode outputs

y\_train = np\_utils.to\_categorical(y\_train)  # it makes label 5 = [ 0, 0, 0, 0, 0, 1, 0, 0, 0, 0]

y\_test = np\_utils.to\_categorical(y\_test)

num\_classes = y\_test.shape[1]

# define a simple CNN model

def baseline\_model():

    # create model

    model = Sequential()

    model.add(Conv2D(32, (2, 2), input\_shape=(28, 28, 1), activation='relu'))   # convolution layer to extract features from the input image

    model.add(MaxPooling2D(pool\_size=(2, 2), strides=(2, 2)))

    model.add(Flatten())        # take the images and flatten them (turn images into a one dimensional array)

    model.add(Dense(128, activation='relu'))

    model.add(Dense(num\_classes, activation='softmax'))

    # Compile model

    optimizer = keras.optimizers.SGD(lr=0.05)

    model.compile(loss='categorical\_crossentropy', optimizer=optimizer, metrics=['accuracy'])

    return model

# build the model

model = baseline\_model()

# Fit the model

model.fit(X\_train, y\_train, validation\_data=(X\_test, y\_test), epochs=10, batch\_size=32, shuffle=True)        # epochs : number of iterations when an entire data set is passed forward and backward through the neural network

model.summary()

model.count\_params()

# Final evaluation of the model

scores = model.evaluate(X\_test, y\_test, verbose=0)

print("CNN Error: %.2f%%" % (100 - scores[1] \* 100))

# A picture containing text, screenshot, black, computer Description automatically generated

# 1-Final accuracy of the model and the accuracy in the first 5 epoch:

-Accuracy of the model: 100-88.65=11.35%

|  |  |
| --- | --- |
| 1 Epoch 1 | 0.1107 |
| 2 Epoch 2 | 0.1130 |
| 3 Epoch 3 | 0.1151 |
| 4 Epoch 4 | 0.1109 |
| 5 Epoch 5 | 0.1099 |

# 2-The number of parameters in the model:

Total params: 693,802

Trainable params: 693,802

Non-trainable params: 0

# 3&4- The average time to train in each epoch &The average test time in each epoch:

|  |
| --- |
| Epoch 1/10  1875/1875 [==============================] - 45s 14ms/step  Epoch 2/10  1875/1875 [==============================] - 26s 14ms/step  Epoch 3/10  1875/1875 [==============================] - 27s 14ms/step  Epoch 4/10  1875/1875 [==============================] - 25s 13ms/step  Epoch 5/10  1875/1875 [==============================] - 24s 13ms/step  Epoch 6/10  1875/1875 [==============================] - 24s 13ms/step  Epoch 7/10  1875/1875 [==============================] - 23s 12ms/step  Epoch 8/10  1875/1875 [==============================] - 23s 12ms/step  Epoch 9/10  1875/1875 [==============================] - 23s 12ms/step  Epoch 10/10  1875/1875 [==============================] - 23s 12ms/step |

# 5-The layers of each model (including activations):

* Conv2D
* MaxPooling2D
* Flatten
* Dense
* Rule” activation “
* Softmax “activation”

# 6-The learning rate used and configuration of the optimizers:

* Optimizer :SGD
* Configuration: learing rate
* Lr:0.05

# 7-You are not required to test learning decays, if you wish to include it, then include it:

* Null

# 8-The optimizer used with its configuration:

* Optimizer :SGD
* Configuration: learing rate

# 9-If you used a dropout layer (you will use it in #7), write where you put it, what dropout rate used and why (your perspective on the location you’ve put it):

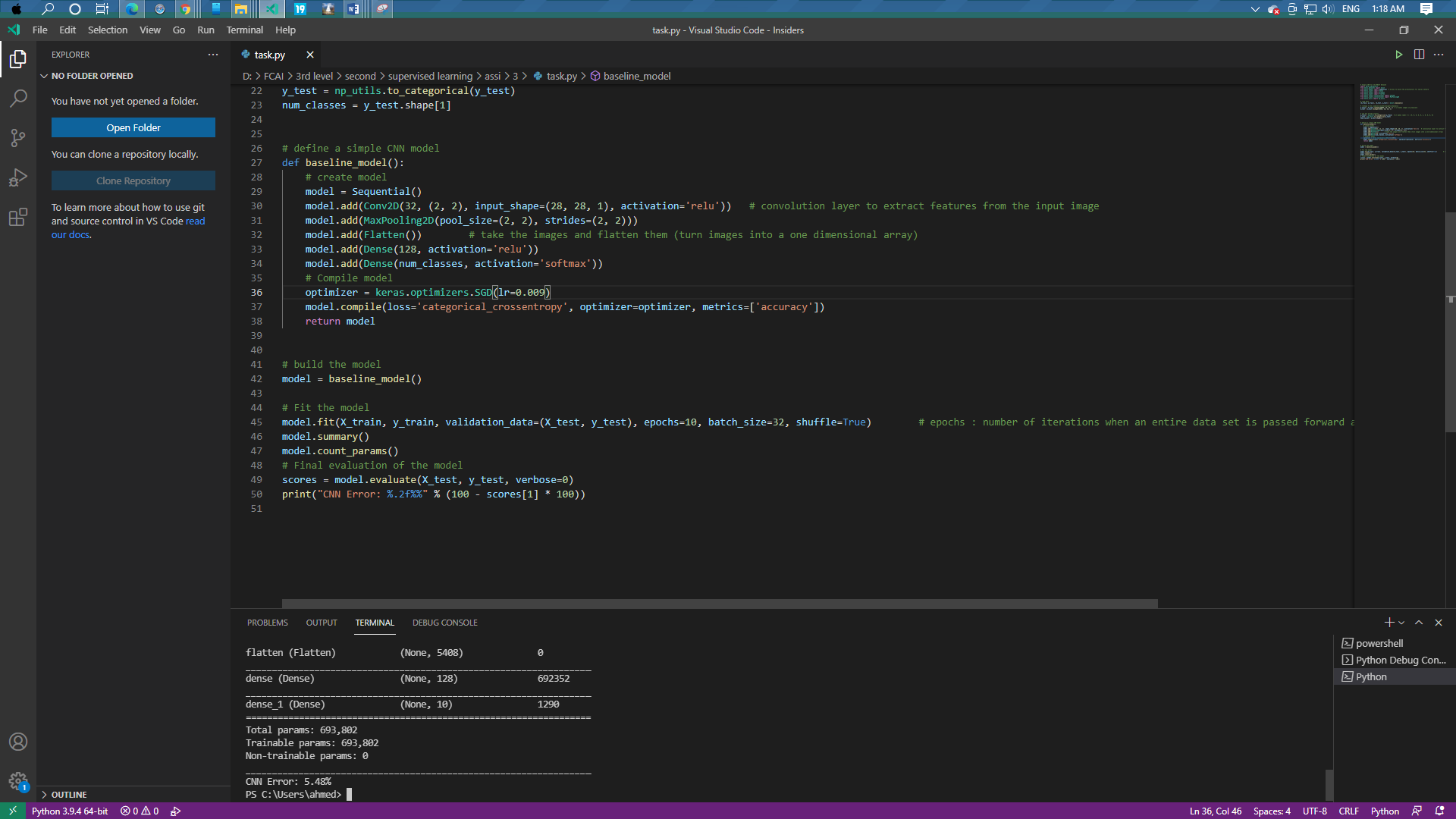
* Null

# 10-Write what you observed has changed due to the parameter you changed (according to the above part):

* Null

# 0-The code: we will change the learning rate from 0.05 into 0.009:

    optimizer = keras.optimizers.SGD(lr=0.009)



# 1-Final accuracy of the model and the accuracy in the first 5 epoch:

* Accuracy:100-5.48=94.52%

|  |  |
| --- | --- |
| 1 | 0.2268 |
| 2 | 0.8661 |
| 3 | 0.8894 |
| 4 | 0.9017 |
| 5 | 0.9152 |

# 2-The number of parameters in the model:

* Total params: 693,802
* Trainable params: 693,802
* Non-trainable params: 0

# 3&4- The average time to train in each epoch &The average test time in each epoch:

|  |
| --- |
| Epoch 1/10  1875/1875 [==============================] - 36s 13ms/step  Epoch 2/10  1875/1875 [==============================] - 24s 13ms/step  Epoch 3/10  1875/1875 [==============================] - 23s 13ms/step  Epoch 4/10  1875/1875 [==============================] - 23s 12ms/step  Epoch 5/10  1875/1875 [==============================] - 23s 12ms/step  Epoch 6/10  1875/1875 [==============================] - 23s 12ms/step  Epoch 7/10  1875/1875 [==============================] - 24s 13ms/step  Epoch 8/10  1875/1875 [==============================] - 23s 12ms/step  Epoch 9/10  1875/1875 [==============================] - 24s 13ms/step  Epoch 10/10  1875/1875 [==============================] - 23s 12ms/step |

# 5-The layers of each model (including activations):

* Conv2D
* MaxPooling2D
* Flatten
* Dense
* Rule” activation “
* Softmax “activation”

# 6-The learning rate used and configuration of the optimizers:

* Optimizer :SGD
* Configuration: learing rate
* Lr:0.009

# 7-You are not required to test learning decays, if you wish to include it, then include it:

* Null

# 8-The optimizer used with its configuration:

* Optimizer :SGD
* Configuration: learing rate

# 9-If you used a dropout layer (you will use it in #7), write where you put it, what dropout rate used and why (your perspective on the location you’ve put it):

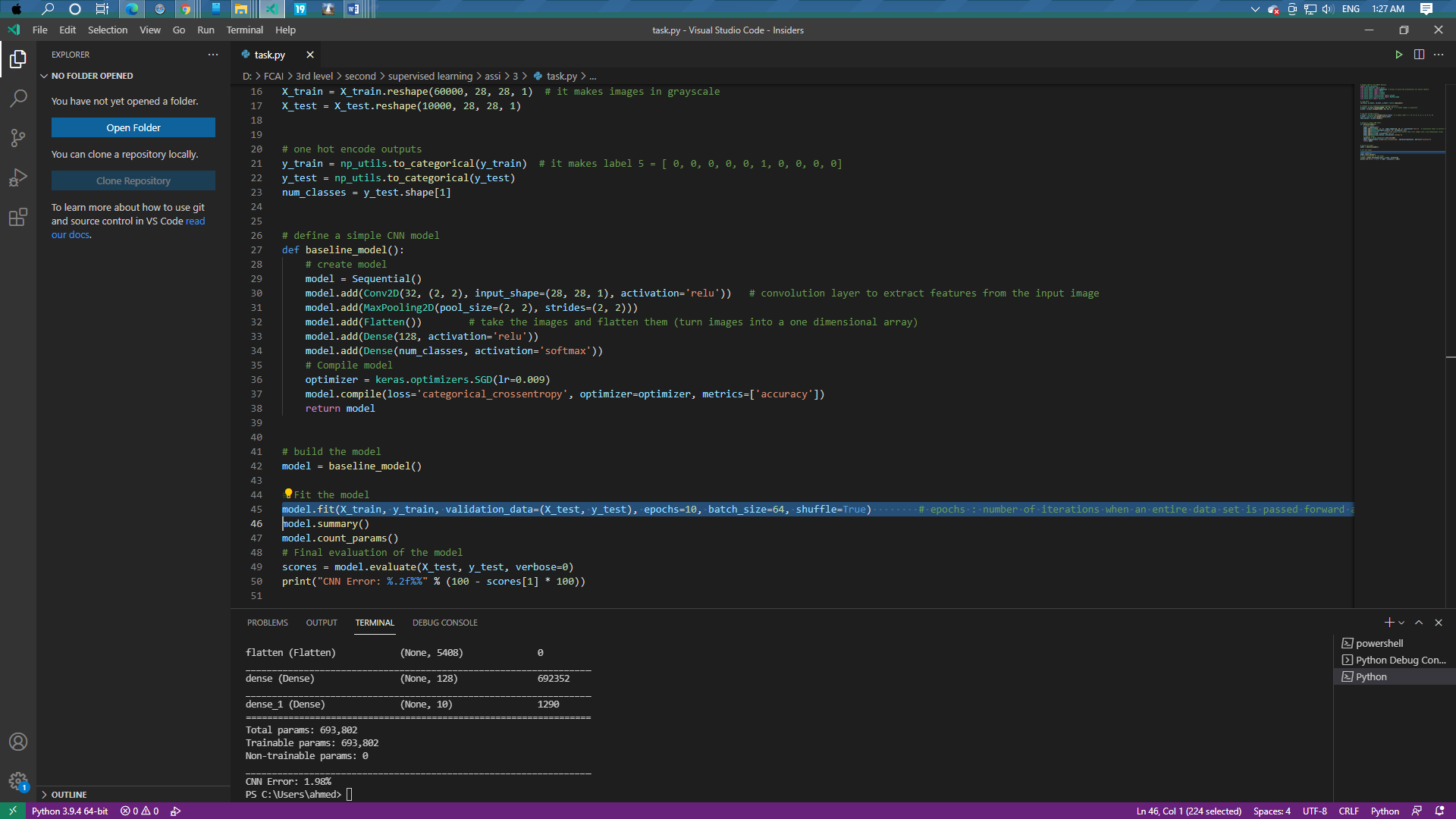
* Null

# 10-Write what you observed has changed due to the parameter you changed (according to the above part):

* The CNN Error is decreased from **88.65%** to **5.48%**. It means that when you decrease the learning rate that accuracy will increase, and the error will decrease.

# 0-The code: we will change the batch size from 32 into 64 (32 \* 2) with the same learning rate of the previous one (0.009).

model.fit(X\_train, y\_train, validation\_data=(X\_test, y\_test), epochs=10, batch\_size=64, shuffle=True)        # epochs : number of iterations when an entire data set is passed forward and backward through the neural network



# 1-Final accuracy of the model and the accuracy in the first 5 epoch:

* Accuracy:100-1.98=98.02%

|  |  |
| --- | --- |
| 1 | 0.8267 |
| 2 | 0.9737 |
| 3 | 0.9848 |
| 4 | 0.9908 |
| 5 | 0.9943 |

# 2-The number of parameters in the model:

* Total params: 693,802
* Trainable params: 693,802
* Non-trainable params: 0

# 3&4- The average time to train in each epoch &The average test time in each epoch:

|  |
| --- |
| Epoch 1/10  938/938 [==============================] - 41s 22ms/step  Epoch 2/10  938/938 [==============================] - 20s 21ms/step  Epoch 3/10  938/938 [==============================] - 21s 22ms/step  Epoch 4/10  938/938 [==============================] - 19s 21ms/step  Epoch 5/10  938/938 [==============================] - 19s 21ms/step  Epoch 6/10  938/938 [==============================] - 19s 21ms/step  Epoch 7/10  938/938 [==============================] - 20s 21ms/step  Epoch 8/10  938/938 [==============================] - 19s 21ms/step  Epoch 9/10  938/938 [==============================] - 19s 21ms/step  Epoch 10/10  938/938 [==============================] - 19s 21ms/step |

# 5-The layers of each model (including activations):

* Conv2D
* MaxPooling2D
* Flatten
* Dense
* Rule” activation “
* Softmax “activation”

# 6-The learning rate used and configuration of the optimizers:

* Optimizer :SGD
* Configuration: learing rate
* Lr:0.009

# 7-You are not required to test learning decays, if you wish to include it, then include it:

* Null

# 8-The optimizer used with its configuration:

* Optimizer :SGD
* Configuration: learing rate

# 9-If you used a dropout layer (you will use it in #7), write where you put it, what dropout rate used and why (your perspective on the location you’ve put it):

* Null

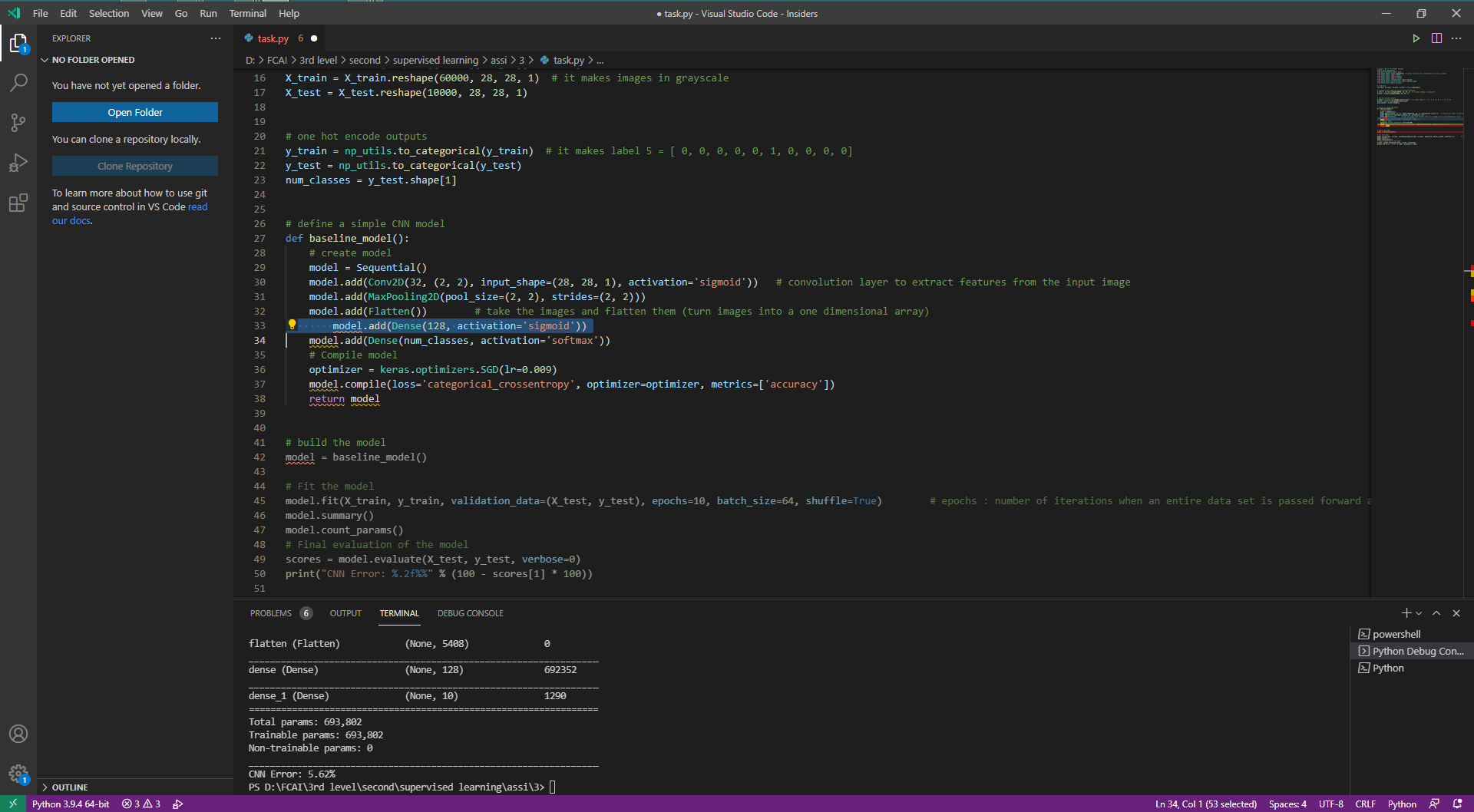
# 10-Write what you observed has changed due to the parameter you changed (according to the above part):

* As you see that the CNN Error decreased then when we increase the batch size from 32 to 64 with the same Learning rate the accuracy will increase, and the error will decrease.

# 0-The code: we will change all activations from relu to sigmoid with the same previous parameters (Learning rate = 0.009, Epochs = 10, Batch size = 64)

    model.add(Conv2D(32, (2, 2), input\_shape=(28, 28, 1), activation='sigmoid'))   # convolution layer to extract features from the input image

    model.add(Dense(128, activation='sigmoid'))



# 1-Final accuracy of the model and the accuracy in the first 5 epoch:

* Accuracy:100-5.62=94.38%

|  |  |
| --- | --- |
| 1 | 0.5635 |
| 2 | 0.8558 |
| 3 | 0.8876 |
| 4 | 0.9046 |
| 5 | 0.9154 |

# 2-The number of parameters in the model:

* Total params: 693,802
* Trainable params: 693,802
* Non-trainable params: 0

# 3&4- The average time to train in each epoch &The average test time in each epoch:

|  |
| --- |
| Epoch 1/10  938/938 [==============================] - 89s 45ms/step  Epoch 2/10  938/938 [==============================] - 40s 42ms/step  Epoch 3/10  938/938 [==============================] - 31s 33ms/step  Epoch 4/10  938/938 [==============================] - 23s 25ms/step  Epoch 5/10  938/938 [==============================] - 24s 26ms/step  Epoch 6/10  938/938 [==============================] - 22s 23ms/step  Epoch 7/10  938/938 [==============================] - 21s 22ms/step  Epoch 8/10  938/938 [==============================] - 19s 21ms/step  Epoch 9/10  938/938 [==============================] - 20s 21ms/step  Epoch 10/10  938/938 [==============================] - 19s 21ms/step |

# 5-The layers of each model (including activations):

* Conv2D
* MaxPooling2D
* Flatten
* Dense
* sigmoid” activation “
* Softmax “activation”

# 6-The learning rate used and configuration of the optimizers:

* Optimizer :SGD
* Configuration: learing rate
* Lr:0.009

# 7-You are not required to test learning decays, if you wish to include it, then include it:

* Null

# 8-The optimizer used with its configuration:

* Optimizer :SGD
* Configuration: learing rate

# 9-If you used a dropout layer (you will use it in #7), write where you put it, what dropout rate used and why (your perspective on the location you’ve put it):

* Null

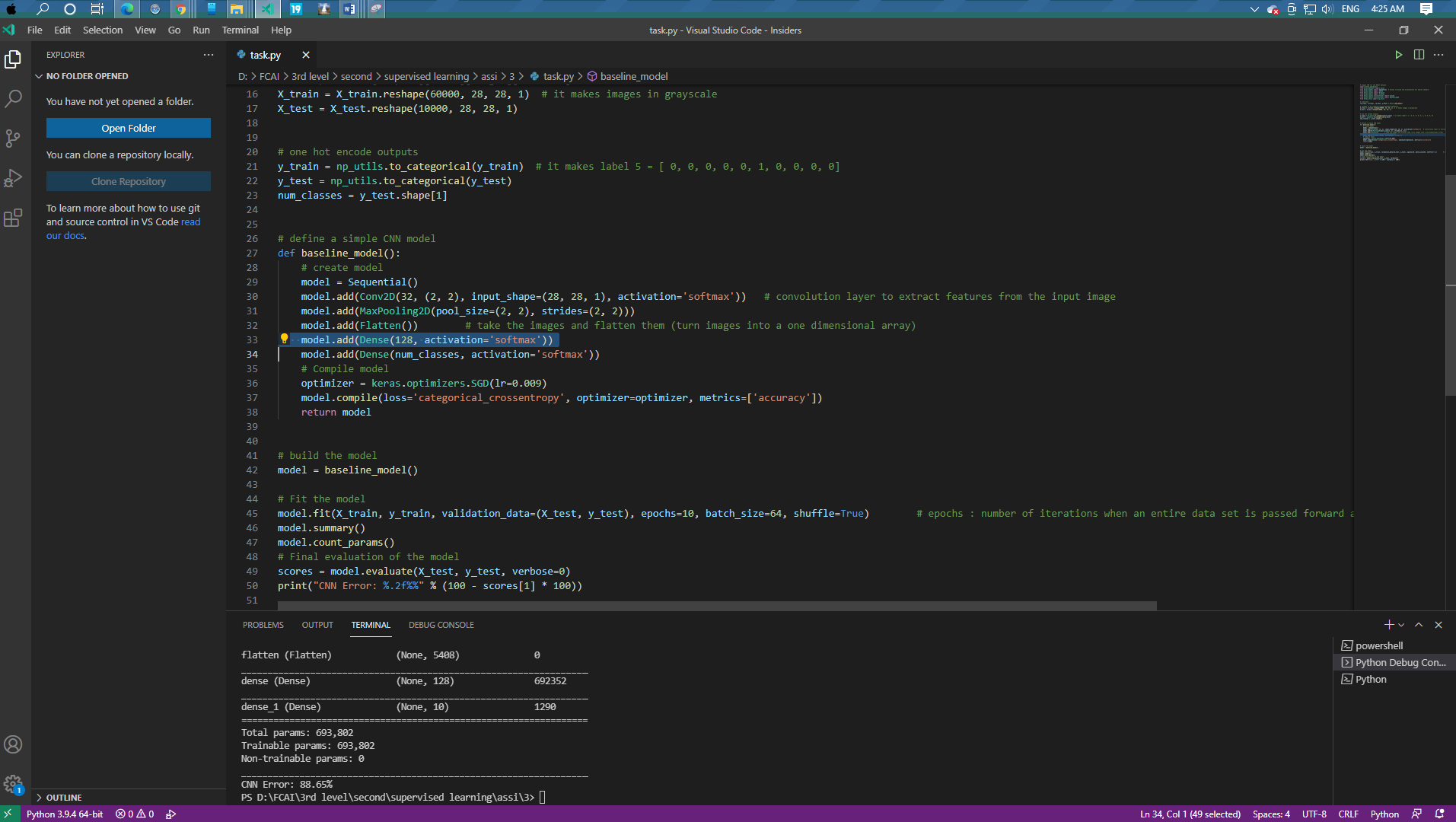
# 10-Write what you observed has changed due to the parameter you changed (according to the above part):

* As you see that the CNN Error increased when we change all activations from relu to sigmoid.

# 0-The code: we will change all activations from sigmoid to softmax with the same previous parameters (Learning rate = 0.009, Epochs = 10, Batch size = 64)

    model.add(Conv2D(32, (2, 2), input\_shape=(28, 28, 1), activation='softmax'))   # convolution layer to extract features from the input image

    model.add(Dense(128, activation='softmax'))



# 1-Final accuracy of the model and the accuracy in the first 5 epoch:

* Accuracy:100-88.65=11.35%

|  |  |
| --- | --- |
| 1 | 0.1125 |
| 2 | 0.1103 |
| 3 | 0.1111 |
| 4 | 0.1137 |
| 5 | 0.1117 |

# 2-The number of parameters in the model:

* Total params: 693,802
* Trainable params: 693,802
* Non-trainable params: 0

# 3&4- The average time to train in each epoch &The average test time in each epoch:

|  |
| --- |
| Epoch 1/10  938/938 [==============================] - 41s 31ms/step  Epoch 2/10  938/938 [==============================] - 28s 30ms/step  Epoch 3/10  938/938 [==============================] - 29s 30ms/step  Epoch 4/10  938/938 [==============================] - 27s 28ms/step  Epoch 5/10  938/938 [==============================] - 26s 27ms/step  Epoch 6/10  938/938 [==============================] - 25s 27ms/step  Epoch 7/10  938/938 [==============================] - 25s 27ms/step  Epoch 8/10  938/938 [==============================] - 25s 27ms/step  Epoch 9/10  938/938 [==============================] - 26s 27ms/step  Epoch 10/10  938/938 [==============================] - 25s 27ms/step |

# 5-The layers of each model (including activations):

* Conv2D
* MaxPooling2D
* Flatten
* Dense
* Softmax “activation”

# 6-The learning rate used and configuration of the optimizers:

* Optimizer :SGD
* Configuration: learing rate
* Lr:0.009

# 7-You are not required to test learning decays, if you wish to include it, then include it:

* Null

# 8-The optimizer used with its configuration:

* Optimizer :SGD
* Configuration: learing rate

# 9-If you used a dropout layer (you will use it in #7), write where you put it, what dropout rate used and why (your perspective on the location you’ve put it):

* Null

# 10-Write what you observed has changed due to the parameter you changed (according to the above part):

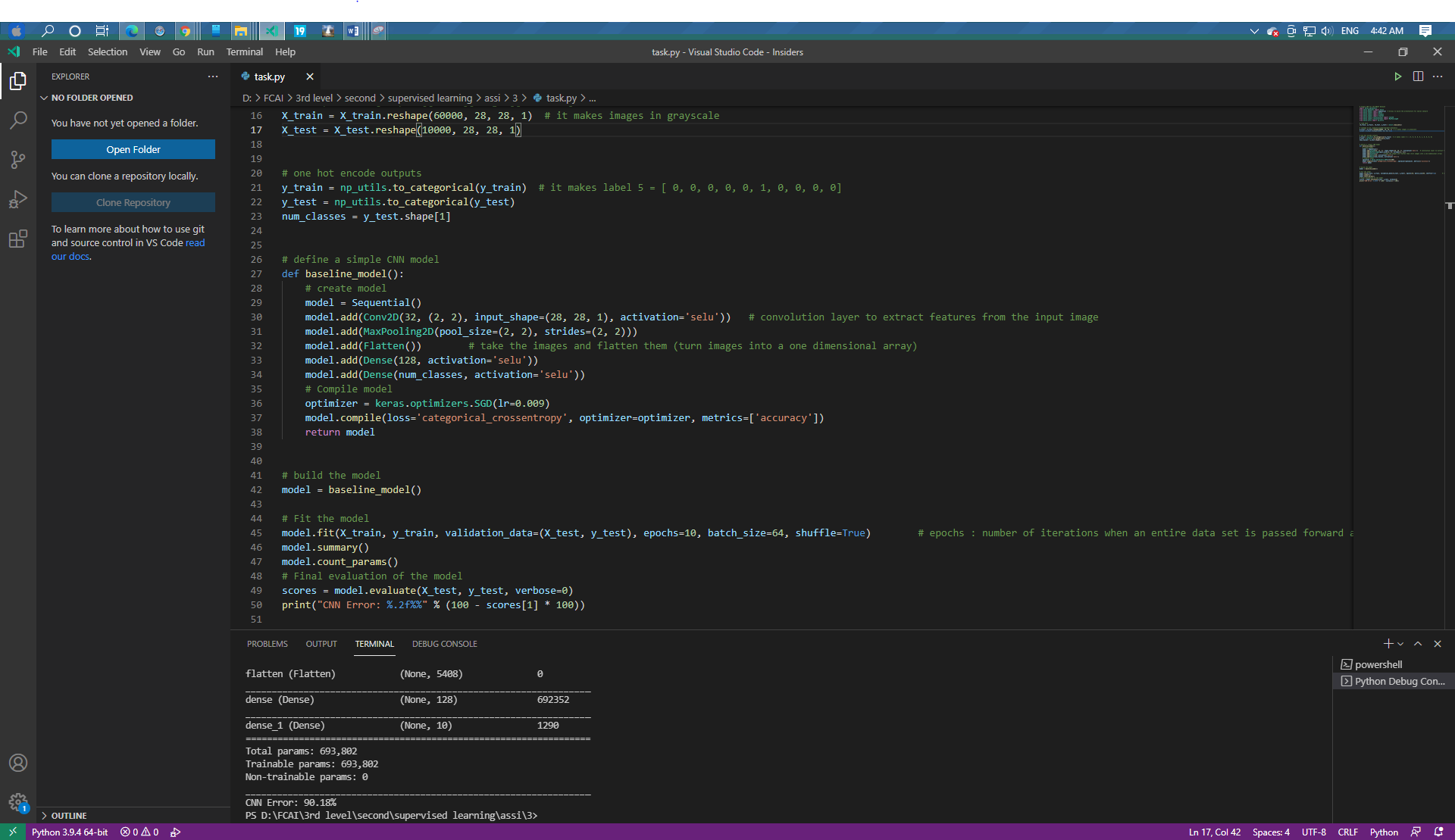
* As you see that the CNN Error increased when we change all activations from sigmoid to softmax.

# 0-The code: we will change all activations from softmax to selu with the same previous parameters (Learning rate = 0.009, Epochs = 10, Batch size = 64).

    model.add(Conv2D(32, (2, 2), input\_shape=(28, 28, 1), activation='selu '))   # convolution layer to extract features from the input image

    model.add(Dense(128, activation='selu '))

    model.add(Dense(num\_classes, activation='selu '))



# 1-Final accuracy of the model and the accuracy in the first 5 epoch:

* Accuracy:100-90.18=9.82%

|  |  |
| --- | --- |
| 1 | 0.2243 |
| 2 | 0.5347 |
| 3 | 0.0968 |
| 4 | 0.0979 |
| 5 | 0.0970 |

# 2-The number of parameters in the model:

* Total params: 693,802
* Trainable params: 693,802
* Non-trainable params: 0

# 3&4- The average time to train in each epoch &The average test time in each epoch:

|  |
| --- |
| Epoch 1/10  938/938 [==============================] - 39s 27ms/step  Epoch 2/10  938/938 [==============================] - 25s 26ms/step  Epoch 3/10  938/938 [==============================] - 23s 24ms/step  Epoch 4/10  938/938 [==============================] - 22s 24ms/step  Epoch 5/10  938/938 [==============================] - 22s 24ms/step  Epoch 6/10  938/938 [==============================] - 25s 27ms/step  Epoch 7/10  938/938 [==============================] - 23s 24ms/step  Epoch 8/10  938/938 [==============================] - 24s 26ms/step  Epoch 9/10  938/938 [==============================] - 22s 24ms/step  Epoch 10/10  938/938 [==============================] - 23s 24ms/step |

# 5-The layers of each model (including activations):

* Conv2D
* MaxPooling2D
* Flatten
* Dense
* Selu “activation”

# 6-The learning rate used and configuration of the optimizers:

* Optimizer :SGD
* Configuration: learing rate
* Lr:0.009

# 7-You are not required to test learning decays, if you wish to include it, then include it:

* Null

# 8-The optimizer used with its configuration:

* Optimizer :SGD
* Configuration: learing rate

# 9-If you used a dropout layer (you will use it in #7), write where you put it, what dropout rate used and why (your perspective on the location you’ve put it):

* Null

# 10-Write what you observed has changed due to the parameter you changed (according to the above part):

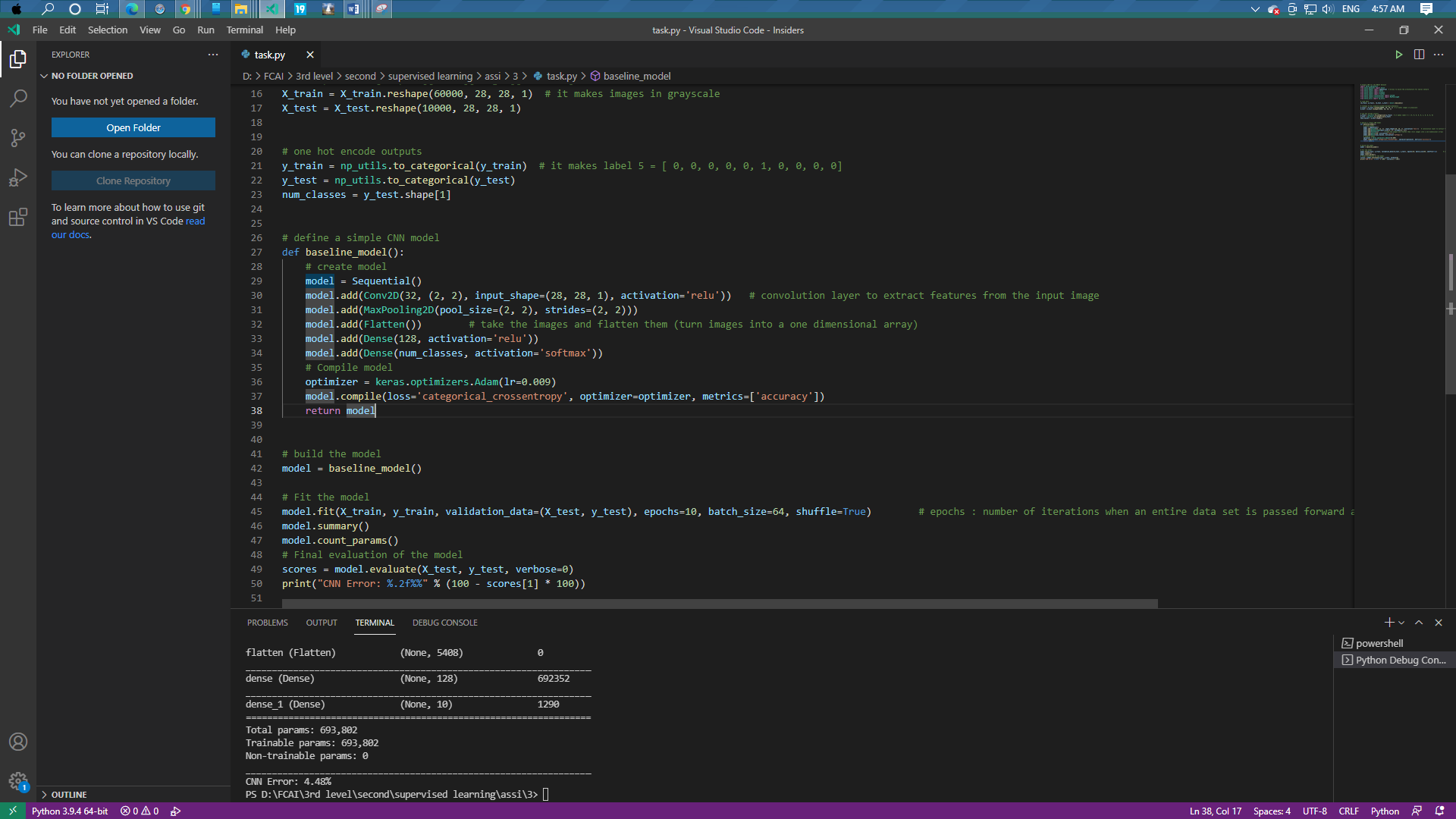
* As you see that the CNN Error increased when we change all activations from softmax to selu.

# 0-The code: we will change the optimizer from SGD to Adam with the same previous parameters (Learning rate = 0.009, Epochs = 10, Batch size = 64, activation = relu).

    model.add(Conv2D(32, (2, 2), input\_shape=(28, 28, 1), activation='relu'))   # convolution layer to extract features from the input image

    model.add(Dense(128, activation='relu'))

    optimizer = keras.optimizers.Adam(lr=0.009)



# 1-Final accuracy of the model and the accuracy in the first 5 epoch:

* Accuracy: 100-4.48=95.52%

|  |  |
| --- | --- |
| 1 | 0.8572 |
| 2 | 0.9656 |
| 3 | 0.9726 |
| 4 | 0.9730 |
| 5 | 0.9715 |

# 2-The number of parameters in the model:

* Total params: 693,802
* Trainable params: 693,802
* Non-trainable params: 0

# 3&4- The average time to train in each epoch &The average test time in each epoch:

|  |
| --- |
| Epoch 1/10  938/938 [==============================] - 37s 25ms/step  Epoch 2/10  938/938 [==============================] - 22s 24ms/step  Epoch 3/10  938/938 [==============================] - 21s 22ms/step  Epoch 4/10  938/938 [==============================] - 24s 26ms/step  Epoch 5/10  938/938 [==============================] - 23s 25ms/step  Epoch 6/10  938/938 [==============================] - 21s 23ms/step  Epoch 7/10  938/938 [==============================] - 22s 24ms/step  Epoch 8/10  938/938 [==============================] - 21s 22ms/step  Epoch 9/10  938/938 [==============================] - 21s 23ms/step  Epoch 10/10  938/938 [==============================] - 22s 24ms/step |

# 5-The layers of each model (including activations):

* Conv2D
* MaxPooling2D
* Flatten
* Dense
* Relu “activation”
* Softmax “activation”

# 6-The learning rate used and configuration of the optimizers:

* Optimizer :Adem
* Configuration: learing rate
* Lr:0.009

# 7-You are not required to test learning decays, if you wish to include it, then include it:

* Null

# 8-The optimizer used with its configuration:

* Optimizer :Adem
* Configuration: learing rate

# 9-If you used a dropout layer (you will use it in #7), write where you put it, what dropout rate used and why (your perspective on the location you’ve put it):

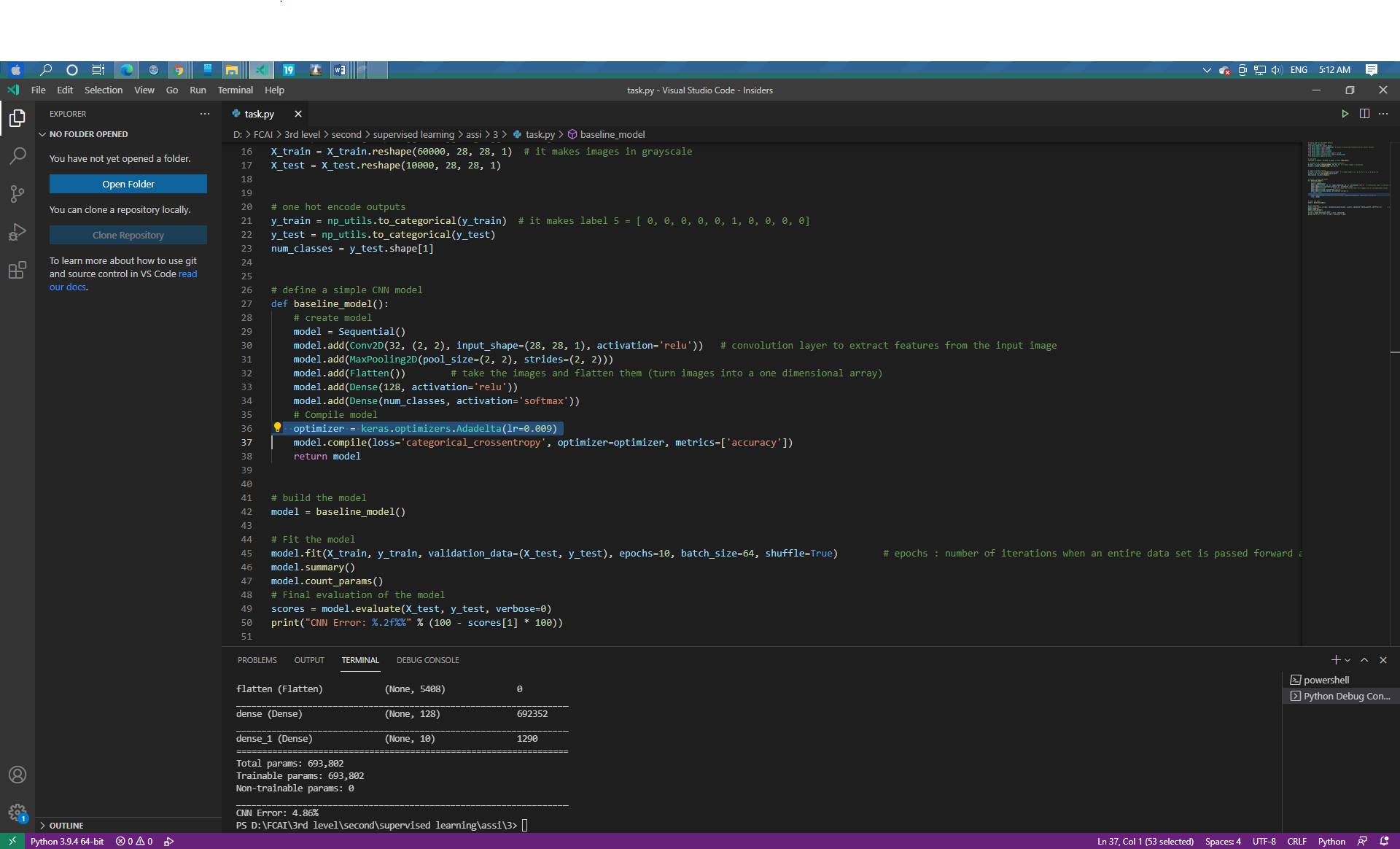
* Null

# 10-Write what you observed has changed due to the parameter you changed (according to the above part):

* As you see that the CNN Error decreased from 5.48% to 4.48% when we change the optimizer from SGD to Adam.

# 0-The code: we will change the optimizer from Adam to Adadelta with the same previous parameters (Learning rate = 0.009, Epochs = 10, Batch size = 64, activation = relu).

    optimizer = keras.optimizers.Adadelta(lr=0.009)



# 1-Final accuracy of the model and the accuracy in the first 5 epoch:

* Accuracy:100-4.86=95.14%

|  |  |
| --- | --- |
| 1 | 0.5077 |
| 2 | 0.8715 |
| 3 | 0.9053 |
| 4 | 0.9222 |
| 5 | 0.9357 |

# 2-The number of parameters in the model:

* Total params: 693,802
* Trainable params: 693,802
* Non-trainable params: 0

# 3&4- The average time to train in each epoch &The average test time in each epoch:

|  |
| --- |
| Epoch 1/10  938/938 [==============================] - 37s 26ms/step  Epoch 2/10  938/938 [==============================] - 26s 28ms/step  Epoch 3/10  938/938 [==============================] - 25s 26ms/step  Epoch 4/10  938/938 [==============================] - 23s 25ms/step  Epoch 5/10  938/938 [==============================] - 22s 24ms/step  Epoch 6/10  938/938 [==============================] - 22s 23ms/step  Epoch 7/10  938/938 [==============================] - 23s 25ms/step  Epoch 8/10  938/938 [==============================] - 22s 24ms/step  Epoch 9/10  938/938 [==============================] - 21s 23ms/step  Epoch 10/10  938/938 [==============================] - 22s 24ms/step |

# 5-The layers of each model (including activations):

* Conv2D
* MaxPooling2D
* Flatten
* Dense
* Relu “activation”
* Softmax “activation”

# 6-The learning rate used and configuration of the optimizers:

* Optimizer : Adadelta
* Configuration: learing rate
* Lr:0.009

# 7-You are not required to test learning decays, if you wish to include it, then include it:

* Null

# 8-The optimizer used with its configuration:

* Optimizer : Adadelta
* Configuration: learing rate

# 9-If you used a dropout layer (you will use it in #7), write where you put it, what dropout rate used and why (your perspective on the location you’ve put it):

* Null

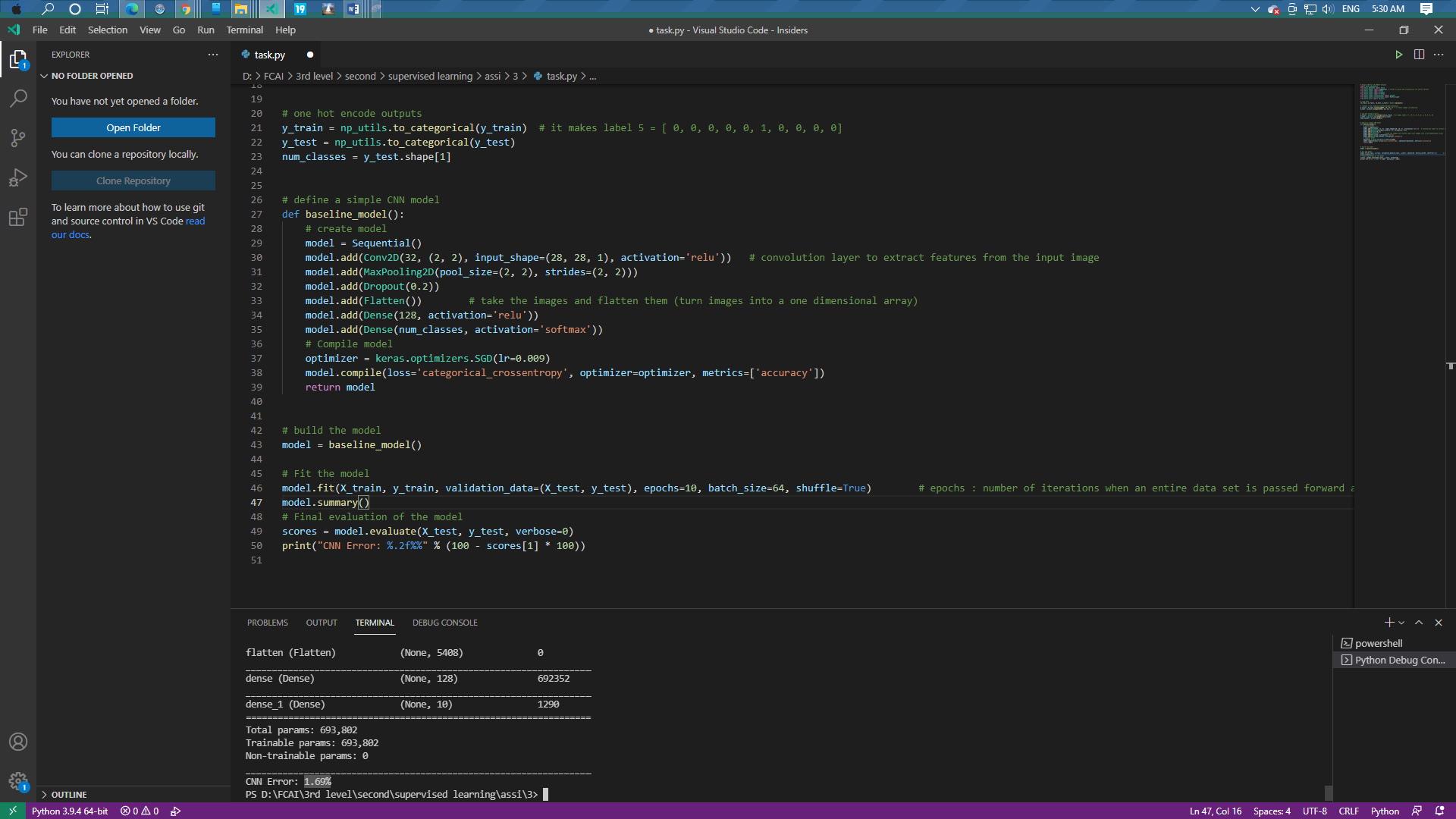
# 10-Write what you observed has changed due to the parameter you changed (according to the above part):

* As you see that the CNN Error increased from 4.48% to 4.86% when we change the optimizer from Adam to Adadelta.

# 0-The code: we will add dropout layer with dropout rate = 0.2 in the model after max pool layer with the same previous parameters (Learning rate = 0.009, Epochs = 10, Batch size = 64, activation = relu, optimizer = SGD).

    model.add(Dropout(0.2))

    optimizer = keras.optimizers.SGD(lr=0.009)



# 1-Final accuracy of the model and the accuracy in the first 5 epoch:

* Accuracy :100-1.69=98.31%

|  |  |
| --- | --- |
| 1 | 0.7953 |
| 2 | 0.9589 |
| 3 | 0.9688 |
| 4 | 0.9749 |
| 5 | 0.9789 |

# 2-The number of parameters in the model:

* Total params: 693,802
* Trainable params: 693,802
* Non-trainable params: 0

# 3&4- The average time to train in each epoch &The average test time in each epoch:

|  |
| --- |
| Epoch 1/10  938/938 [==============================] - 42s 29ms/step  Epoch 2/10  938/938 [==============================] - 26s 27ms/step  Epoch 3/10  938/938 [==============================] - 25s 27ms/step  Epoch 4/10  938/938 [==============================] - 26s 28ms/step  Epoch 5/10  938/938 [==============================] - 25s 27ms/step  Epoch 6/10  938/938 [==============================] - 24s 26ms/step  Epoch 7/10  938/938 [==============================] - 25s 27ms/step  Epoch 8/10  938/938 [==============================] - 26s 27ms/step  Epoch 9/10  938/938 [==============================] - 25s 27ms/step  Epoch 10/10  938/938 [==============================] - 25s 26ms/step |

# 5-The layers of each model (including activations):

* Conv2D
* MaxPooling2D
* Flatten
* Dense
* Dropout
* Relu “activation”
* Softmax “activation”

# 6-The learning rate used and configuration of the optimizers:

* Optimizer : SGD
* Configuration: learing rate
* Lr:0.009

# 7-You are not required to test learning decays, if you wish to include it, then include it:

* Null

# 8-The optimizer used with its configuration:

* Optimizer : SGD
* Configuration: learing rate

# 9-If you used a dropout layer (you will use it in #7), write where you put it, what dropout rate used and why (your perspective on the location you’ve put it):

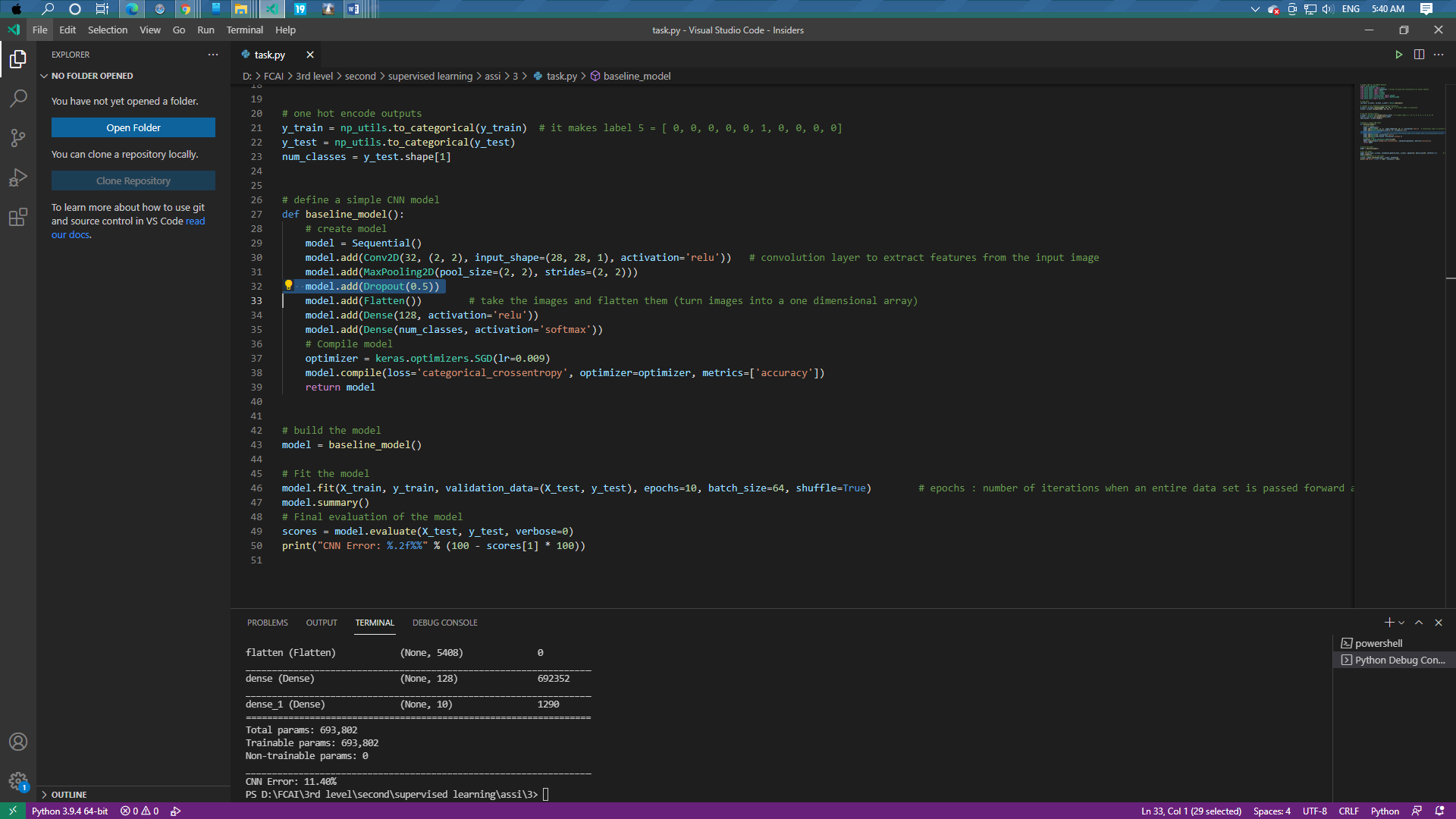
* add dropout layer with dropout rate = 0.2 in the model after max pool layer

# 10-Write what you observed has changed due to the parameter you changed (according to the above part):

* As you see that the CNN Error decreased from 2.04% to 1.69%when we add dropout layer after max pool layer with dropout rate = 0.2.

# 0-The code: we will add dropout layer with dropout rate = 0.5 in the model after max pool layer with the same previous parameters (Learning rate = 0.009, Epochs = 10, Batch size = 64, activation = relu, optimizer = SGD).

    model.add(Dropout(0.5))



# 1-Final accuracy of the model and the accuracy in the first 5 epoch:

* Accuracy:100-11.40=88.60%

|  |  |
| --- | --- |
| 1 | 0.1264 |
| 2 | 0.5525 |
| 3 | 0.6747 |
| 4 | 0.6850 |
| 5 | 0.6908 |

# 2-The number of parameters in the model:

* Total params: 693,802
* Trainable params: 693,802
* Non-trainable params: 0

# 3&4- The average time to train in each epoch &The average test time in each epoch:

|  |
| --- |
| Epoch 1/10  938/938 [==============================] - 39s 28ms/step  Epoch 2/10  938/938 [==============================] - 27s 29ms/step  Epoch 3/10  938/938 [==============================] - 24s 26ms/step  Epoch 4/10  938/938 [==============================] - 25s 27ms/step  Epoch 5/10  938/938 [==============================] - 27s 29ms/step  Epoch 6/10  938/938 [==============================] - 24s 26ms/step  Epoch 7/10  938/938 [==============================] - 24s 26ms/step  Epoch 8/10  938/938 [==============================] - 25s 26ms/step  Epoch 9/10  938/938 [==============================] - 26s 28ms/step  Epoch 10/10  938/938 [==============================] - 26s 28ms/step |

# 5-The layers of each model (including activations):

* Conv2D
* MaxPooling2D
* Flatten
* Dense
* Dropout
* Relu “activation”
* Softmax “activation”

# 6-The learning rate used and configuration of the optimizers:

* Optimizer : SGD
* Configuration: learing rate
* Lr:0.009

# 7-You are not required to test learning decays, if you wish to include it, then include it:

* Null

# 8-The optimizer used with its configuration:

* Optimizer : SGD
* Configuration: learing rate

# 9-If you used a dropout layer (you will use it in #7), write where you put it, what dropout rate used and why (your perspective on the location you’ve put it):

* add dropout layer with dropout rate = 0.5 in the model after max pool layer

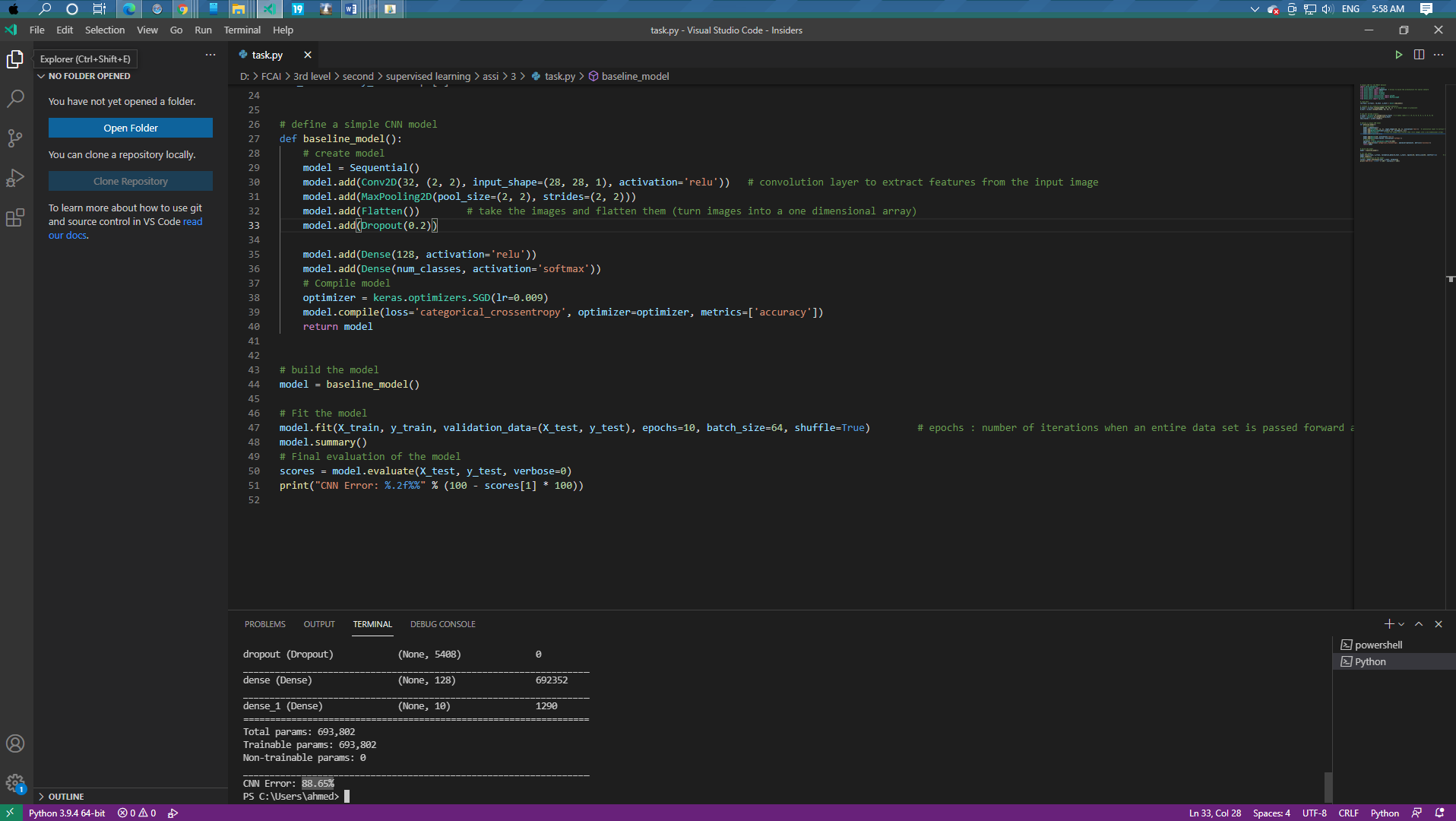
# 10-Write what you observed has changed due to the parameter you changed (according to the above part):

* As you see that the CNN Error increased from 2.04% to 11.40% when we add dropout layer after max pool layer with dropout rate = 0.5.

# 0-The code: we will add dropout layer with dropout rate = 0.2 in the model after flatten layer with the same previous parameters (Learning rate = 0.009, Epochs = 10, Batch size = 64, activation = relu, optimizer = SGD).

 model.add(Flatten())

 model.add(Dropout(0.2))



# 1-Final accuracy of the model and the accuracy in the first 5 epoch:

* Accuracy: 100- 88.65%=11.35%

|  |  |
| --- | --- |
| 1 | 0.1065 |
| 2 | 0.1120 |
| 3 | 0.1118 |
| 4 | 0.1108 |
| 5 | 0.1133 |

# 2-The number of parameters in the model:

• Total params: 693,802

• Trainable params: 693,802

• Non-trainable params: 0

# 3&4- The average time to train in each epoch &The average test time in each epoch:

|  |
| --- |
| Epoch 1/10  938/938 [==============================] - 36s 25ms/step  Epoch 2/10  938/938 [==============================] - 24s 25ms/step  Epoch 3/10  938/938 [==============================] - 28s 30ms/step  Epoch 4/10  938/938 [==============================] - 27s 28ms/step  Epoch 5/10  938/938 [==============================] - 26s 28ms/step  Epoch 6/10  938/938 [==============================] - 25s 27ms/step  Epoch 7/10  938/938 [==============================] - 25s 26ms/step  Epoch 8/10  938/938 [==============================] - 26s 27ms/step  Epoch 9/10  938/938 [==============================] - 26s 27ms/step  Epoch 10/10  938/938 [==============================] - 26s 28ms/step |

# 5-The layers of each model (including activations):

* Conv2D
* MaxPooling2D
* Flatten
* Dense
* Dropout
* Relu “activation”
* Softmax “activation”

# 6-The learning rate used and configuration of the optimizers:

* Optimizer : SGD
* Configuration: learing rate
* Lr:0.009

# 7-You are not required to test learning decays, if you wish to include it, then include it:

* Null

# 8-The optimizer used with its configuration:

* Optimizer : SGD
* Configuration: learing rate

# 9-If you used a dropout layer (you will use it in #7), write where you put it, what dropout rate used and why (your perspective on the location you’ve put it):

# add dropout layer with dropout rate = 0.2 in the model after flatten layer

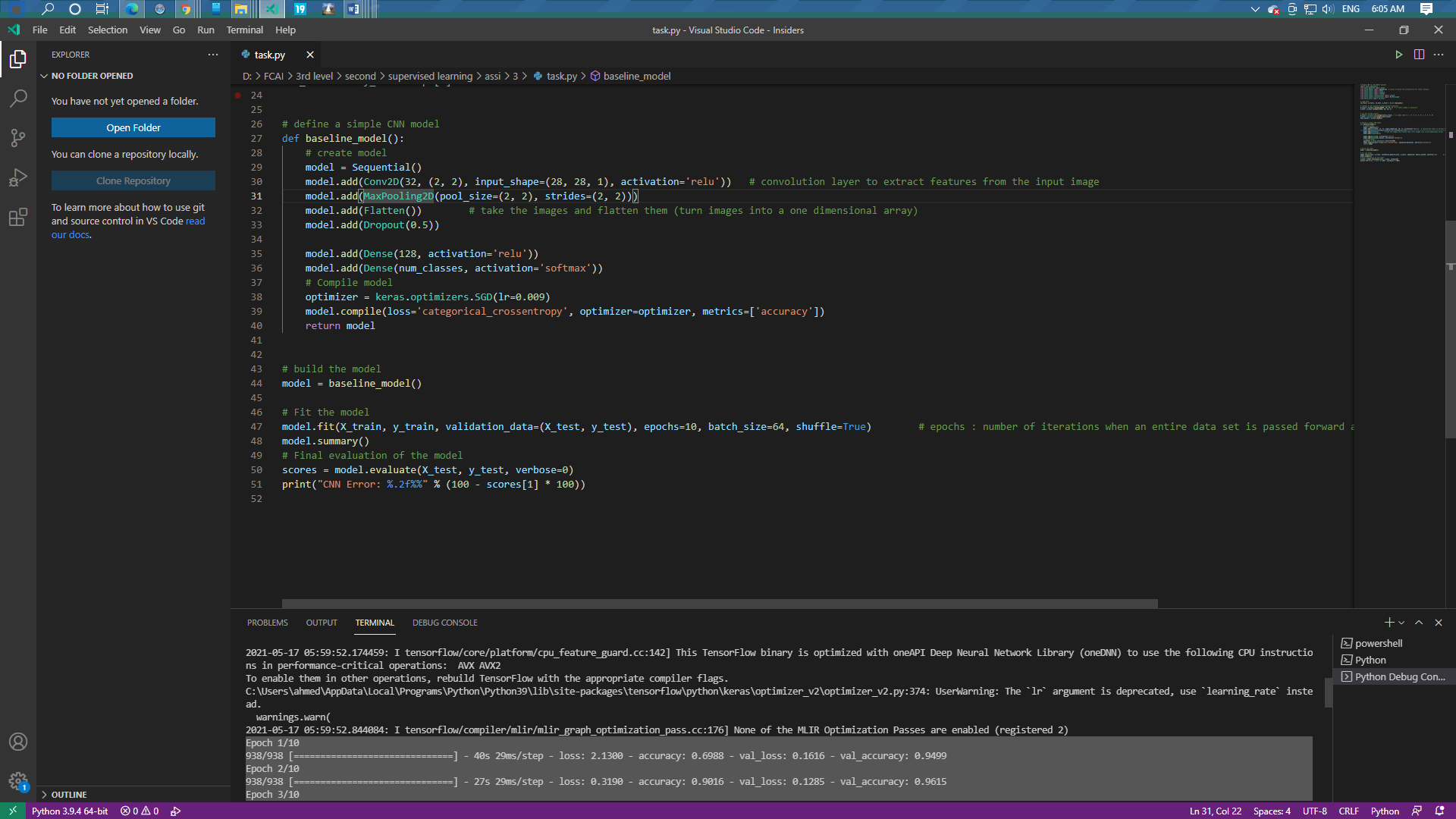
# 10-Write what you observed has changed due to the parameter you changed (according to the above part):

* As you see that the CNN Error increased from 2.04% to 88.65% when we add dropout layer after flatten layer with dropout rate = 0.2.

# 0-The code: we will add dropout layer with dropout rate = 0.5 in the model after flatten layer with the same previous parameters (Learning rate = 0.009, Epochs = 10, Batch size = 64, activation = relu, optimizer = SGD).

 model.add(Flatten())

    model.add(Dropout(0.5))



# 1-Final accuracy of the model and the accuracy in the first 5 epoch:

|  |  |
| --- | --- |
| 1 | 0.6988 |
| 2 | 0.9016 |
| 3 | 0.9156 |
| 4 | 0.9263 |
| 5 | 0.9333 |

# 2-The number of parameters in the model:

* Total params: 693,802
* Trainable params: 693,802
* Non-trainable params: 0

# 3&4- The average time to train in each epoch &The average test time in each epoch:

|  |
| --- |
| Epoch 1/10  938/938 [==============================] - 40s 29ms/step  Epoch 2/10  938/938 [==============================] - 27s 29ms/step  Epoch 3/10  938/938 [==============================] - 26s 28ms/step  Epoch 4/10  938/938 [==============================] - 25s 26ms/step  Epoch 5/10  938/938 [==============================] - 26s 27ms/step  Epoch 6/10  938/938 [==============================] - 27s 29ms/step  Epoch 7/10  938/938 [==============================] - 27s 29ms/step  Epoch 8/10  938/938 [==============================] - 27s 29ms/step  Epoch 9/10  938/938 [==============================] - 26s 28ms/step  Epoch 10/10  938/938 [==============================] - 26s 28ms/step |

# 5-The layers of each model (including activations):

* Conv2D
* MaxPooling2D
* Flatten
* Dense
* Dropout
* Relu “activation”
* Softmax “activation”

# 6-The learning rate used and configuration of the optimizers:

* Optimizer : SGD
* Configuration: learing rate
* Lr:0.009

# 7-You are not required to test learning decays, if you wish to include it, then include it:

* Null

# 8-The optimizer used with its configuration:

* Optimizer : SGD
* Configuration: learing rate

# 9-If you used a dropout layer (you will use it in #7), write where you put it, what dropout rate used and why (your perspective on the location you’ve put it):

# add dropout layer with dropout rate = 0.5 in the model after flatten layer

# 10-Write what you observed has changed due to the parameter you changed (according to the above part):

* As you see that the CNN Error increased from 2.04% to 2.25% when we add dropout layer after flatten layer with dropout rate = 0.5.