Faculty of Computers and Artificial Intelligence
Cairo University

Supervised Learning
AI322

Assignment 3
(CNN Report)

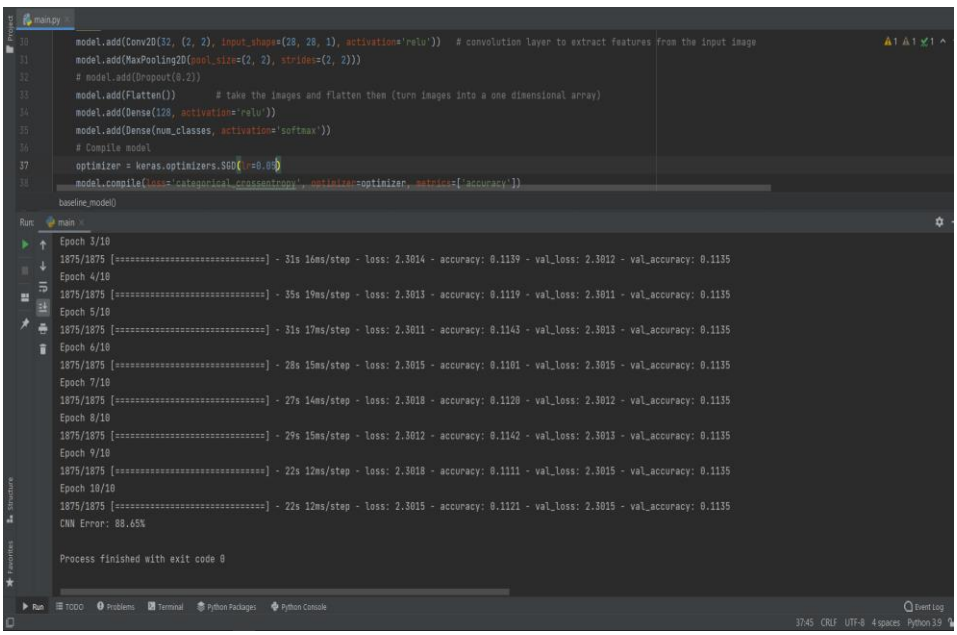
Name	ID
Loai Gamal Mohamed	20180206
Ahmed Kadry Abd El-Shafy	20180018

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 neura
l 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 i
nto 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_si
ze=32, shuffle=True)        # epochs : number of iterations when an entire data s
et 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))
```



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):

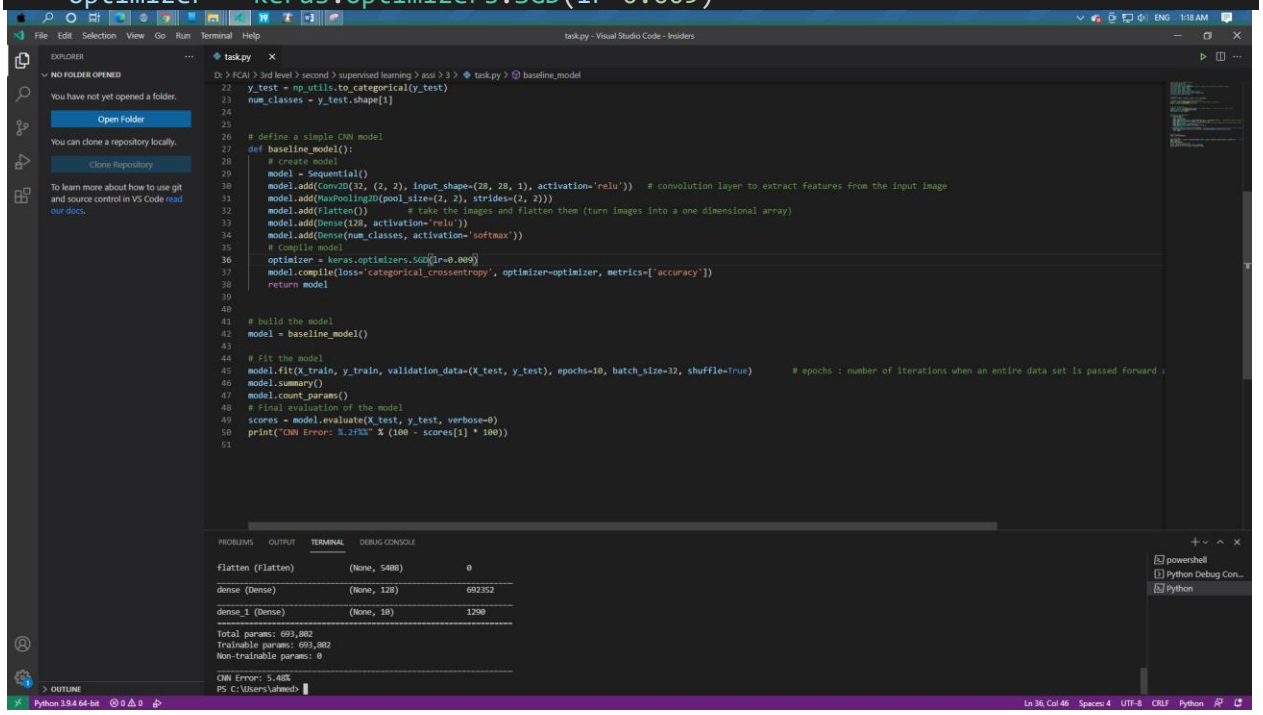
- Null

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 to 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):

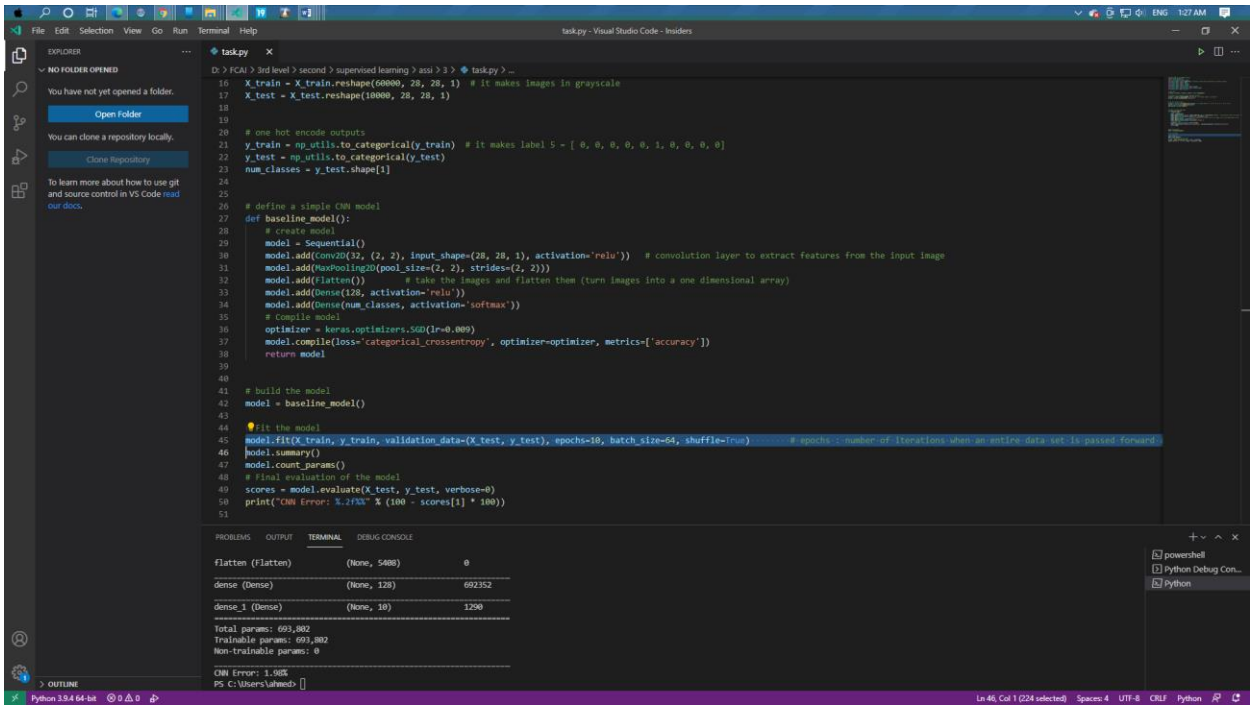
- Null

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%**.

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_si
ze=64, shuffle=True)        # epochs : number of iterations when an entire data s
et 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):

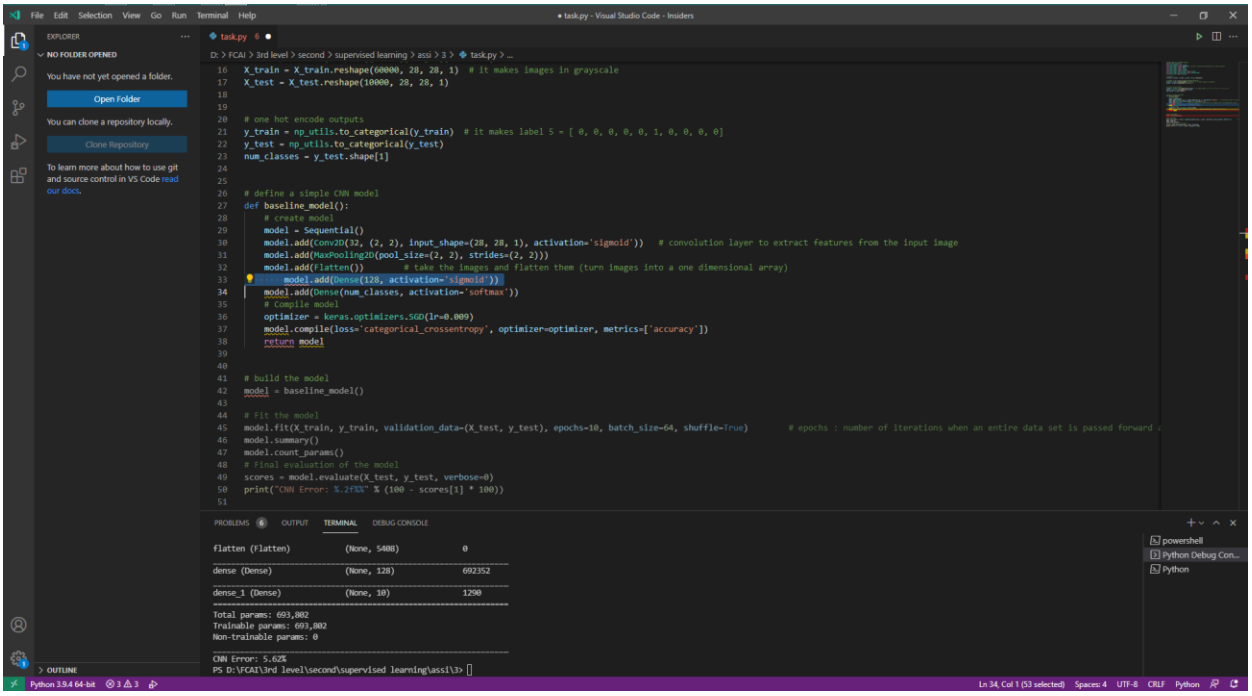
- Null

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

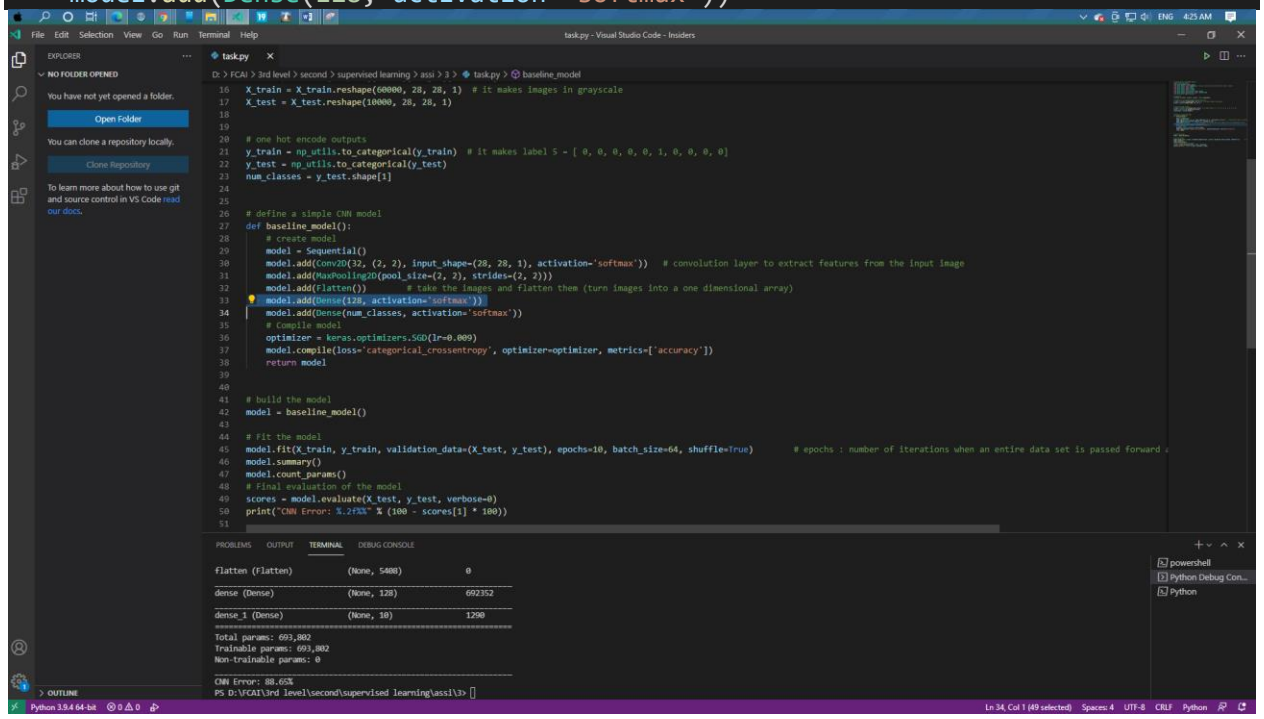
10-Write what you observed has changed due to the parameter you changed (according to the above part):

- 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 relu 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):

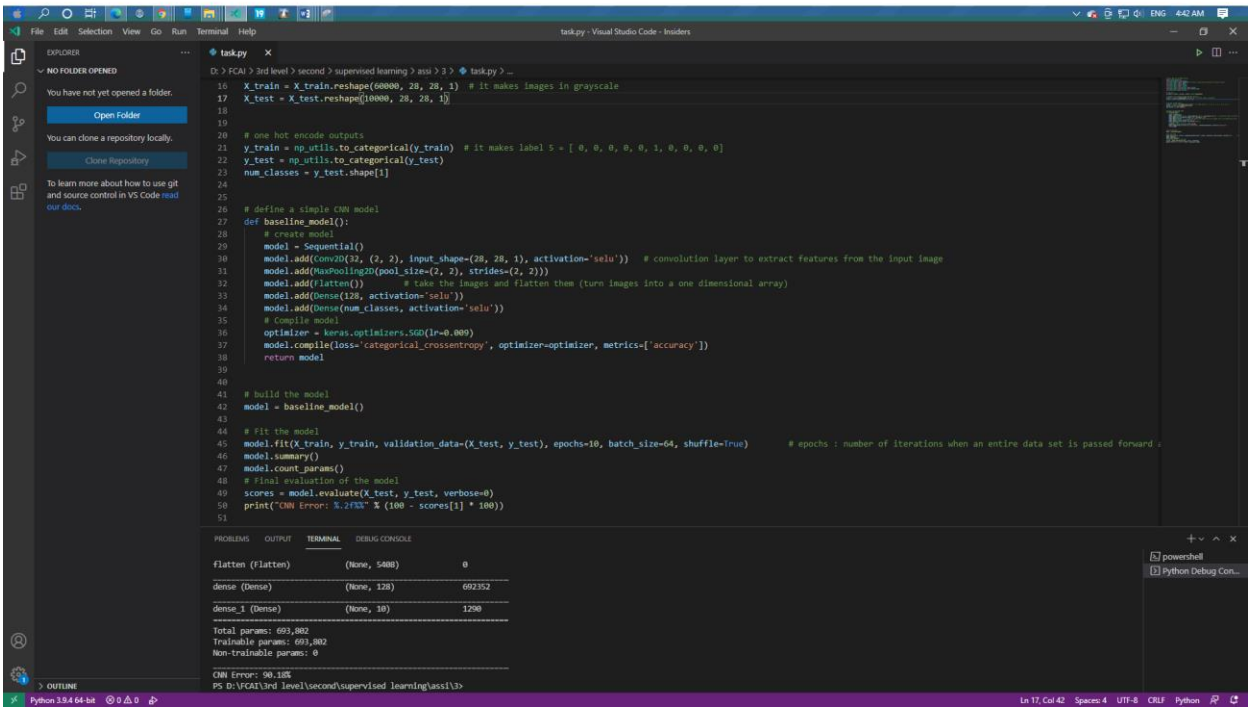
- Null

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 **relu** to **softmax**.

0-The code: we will change all activations from relu 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

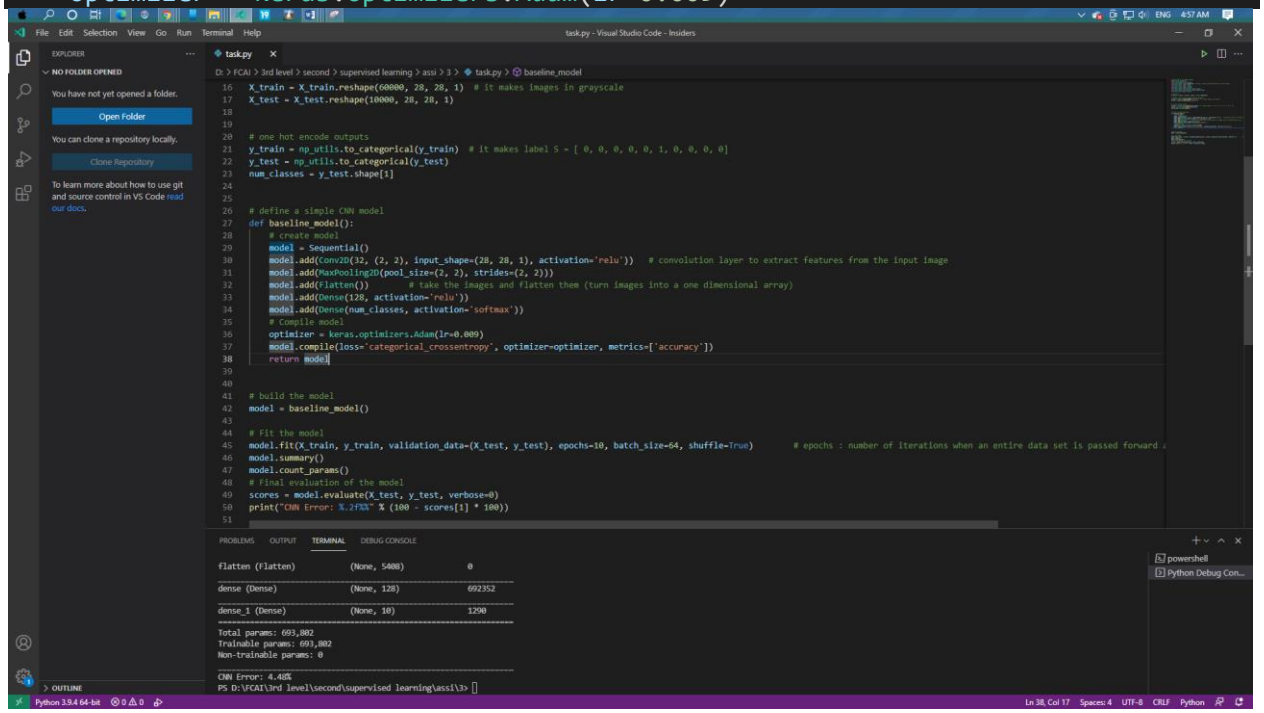
10-Write what you observed has changed due to the parameter you changed (according to the above part):

- As you see that the CNN Error increased when we change all activations from **relu** 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):

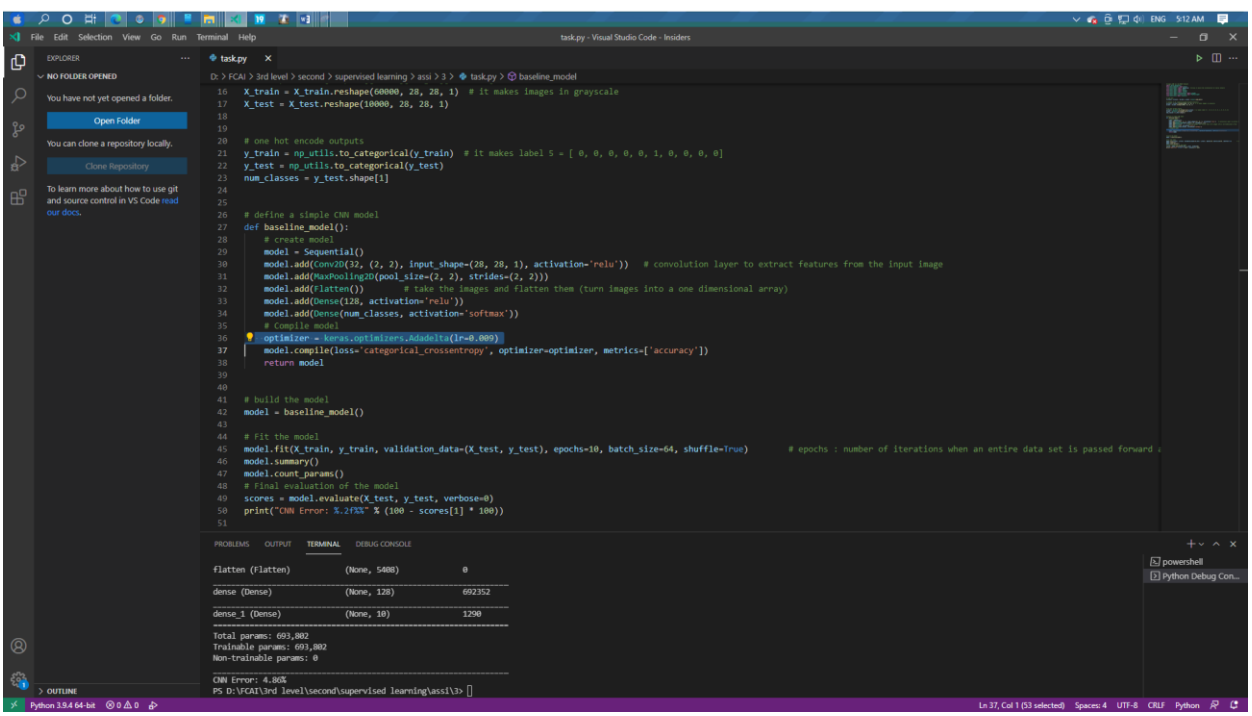
- Null

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 increased when we change the optimizer from **SGD** to **Adam**.

0-The code: we will change the optimizer from SGD 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):

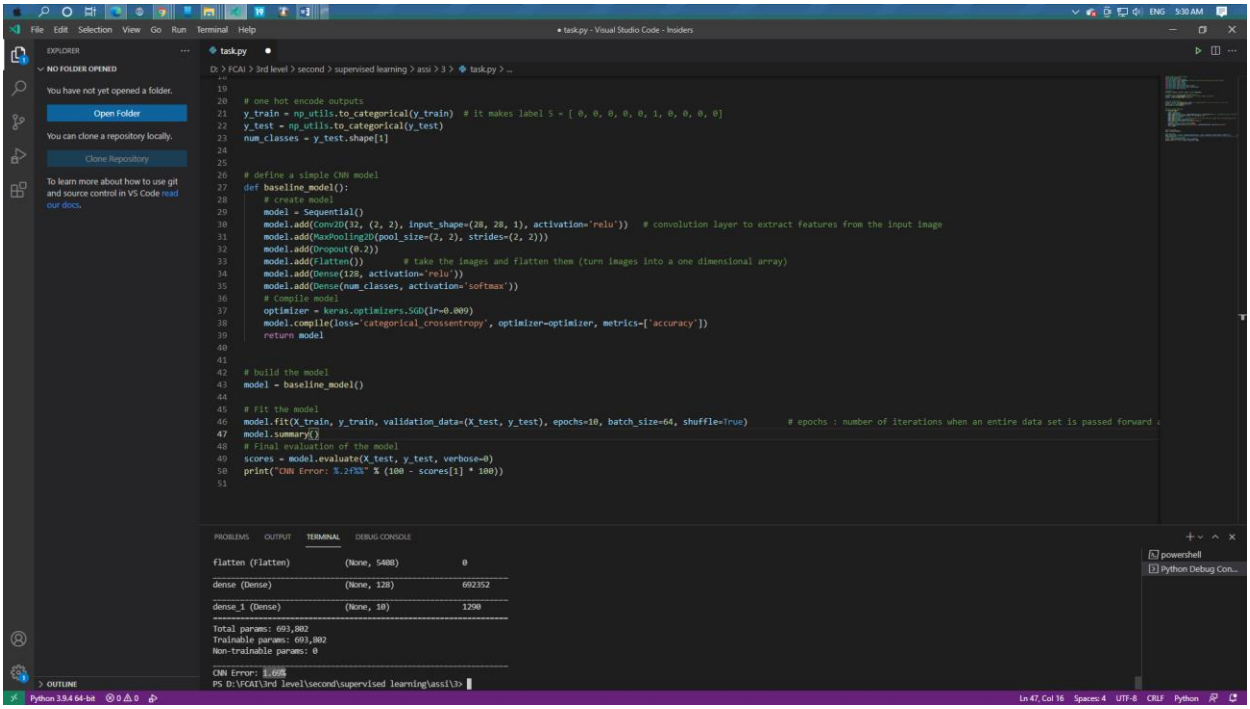
- Null

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 when we change the optimizer from **SGD** 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):

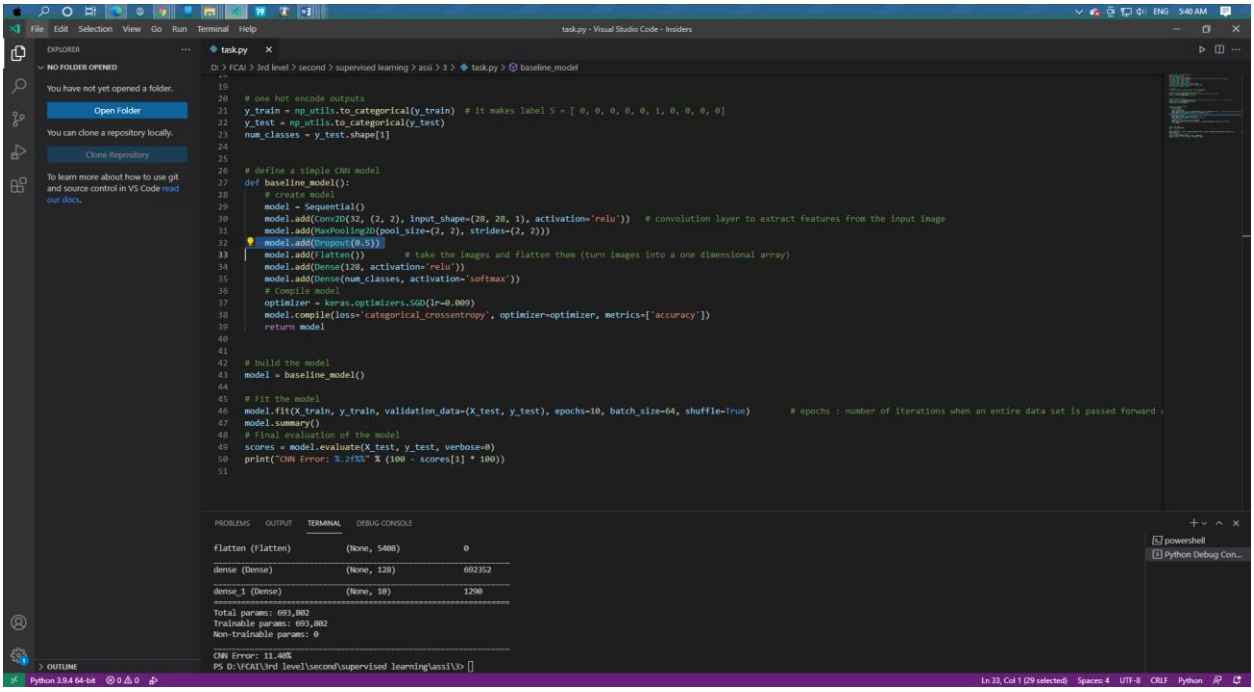
- add dropout layer with dropout rate = 0.2 in the model after max pool layer

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):

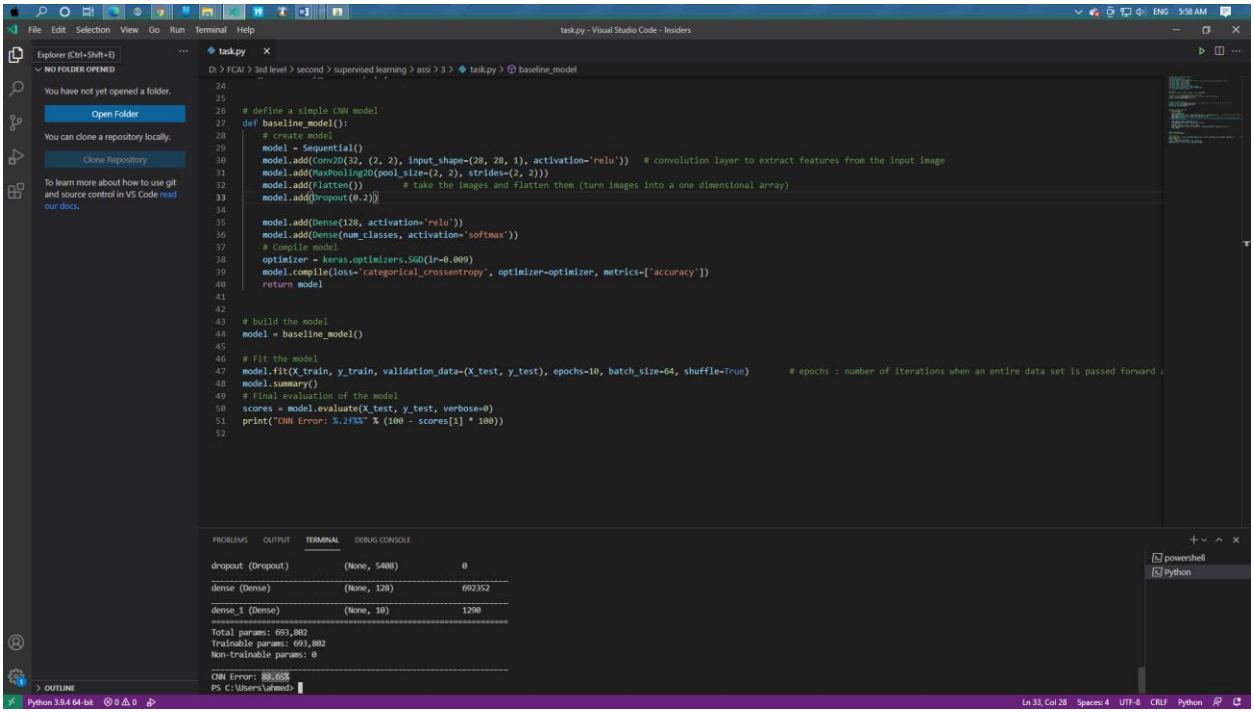
- add dropout layer with dropout rate = 0.5 in the model after max pool layer

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):

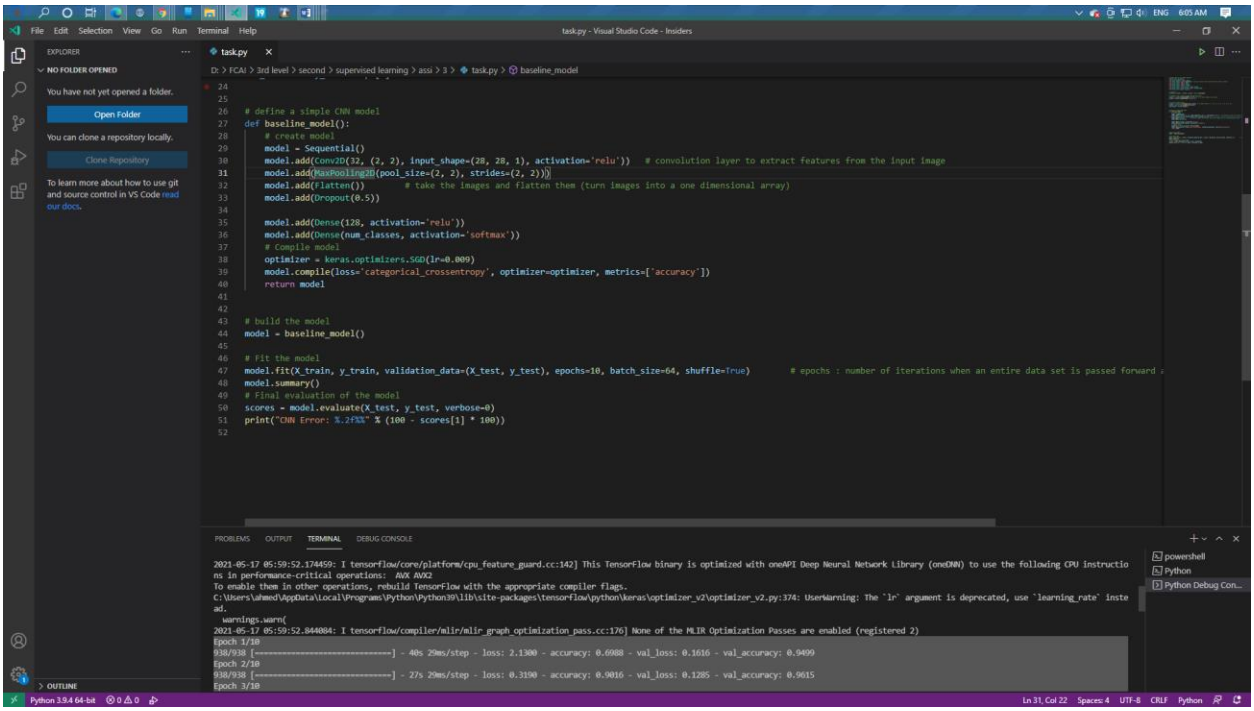
- add dropout layer with dropout rate = 0.2 in the model after flatten layer

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):

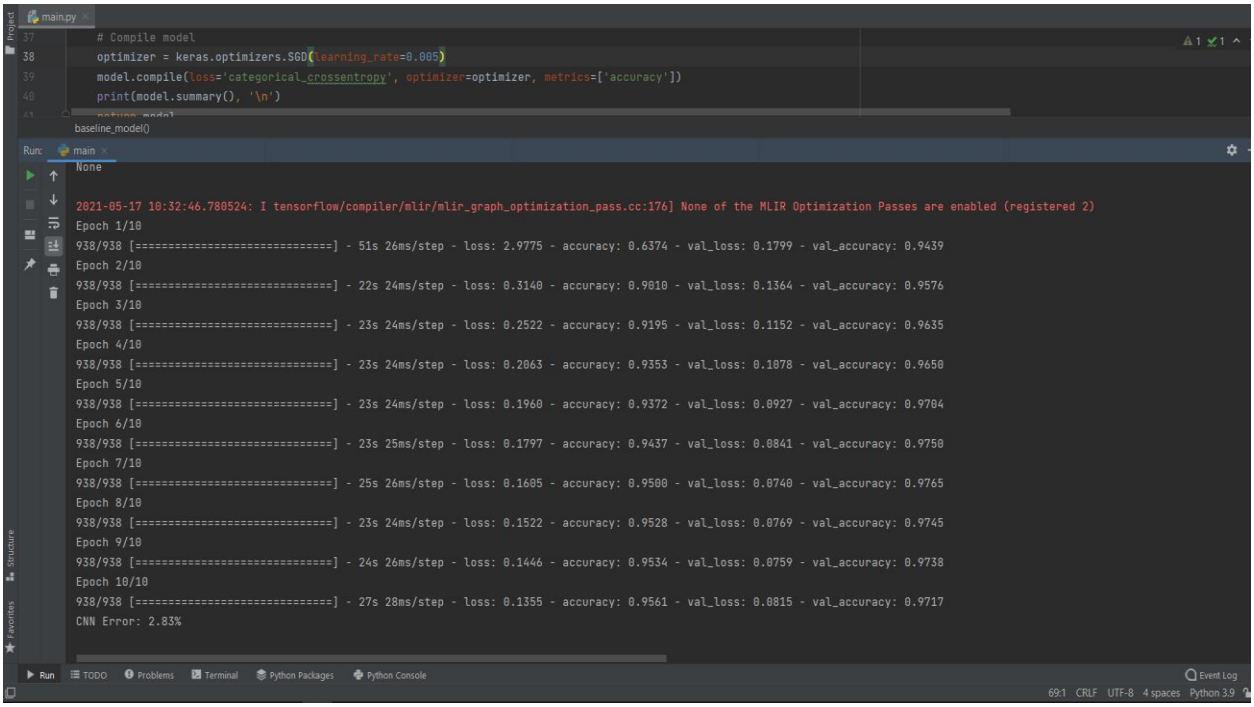
- add dropout layer with dropout rate = 0.5 in the model after flatten layer

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.

0-The code: we will change the learning rate from 0.009 to 0.005 with the same previous parameters (Epochs = 10, Batch size = 64, activation = relu, optimizer = SGD).

```
optimizer = keras.optimizers.SGD(learning_rate=0.005)
```



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

- Accuracy: 100- 2.83%=97.17%

1	0.6374
2	0.9010
3	0.9195
4	0.9353
5	0.9372

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 [==============================] - 51s                    26ms/step
Epoch 2/10
938/938 [==============================] - 22s                    24ms/step
Epoch 3/10
938/938 [==============================] - 23s                    24ms/step
Epoch 4/10
938/938 [==============================] - 23s                    24ms/step
Epoch 5/10
938/938 [==============================] - 23s                    24ms/step
Epoch 6/10
938/938 [==============================] - 23s                    25ms/step
Epoch 7/10
938/938 [==============================] - 25s                    26ms/step
Epoch 8/10
938/938 [==============================] - 23s                    24ms/step
Epoch 9/10
938/938 [==============================] - 24s                    26ms/step
Epoch 10/10
938/938 [==============================] - 27s                    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:0.005

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):

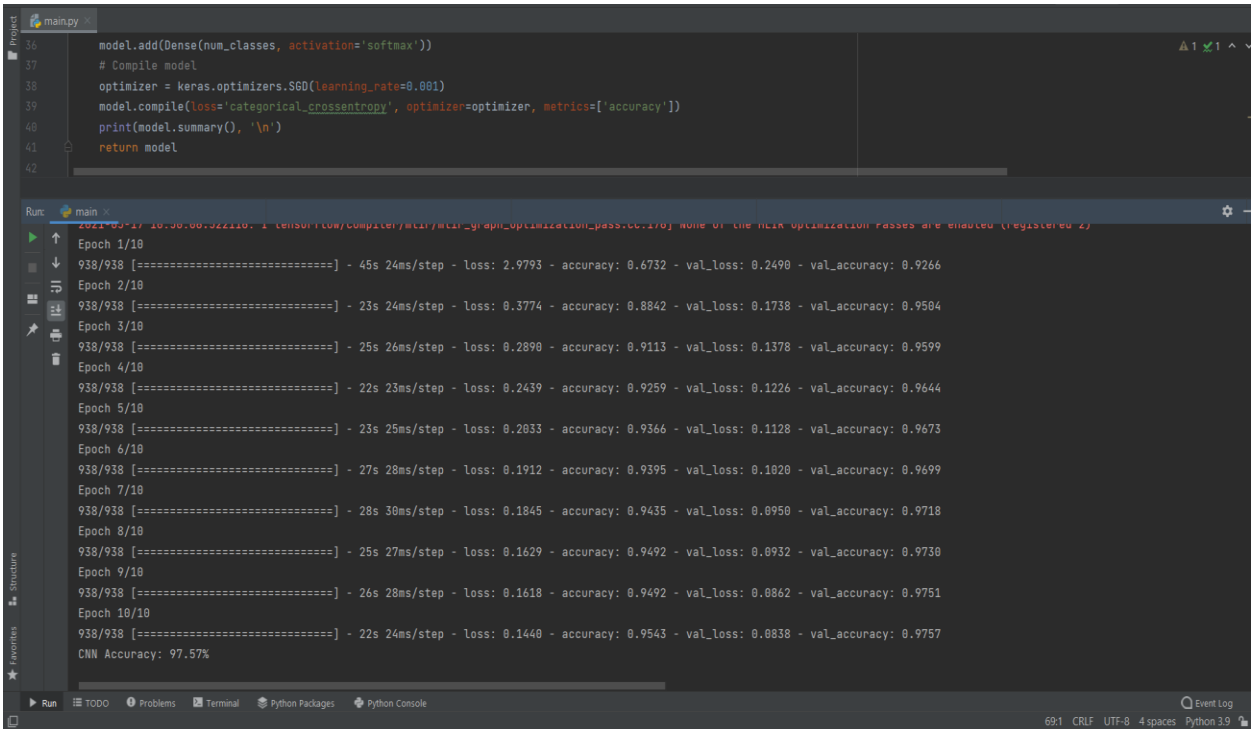
- Null

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 when we change the learning rate from **0.009** to **0.005**

0-The code: we will change the learning rate from 0.005 to 0.001 with the same previous parameters (Epochs = 10, Batch size = 64, activation = relu, optimizer = SGD).

```
optimizer = keras.optimizers.SGD(learning_rate=0.001)
```



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

- Accuracy: 97.57%

1	0.6732
2	0.8842
3	0.9113
4	0.9259
5	0.9366

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 [==============================] - 45s                    24ms/step
Epoch 2/10
938/938 [==============================] - 23s                    24ms/step
Epoch 3/10
938/938 [==============================] - 25s                    26ms/step
Epoch 4/10
938/938 [==============================] - 22s                    23ms/step
Epoch 5/10
938/938 [==============================] - 23s                    25ms/step
Epoch 6/10
938/938 [==============================] - 27s                    28ms/step
Epoch 7/10
938/938 [==============================] - 28s                    30ms/step
Epoch 8/10
938/938 [==============================] - 25s                    27ms/step
Epoch 9/10
938/938 [==============================] - 26s                    28ms/step
Epoch 10/10
938/938 [==============================] - 22s                    24ms/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:0.001

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):

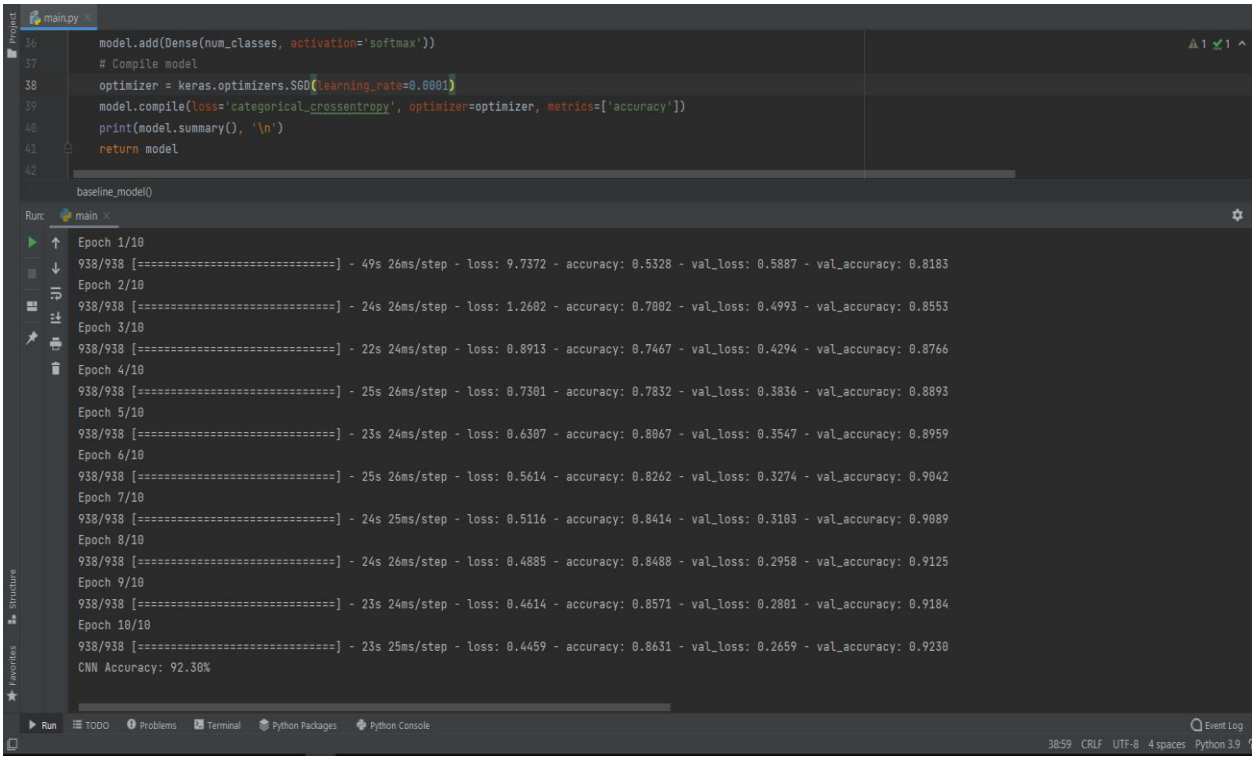
- Null

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 when we change the learning rate from **0.005** to **0.001**

0-The code: we will change the learning rate from 0.001 to 0.0001 with the same previous parameters (Epochs = 10, Batch size = 64, activation = relu, optimizer = SGD).

```
optimizer = keras.optimizers.SGD(learning_rate=0.0001)
```



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

- Accuracy: 92.30%

1	0.5328
2	0.7002
3	0.7467
4	0.7832
5	0.8067

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 [==============================] - 49s                    26ms/step
Epoch 2/10
938/938 [==============================] - 24s                    26ms/step
Epoch 3/10
938/938 [==============================] - 22s                    24ms/step
Epoch 4/10
938/938 [==============================] - 25s                    26ms/step
Epoch 5/10
938/938 [==============================] - 23s                    24ms/step
Epoch 6/10
938/938 [==============================] - 25s                    26ms/step
Epoch 7/10
938/938 [==============================] - 24s                    25ms/step
Epoch 8/10
938/938 [==============================] - 24s                    26ms/step
Epoch 9/10
938/938 [==============================] - 23s                    24ms/step
Epoch 10/10
938/938 [==============================] - 23s                    25ms/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:0.0001

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):

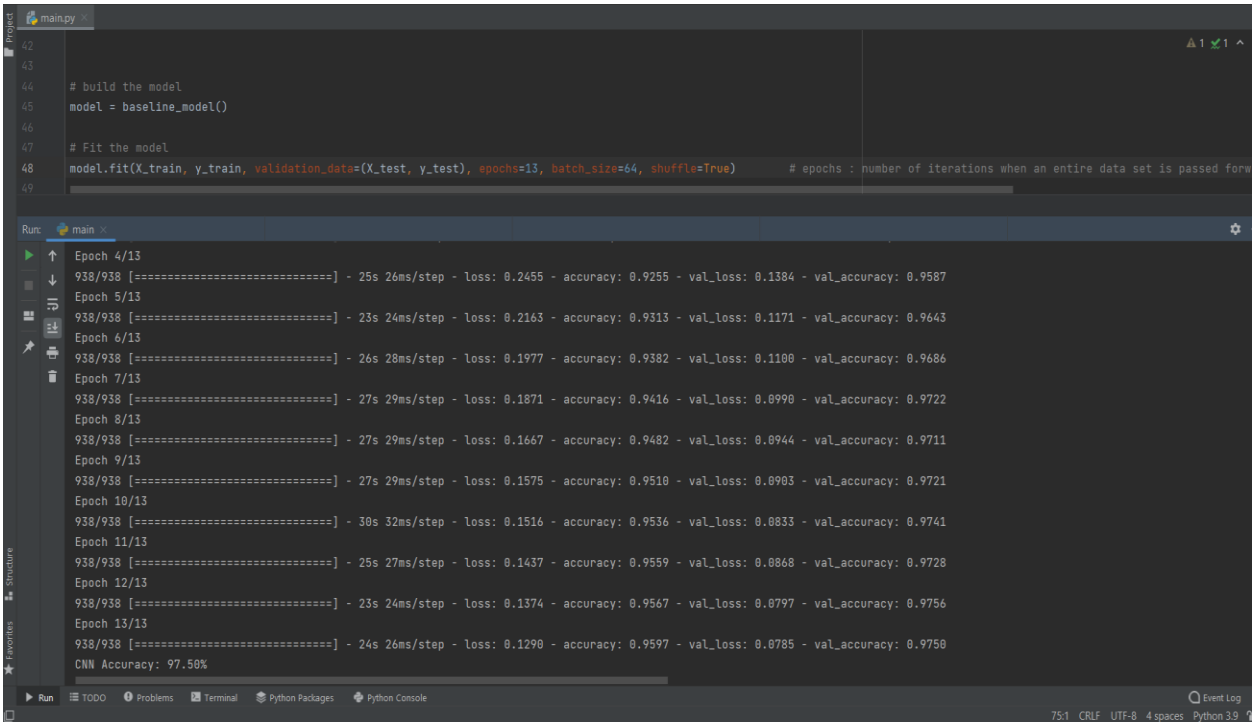
- Null

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 the learning rate from **0.001** to **0.0001**

0-The code: we will change the number of Epochs from 10 to 13 with the same previous parameters (Learning Rate = 0.001, Batch size = 64, activation = relu, optimizer = SGD).

```
model.fit(X_train, y_train, validation_data=(X_test, y_test), epochs=13,
batch_size=64, shuffle=True)
```



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

- Accuracy: 97.50%

1	0.6702
2	0.8743
3	0.9105
4	0.9255
5	0.9313

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 1/13
938/938 [==============================] - 55s                    24ms/step
Epoch 2/13
938/938 [==============================] - 22s                    24ms/step
Epoch 3/13
938/938 [==============================] - 23s                    26ms/step
Epoch 4/13
938/938 [==============================] - 25s                    23ms/step
Epoch 5/13
938/938 [==============================] - 23s                    25ms/step
Epoch 6/13
938/938 [==============================] - 26s                    28ms/step
Epoch 7/13
938/938 [==============================] - 27s                    30ms/step
Epoch 8/13
938/938 [==============================] - 27s                    27ms/step
Epoch 9/13
938/938 [==============================] - 27s                    28ms/step
Epoch 10/13
938/938 [==============================] - 30s                    24ms/step
Epoch 11/13
938/938 [==============================] - 25s                    24ms/step
Epoch 12/13
938/938 [==============================] - 23s                    24ms/step
Epoch 13/13
938/938 [==============================] - 24s                    24ms/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:0.001

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):

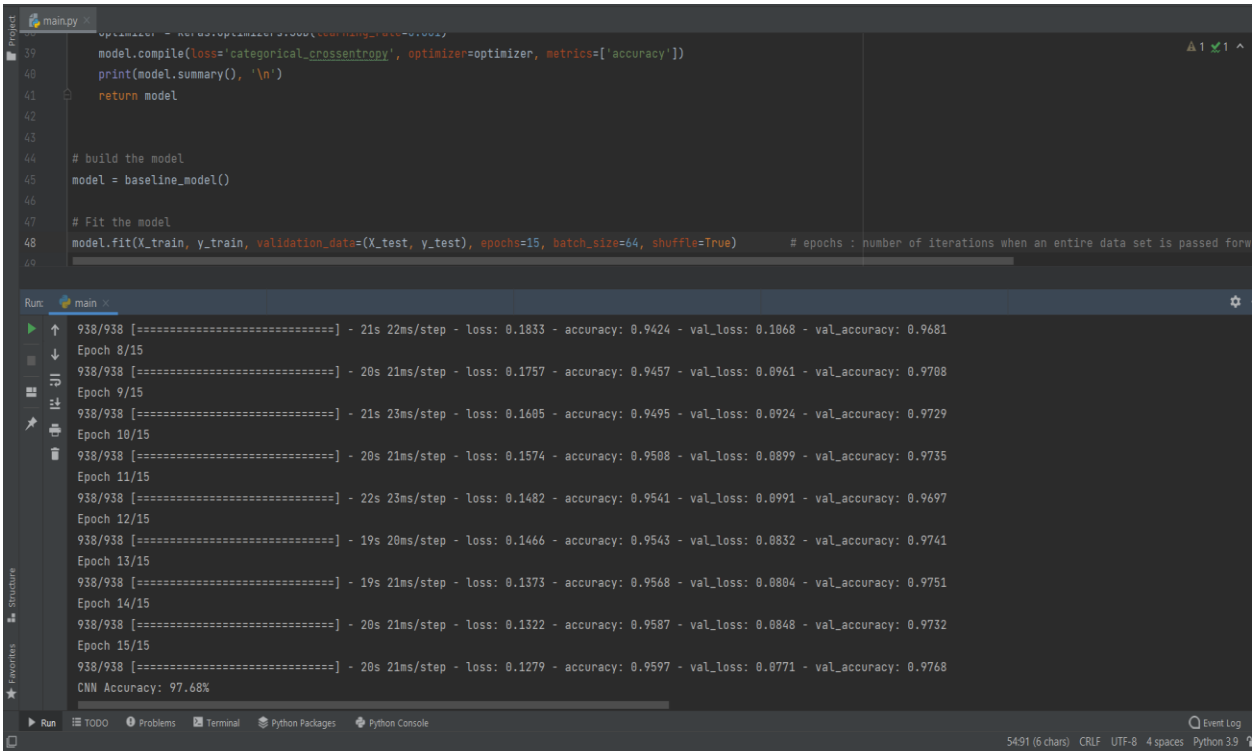
- Null

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 the number of Epochs from **10** to **13**

0-The code: we will change the number of Epochs from 10 to 15 with the same previous parameters (Learning Rate = 0.001, Batch size = 64, activation = relu, optimizer = SGD).

```
model.fit(X_train, y_train, validation_data=(X_test, y_test), epochs=15,
batch_size=64, shuffle=True)
```



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

- Accuracy: 97.68%

1	0.6459
2	0.8718
3	0.9057
4	0.9236
5	0.9327

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 1/15
938/938 [==============================] - 34s                    24ms/step
Epoch 2/15
938/938 [==============================] - 20s                    21ms/step
Epoch 3/15
938/938 [==============================] - 19s                    21ms/step
Epoch 4/15
938/938 [==============================] - 20s                    22ms/step
Epoch 5/15
938/938 [==============================] - 21s                    22ms/step
Epoch 6/15
938/938 [==============================] - 21s                    22ms/step
Epoch 7/15
938/938 [==============================] - 21s                    22ms/step
Epoch 8/15
938/938 [==============================] - 20s                    21ms/step
Epoch 9/15
938/938 [==============================] - 21s                    23ms/step
Epoch 10/15
938/938 [==============================] - 20s                    21ms/step
Epoch 11/15
938/938 [==============================] - 22s                    23ms/step
Epoch 12/15
938/938 [==============================] - 19s                    20ms/step
Epoch 13/15
938/938 [==============================] - 19s                    21ms/step
Epoch 14/15
938/938 [==============================] - 20s                    21ms/step
Epoch 15/15
938/938 [==============================] - 20s                    21ms/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:0.001

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):

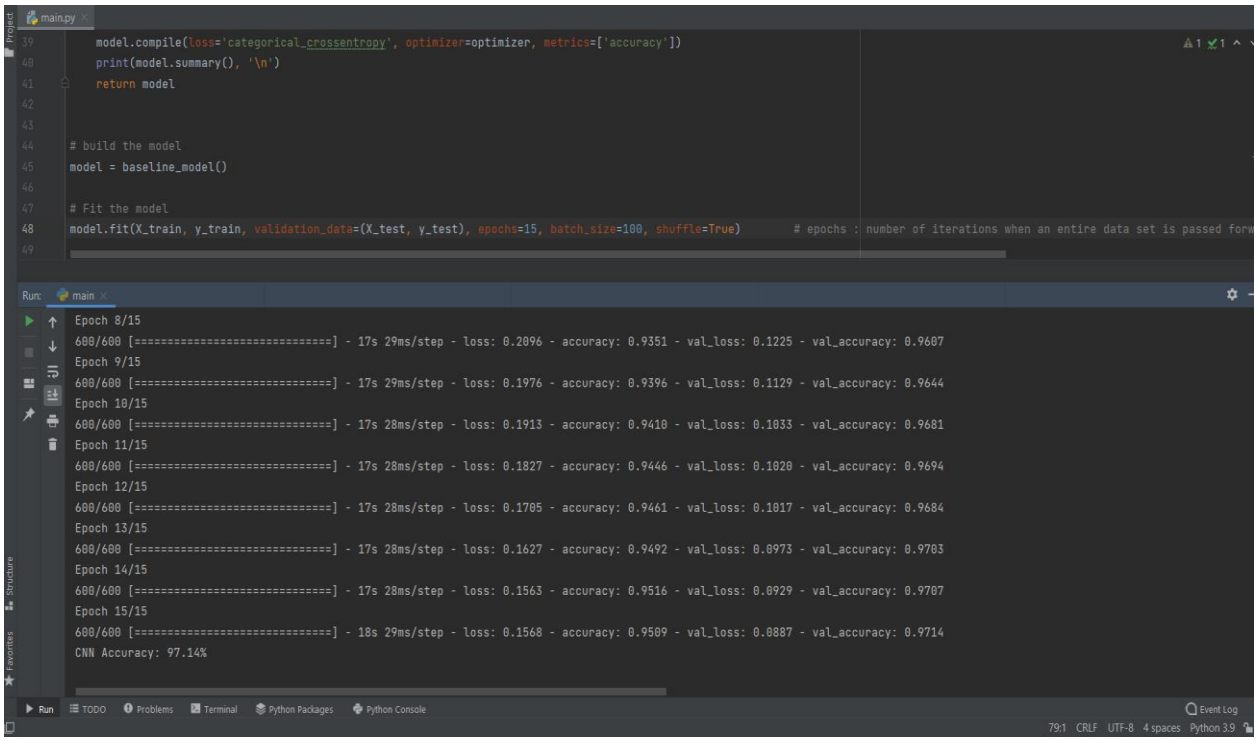
- Null

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 when we change the number of Epochs from **10** to **15**

0-The code: we will change the number of Batch Size from 64 to 100 with the same previous parameters (Learning Rate = 0.001, Epochs = 15, activation = relu, optimizer = SGD).

```
model.fit(X_train, y_train, validation_data=(X_test, y_test), epochs=15,
batch_size=100, shuffle=True)
```



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

- Accuracy: 97.14%

1	0.6117
2	0.8522
3	0.8883
4	0.9080
5	0.9157

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 1/15
600/600[==============================] - 29s                    30ms/step
Epoch 2/15
600/600 [==============================] - 18s                    31ms/step
Epoch 3/15
600/600[==============================] - 17s                    29ms/step
Epoch 4/15
600/600 [==============================] - 17s                    29ms/step
Epoch 5/15
600/600[==============================] - 17s                    28ms/step
Epoch 6/15
600/600 [==============================] - 17s                    29ms/step
Epoch 7/15
600/600[==============================] - 17s                    29ms/step
Epoch 8/15
600/600[==============================] - 17s                    29ms/step
Epoch 9/15
600/600[==============================] - 17s                    28ms/step
Epoch 10/15
600/600[==============================] - 17s                    28ms/step
Epoch 11/15
600/600 [==============================] - 17s                    28ms/step
Epoch 12/15
600/600[==============================] – 17s                    28ms/step
Epoch 13/15
600/600[==============================] - 17s                    28ms/step
Epoch 14/15
600/600 [==============================] - 17s                    28ms/step
Epoch 15/15
600/600[==============================] - 17s                    29ms/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:0.001

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):

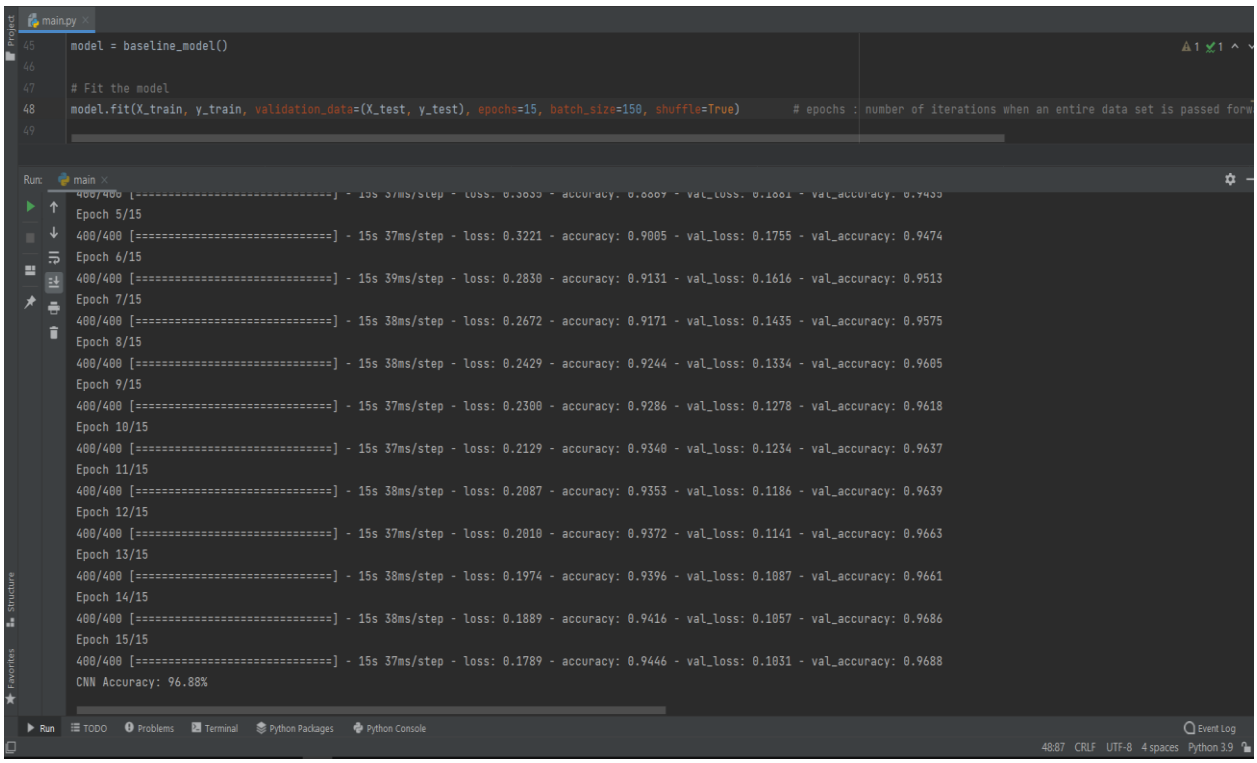
- Null

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 the number of Batch Size from **64** to **100**

0-The code: we will change the number of Batch Size from 64 to 150 with the same previous parameters (Learning Rate = 0.001, Epochs = 15, activation = relu, optimizer = SGD).

```
model.fit(X_train, y_train, validation_data=(X_test, y_test), epochs=15,
batch_size=150, shuffle=True)
```



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

- Accuracy: 96.88%

1	0.5938
2	0.8203
3	0.8633
4	0.8854
5	0.8966

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 1/15
400/400[==============================] - 29s                    43ms/step
Epoch 2/15
400/400[==============================] - 18s                    44ms/step
Epoch 3/15
400/400[==============================] - 17s                    43ms/step
Epoch 4/15
400/400[==============================] - 16s                    41ms/step
Epoch 5/15
400/400[==============================] - 17s                    44ms/step
Epoch 6/15
400/400 [==============================] - 18s                    44ms/step
Epoch 7/15
400/400[==============================] – 18s                    46ms/step
Epoch 8/15
400/400[==============================] - 17s                    42ms/step
Epoch 9/15
400/400[==============================] - 16s                    42ms/step
Epoch 10/15
400/400[==============================] - 16s                    41ms/step
Epoch 11/15
400/400 [==============================] - 16s                    40ms/step
Epoch 12/15
400/400[==============================] – 16s                    41ms/step
Epoch 13/15
400/400[==============================] - 16s                    41ms/step
Epoch 14/15
400/400 [==============================] - 16s                    41ms/step
Epoch 15/15
400/400[==============================] - 16s                    41ms/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:0.001

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):

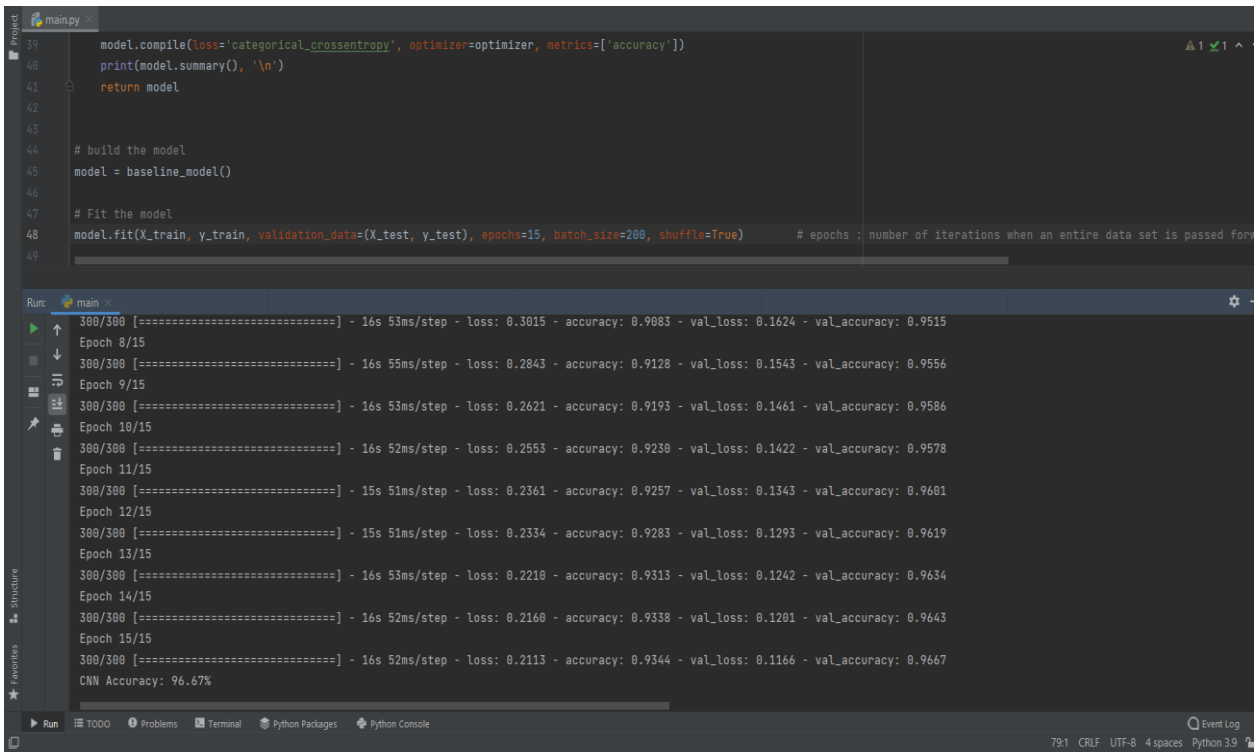
- Null

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 the number of Batch Size from **64** to **150**

0-The code: we will change the number of Batch Size from 64 to 200 with the same previous parameters (Learning Rate = 0.001, Epochs = 15, activation = relu, optimizer = SGD).

```
model.fit(X_train, y_train, validation_data=(X_test, y_test), epochs=15,
batch_size=200, shuffle=True)
```



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

- Accuracy: 96.67%

1	0.6061
2	0.8093
3	0.8548
4	0.8773
5	0.8895

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 1/15
300/300[==============================] - 30s                    59ms/step
Epoch 2/15
300/300[==============================] - 16s                    53ms/step
Epoch 3/15
300/300[==============================] - 17s                    56ms/step
Epoch 4/15
300/300[==============================] - 16s                    54ms/step
Epoch 5/15
300/300[==============================] - 15s                    49ms/step
Epoch 6/15
300/300 [==============================] - 16s                    52ms/step
Epoch 7/15
300/300[==============================] – 16s                    53ms/step
Epoch 8/15
300/300[==============================] - 16s                    55ms/step
Epoch 9/15
300/300[==============================] - 16s                    53ms/step
Epoch 10/15
300/300[==============================] - 16s                    52ms/step
Epoch 11/15
300/300 [==============================] - 15s                    51ms/step
Epoch 12/15
300/300[==============================] – 15s                    51ms/step
Epoch 13/15
300/300[==============================] - 16s                    53ms/step
Epoch 14/15
300/300 [==============================] - 16s                    52ms/step
Epoch 15/15
300/300[==============================] - 16s                    52ms/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:0.001

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

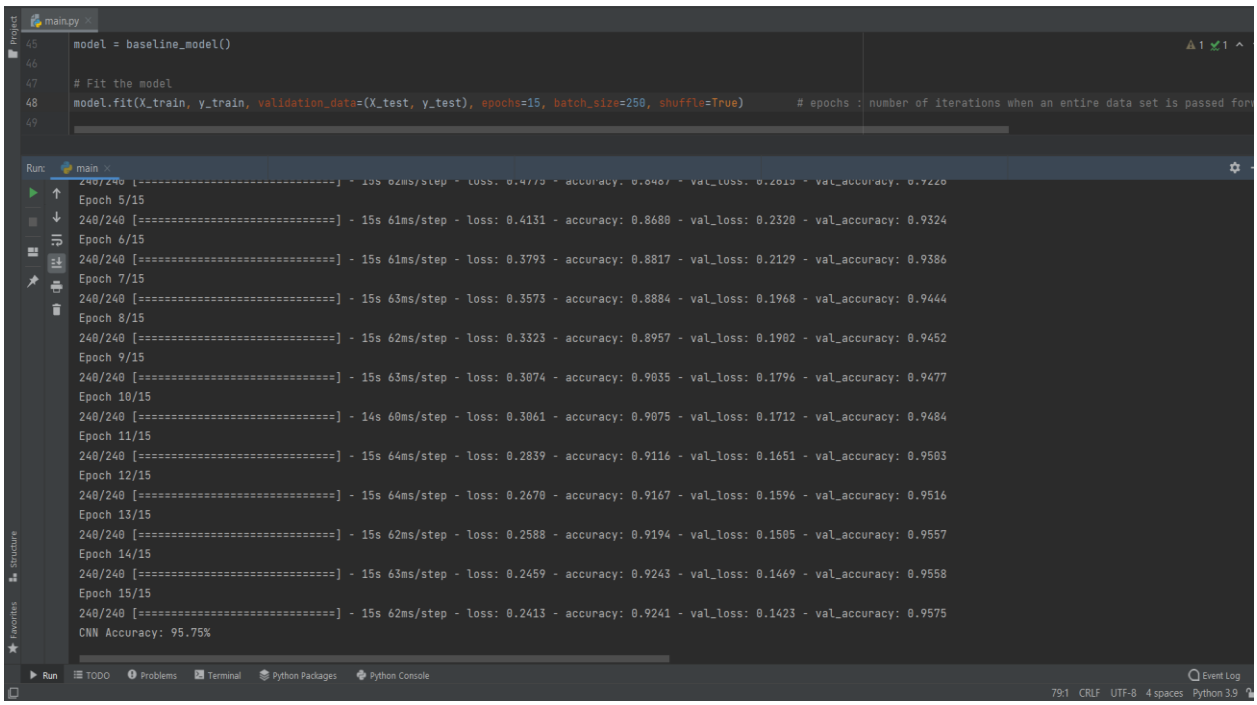
10-Write what you observed has changed due to the parameter you changed (according to the above part):

- As you see that the CNN Error increased when we change the number of Batch Size from **64** to **200**

0-The code: we will change the number of Batch Size from 64 to 250 with the same previous parameters (Learning Rate = 0.001, Epochs = 15, activation = relu, optimizer = SGD).

```
model.fit(X_train, y_train, validation_data=(X_test, y_test), epochs=15,
batch_size=250, shuffle=True)
```

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



- Accuracy: 95.75%

1	0.4920
2	0.7563
3	0.8193
4	0.8487
5	0.8680

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 1/15
240/240[==============================] - 26s                    64ms/step
Epoch 2/15
240/240 [==============================] - 15s                    62ms/step
Epoch 3/15
240/240 [==============================] - 15s                    62ms/step
Epoch 4/15
240/240 [==============================] - 15s                    62ms/step
Epoch 5/15
240/240 [==============================] - 15s                    61ms/step
Epoch 6/15
240/240 [==============================] - 15s                    61ms/step
Epoch 7/15
240/240 [==============================] – 15s                    63ms/step
Epoch 8/15
240/240 [==============================] - 15s                    62ms/step
Epoch 9/15
240/240 [==============================] - 15s                    63ms/step
Epoch 10/15
240/240 [==============================] - 14s                    60ms/step
Epoch 11/15
240/240 [==============================] - 15s                    64ms/step
Epoch 12/15
240/240 [==============================] – 15s                    64ms/step
Epoch 13/15
240/240 [==============================] - 15s                    62ms/step
Epoch 14/15
240/240 [==============================] - 15s                    63ms/step
Epoch 15/15
240/240 [==============================] - 15s                    62ms/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:0.001

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 the number of Batch Size from **64** to **250**

So, the best model we made is:

```
# 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.np_utils import to_categorical


# 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 = to_categorical(y_train)  # it makes label 5 = [ 0, 0, 0, 0, 0, 1,
0, 0, 0, 0]
y_test = 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(Dropout(0.5))
    model.add(Dense(128, activation='relu'))
    model.add(Dense(num_classes, activation='softmax'))
    # Compile model
    optimizer = keras.optimizers.SGD(learning_rate=0.001)
    model.compile(loss='categorical_crossentropy', optimizer=optimizer,
metrics=['accuracy'])
    print(model.summary(), '\n')
    return model


# build the model
model = baseline_model()
```

```
# Fit the model
model.fit(X_train, y_train, validation_data=(X_test, y_test), epochs=15,
batch_size=64, shuffle=True)        # epochs : number of iterations when an
entire data set is passed forward and backward through the neural network

# Final evaluation of the model
scores = model.evaluate(X_test, y_test, verbose=0)
print("CNN Accuracy: %.2f%%" % (scores[1] * 100), '\n')
```