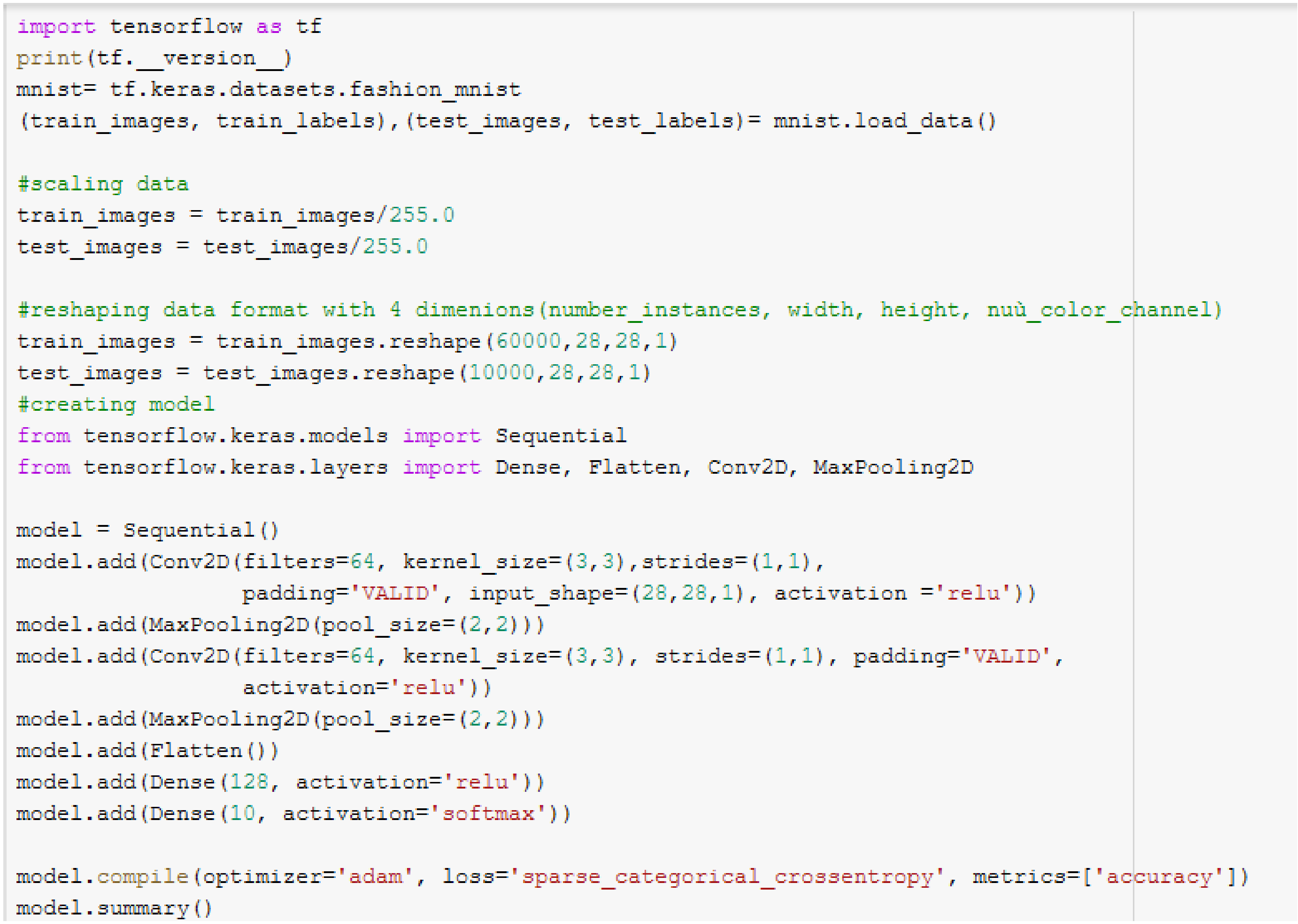
# Convolutional Neural Network (CNN) for MNIST_Fashion Dataset (tensorflow.keras.datasets):

## Visuals & Parameters explanation by Abir ELTAIEF :

## 0- Code :



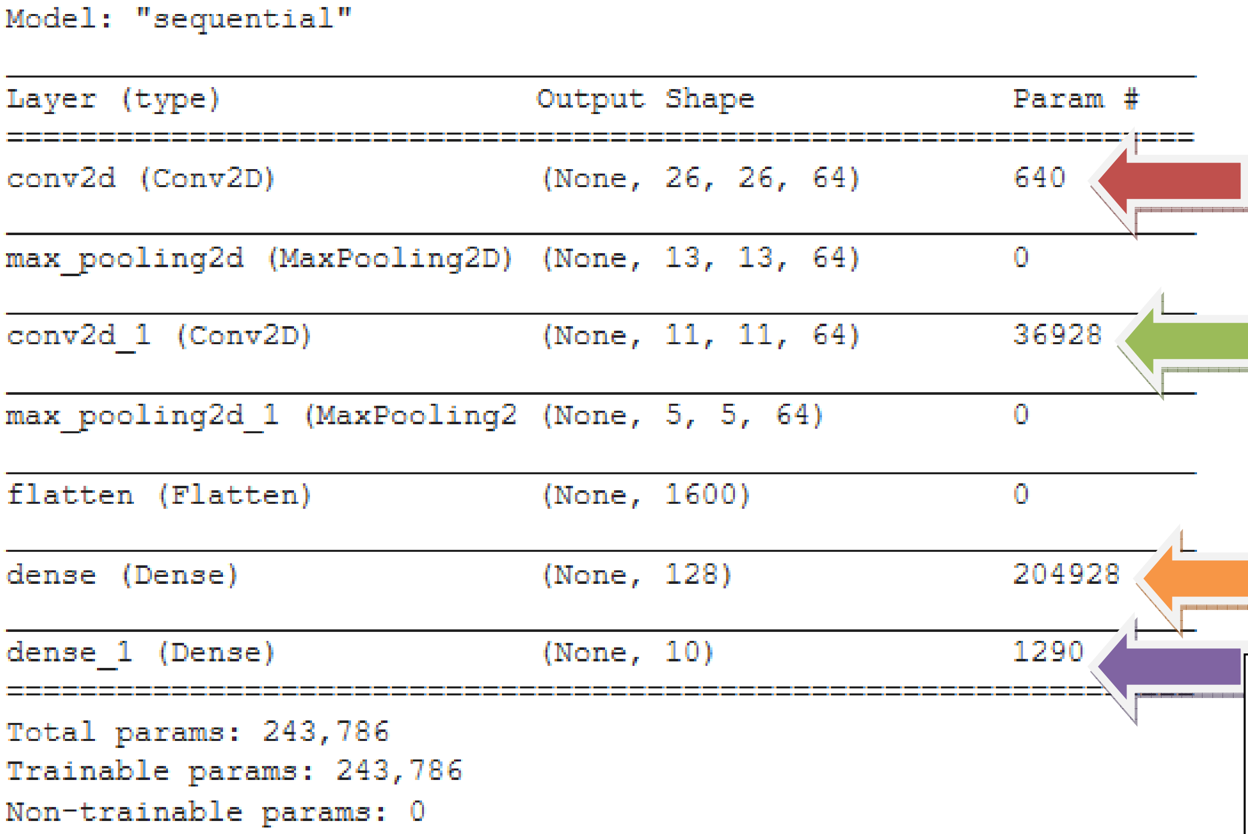
```
import tensorflow as tf
print(tf.__version__)
mnist= tf.keras.datasets.fashion_mnist
(train_images, train_labels),(test_images, test_labels)= mnist.load_data()

#scaling data
train_images = train_images/255.0
test_images = test_images/255.0

#reshaping data format with 4 dimenions(number_instances, width, height, nuù_color_channel)
train_images = train_images.reshape(60000,28,28,1)
test_images = test_images.reshape(10000,28,28,1)
#creating model
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, Flatten, Conv2D, MaxPooling2D

model = Sequential()
model.add(Conv2D(filters=64, kernel_size=(3,3),strides=(1,1),
                 padding='VALID', input_shape=(28,28,1), activation ='relu'))
model.add(MaxPooling2D(pool_size=(2,2)))
model.add(Conv2D(filters=64, kernel_size=(3,3), strides=(1,1), padding='VALID',
                 activation='relu'))
model.add(MaxPooling2D(pool_size=(2,2)))
model.add(Flatten())
model.add(Dense(128, activation='relu'))
model.add(Dense(10, activation='softmax'))

model.compile(optimizer='adam', loss='sparse_categorical_crossentropy', metrics=['accuracy'])
model.summary()
```



```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #   
=================================================================
conv2d (Conv2D)              (None, 26, 26, 64)        640       
_________________________________________________________________
max_pooling2d (MaxPooling2D) (None, 13, 13, 64)        0         
_________________________________________________________________
conv2d_1 (Conv2D)            (None, 11, 11, 64)        36928     
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 5, 5, 64)          0         
_________________________________________________________________
flatten (Flatten)            (None, 1600)              0         
_________________________________________________________________
dense (Dense)                (None, 128)               204928    
_________________________________________________________________
dense_1 (Dense)              (None, 10)                1290      
=================================================================
Total params: 243,786
Trainable params: 243,786
Non-trainable params: 0
```

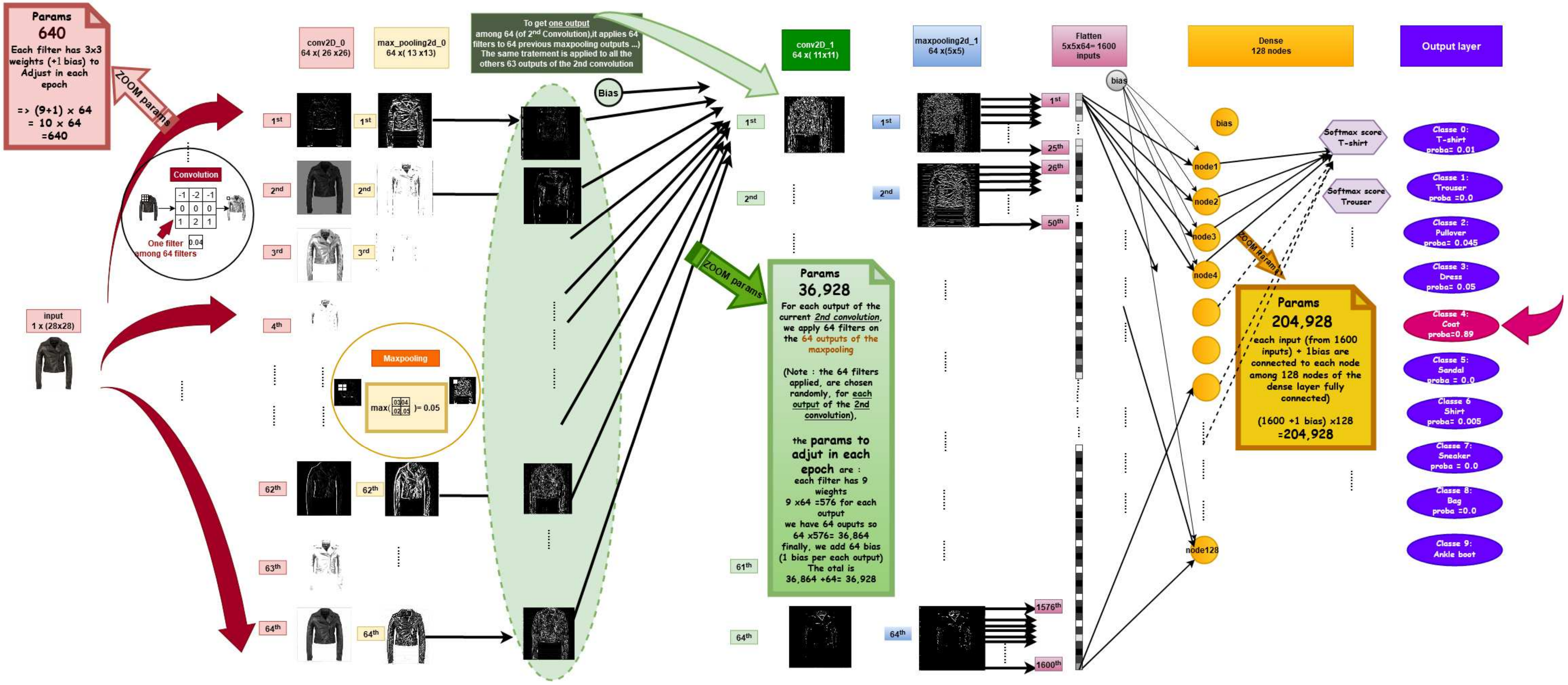
See params explanation (View 1)

See params explanation (View 2)

See params explanation (View 3)

See params explanation (View 4)+ comment below the visual

# 1- *General Visual : view 0 (See enlarged views below)*



Params
640
Each filter has 3x3 weights (+1 bias) to Adjust in each epoch

=> (9+1) x 64
= 10 x 64
=640

To get one output among 64 (of 2nd Convolution),it applies 64 filters to 64 previous maxpooling outputs ...) The same tratement is applied to all the others 63 outputs of the 2nd convolution

Params
36,928
For each output of the current *2nd convolution*, we apply 64 filters on the 64 outputs of the maxpooling

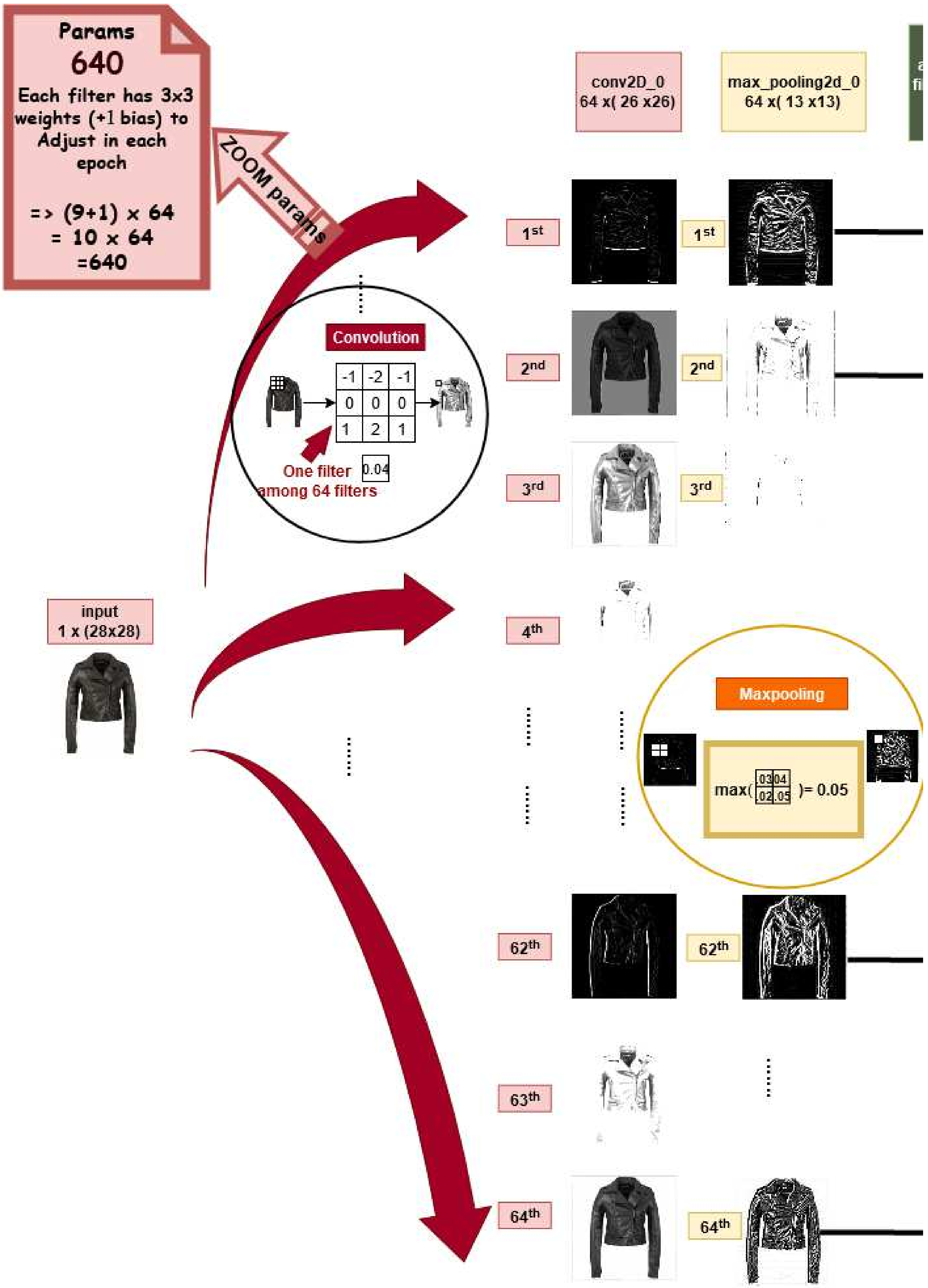
(Note : the 64 filters applied, are chosen randomly, for each output of the 2nd convolution),

the **params to adjut in each epoch** are :
each filter has 9 wieghts
9 x64 =576 for each output
we have 64 ouputs so
64 x576= 36,864
finally, we add 64 bias (1 bias per each output)
The otal is
36,864 +64= 36,928

Params
204,928
each input (from 1600 inputs) + 1bias are connected to each node among 128 nodes of the dense layer fully connected)

(1600 +1 bias) x128
=204,928

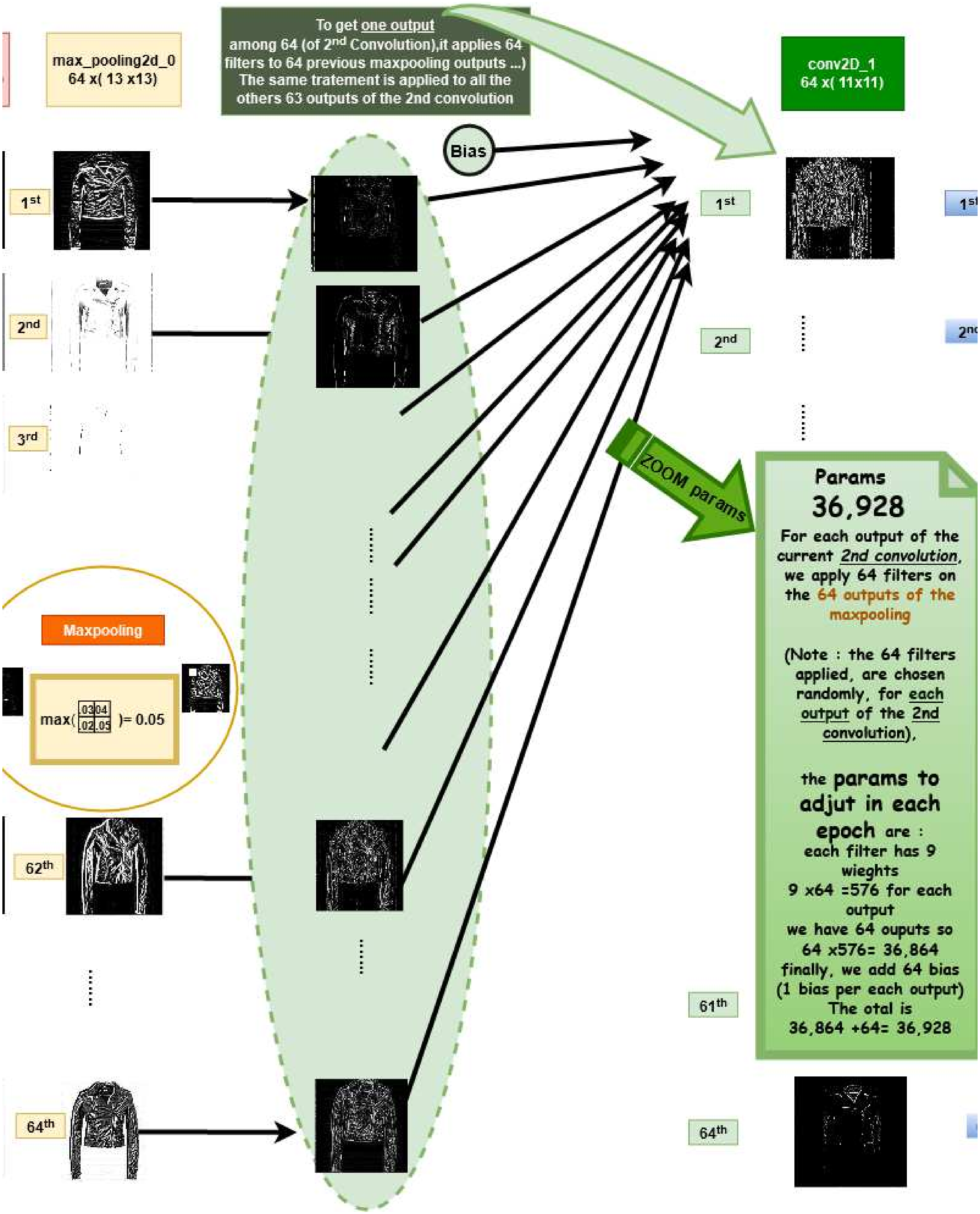
## 2- Detailed visual (view1 :1st Convolutional layer+ 1st maxpooling layer)



Params
640
Each filter has 3x3 weights (+1 bias) to Adjust in each epoch

=> (9+1) x 64
= 10 x 64
=640

## 3- Detailed visual (view2 :2nd Convolutional layer)



To get one output among 64 (of 2nd Convolution),it applies 64 filters to 64 previous maxpooling outputs ...) The same tratement is applied to all the others 63 outputs of the 2nd convolution

**Params**

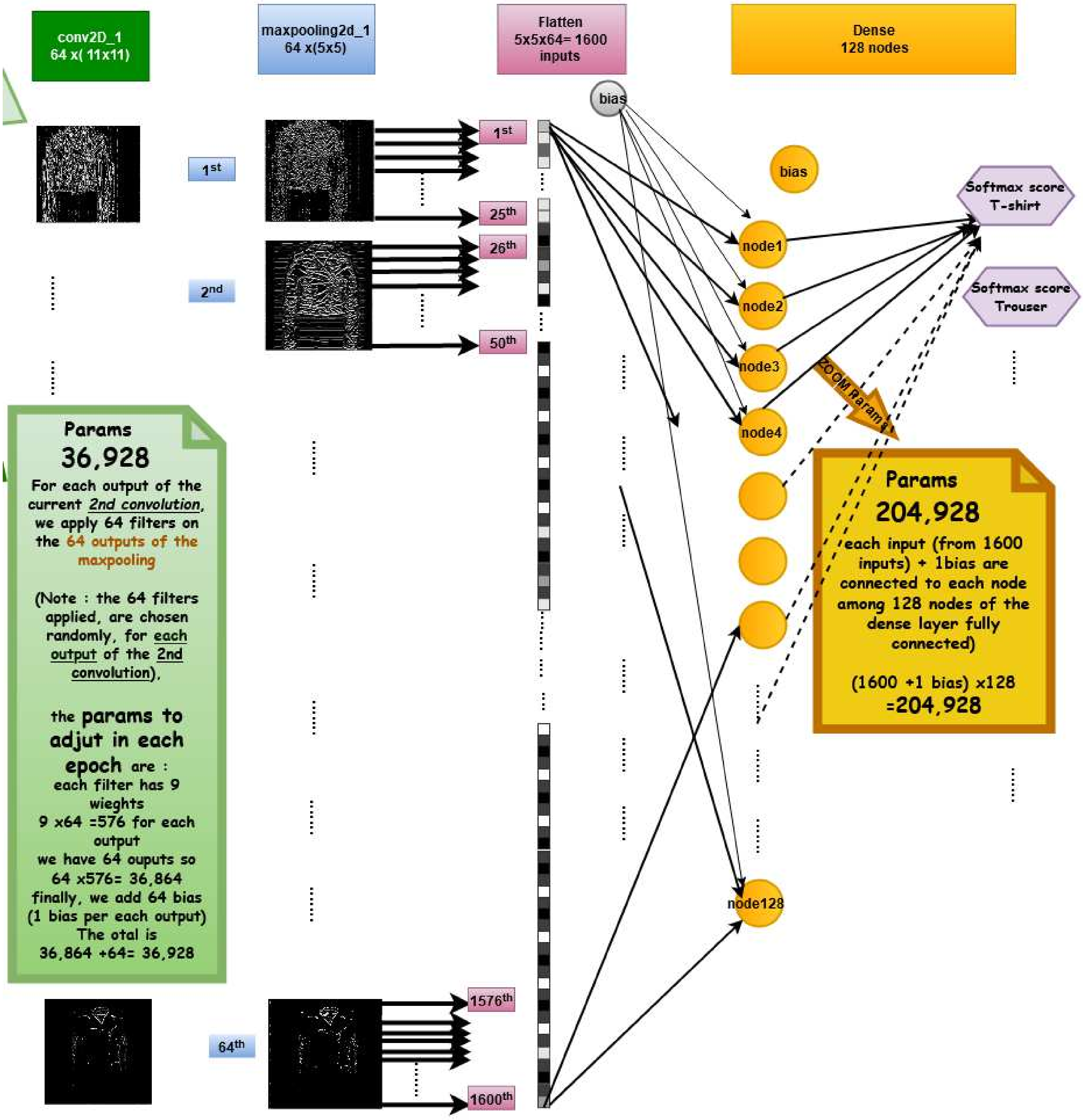
**36,928**

For each output of the current *2nd convolution*, we apply 64 filters on the 64 outputs of the maxpooling

(Note : the 64 filters applied, are chosen randomly, for each output of the 2nd convolution),

**the params to adjut in each epoch** are :
each filter has 9 wieghts
9 x64 =576 for each output
we have 64 ouputs so
64 x576= 36,864
finally, we add 64 bias
(1 bias per each output)
The otal is
36,864 +64= 36,928

## 4- Detailed visual (view3 :2nd maxpooling layer + Flatten layer+ Dense layer fully connected)



Params

36,928

For each output of the current *2nd convolution*, we apply 64 filters on the 64 outputs of the maxpooling

(Note : the 64 filters applied, are chosen randomly, for each output of the 2nd convolution),

the params to adjut in each epoch are :
each filter has 9 wieghts
9 x64 =576 for each output
we have 64 ouputs so
64 x576= 36,864
finally, we add 64 bias (1 bias per each output)
The otal is
36,864 +64= 36,928

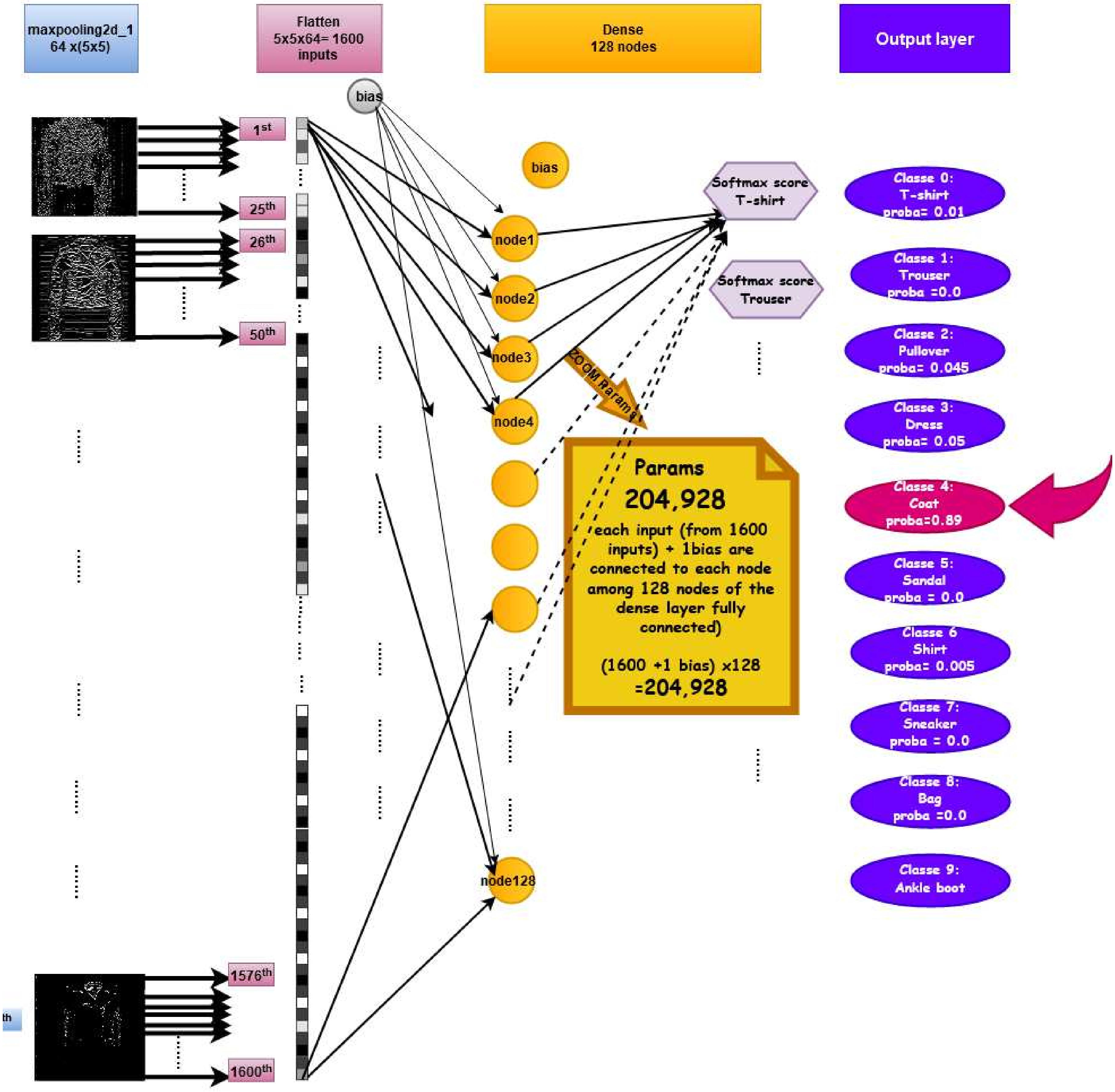
Params

204,928

each input (from 1600 inputs) + 1bias are connected to each node among 128 nodes of the dense layer fully connected)

(1600 +1 bias) x128
=204,928

## 5- Detailed visual (view4 :Output layer)



***The final 1290 params ? :***

= (128 weights + 1 bias) x 10 (classes)= 129 x10= **1290**.

**By Abir ELTAIEF**

**Passion for Maths, Algorithms & Data Science…**