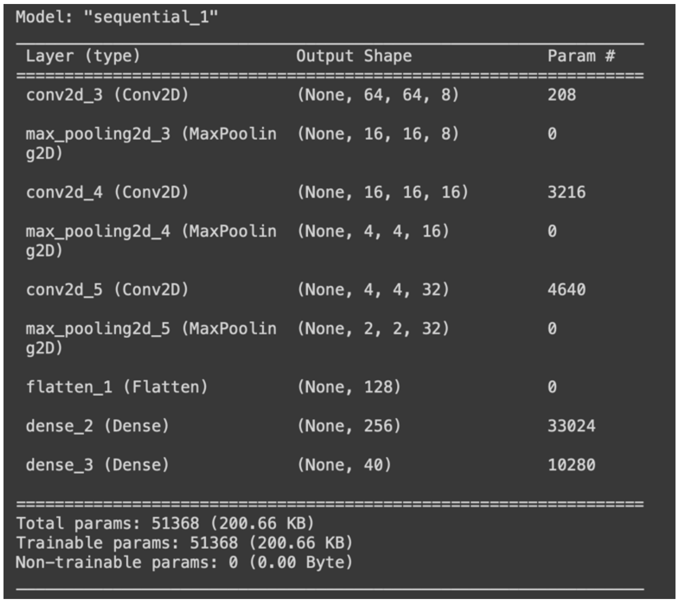
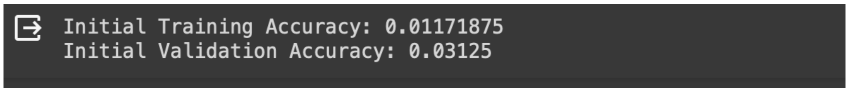
Artificial neural network project
Student name: Khalid Nimri
Student ID: 2140145 (TL1)

Question 1: What is the output of the model.summary()?



```
Model: "sequential_1"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d_3 (Conv2D)           (None, 64, 64, 8)         208

 max_pooling2d_3 (MaxPoolin  (None, 16, 16, 8)         0
 g2D)

 conv2d_4 (Conv2D)           (None, 16, 16, 16)        3216

 max_pooling2d_4 (MaxPoolin  (None, 4, 4, 16)          0
 g2D)

 conv2d_5 (Conv2D)           (None, 4, 4, 32)          4640

 max_pooling2d_5 (MaxPoolin  (None, 2, 2, 32)          0
 g2D)

 flatten_1 (Flatten)         (None, 128)               0

 dense_2 (Dense)             (None, 256)               33024

 dense_3 (Dense)             (None, 40)                10280

=================================================================
Total params: 51368 (200.66 KB)
Trainable params: 51368 (200.66 KB)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
```

Question 2: What the initial training accuracy and validation accuracy of the CNN?



```
Initial Training Accuracy: 0.01171875
Initial Validation Accuracy: 0.03125
```

Question 3: How many convolutional layers and pooling layers does this network have?

3 convolutional layers which are:

```
conv2d_3 (Conv2D)           (None, 64, 64, 8)         208
conv2d_4 (Conv2D)           (None, 16, 16, 16)        3216
conv2d_5 (Conv2D)           (None, 4, 4, 32)          4640
```

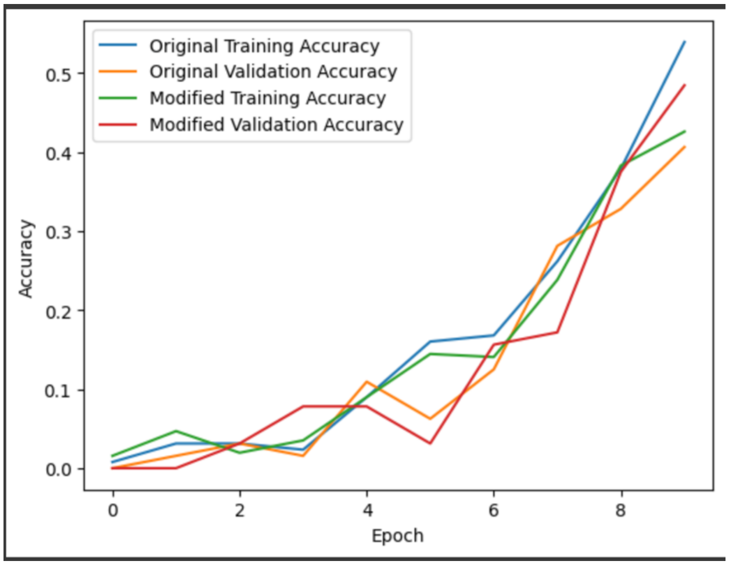
3 pooling layers which are:

```
max_pooling2d_3 (MaxPoolin  (None, 16, 16, 8)         0
 g2D)
max_pooling2d_4 (MaxPoolin  (None, 4, 4, 16)          0
 g2D)
max_pooling2d_5 (MaxPoolin  (None, 2, 2, 32)          0
 g2D)
```

Question 4: Generally, the larger the size of the image the more the information in it.
The max-pooling layers after first and second Convolutional layer decrease the size of the image by 4.
Check if this is causing the network to have such a poor validation accuracy?
If the size of pooling layers size is changed from (4,4) to (2,2) what is the effect on accuracy of the network?

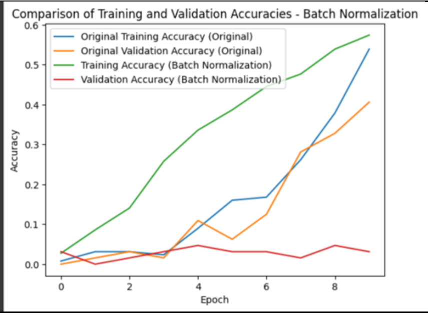
We can clearly see that when we reduced the size of the pooling layers the accuracy is better in the validation accuracy after the 6th epoch.
the original training accuracy is better than the modified although the original validation accuracy is lower than the modified validation accuracy.

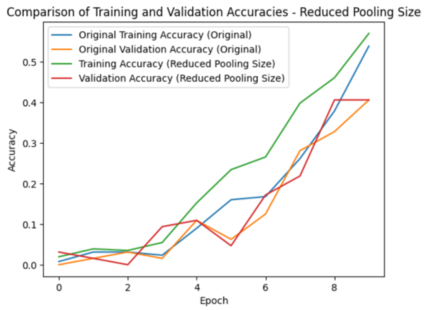
-	Size (4,4) original	Size (2,2) modified
Training accuracy	better	worse
validation accuracy	worse	better



Question 5: Dr. Hinton, has highlighted that aggressively using pooling layers may result in loss of important information.
Is there a way that the CNN architecture starts producing better training and validation accuracy?

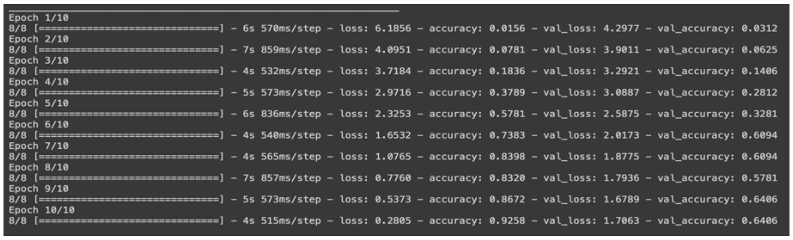
Yes, and these are some methods with their outputs:

Method	Output accuracy
Batch Normalization	

Reducing pooling size **Increasing the number of epochs will increase the accuracy by a lot!** **In this plot the number of epochs is the same which is 10**	

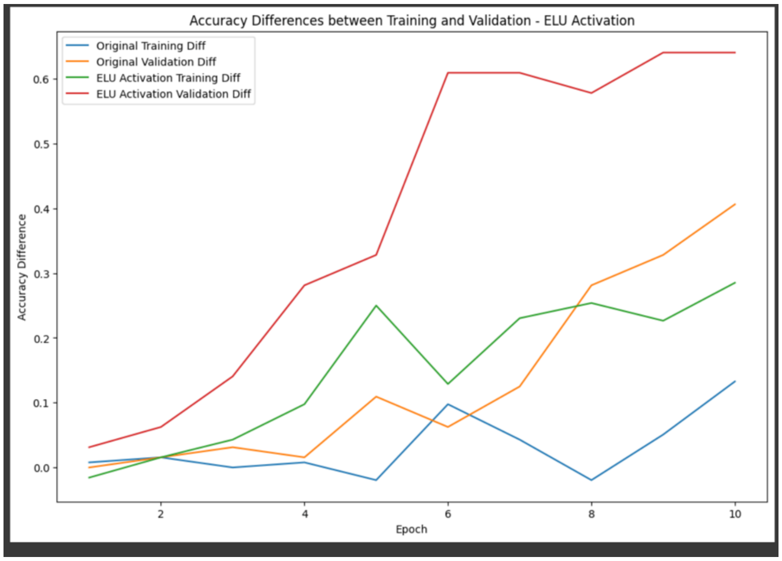
Question 6: Make changes to the convolutional neural network to get the best validation accuracy.
You are not allowed to change the number of epochs or batch size for this task.

I tried increasing the complexity of the model without changing the batch size or the number of epochs, also I used different activation function which is the Elu ,the training accuracy as well as the validation accuracy increased
And these are the results I have got.



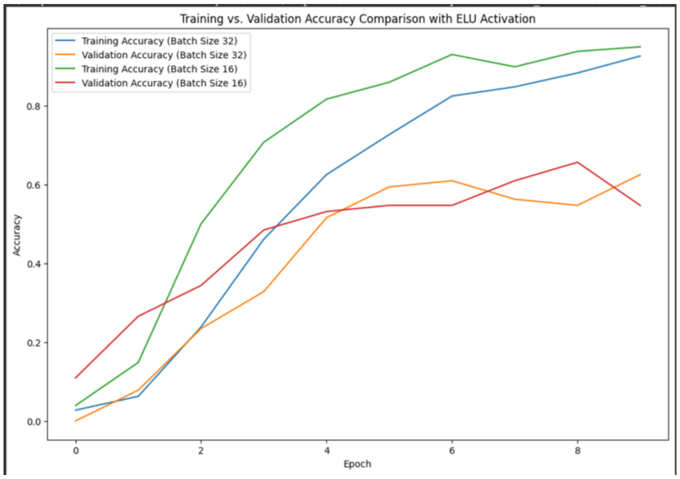
```
Epoch 1/10
8/8 [==============================] - 6s 570ms/step - loss: 6.1856 - accuracy: 0.0156 - val_loss: 4.2977 - val_accuracy: 0.0312
Epoch 2/10
8/8 [==============================] - 7s 859ms/step - loss: 4.0951 - accuracy: 0.0781 - val_loss: 3.9011 - val_accuracy: 0.0625
Epoch 3/10
8/8 [==============================] - 4s 532ms/step - loss: 3.7184 - accuracy: 0.1836 - val_loss: 3.2921 - val_accuracy: 0.1406
Epoch 4/10
8/8 [==============================] - 5s 573ms/step - loss: 2.9716 - accuracy: 0.3789 - val_loss: 3.0887 - val_accuracy: 0.2812
Epoch 5/10
8/8 [==============================] - 6s 836ms/step - loss: 2.3253 - accuracy: 0.5781 - val_loss: 2.5875 - val_accuracy: 0.3281
Epoch 6/10
8/8 [==============================] - 4s 540ms/step - loss: 1.6532 - accuracy: 0.7383 - val_loss: 2.0173 - val_accuracy: 0.6094
Epoch 7/10
8/8 [==============================] - 4s 565ms/step - loss: 1.0765 - accuracy: 0.8398 - val_loss: 1.8775 - val_accuracy: 0.6094
Epoch 8/10
8/8 [==============================] - 7s 857ms/step - loss: 0.7760 - accuracy: 0.8320 - val_loss: 1.7936 - val_accuracy: 0.5781
Epoch 9/10
8/8 [==============================] - 5s 573ms/step - loss: 0.5373 - accuracy: 0.8672 - val_loss: 1.6789 - val_accuracy: 0.6406
Epoch 10/10
8/8 [==============================] - 4s 515ms/step - loss: 0.2805 - accuracy: 0.9258 - val_loss: 1.7063 - val_accuracy: 0.6406
```

Question 7: Plot the difference between training and validation accuracy for each epoch.



Question 8: For the best network architecture change the batch size to 16 and plot the training vs validation accuracy graph. What happened to the validation accuracy after last epoch as compared to when the batch size was 32.

The accuracy increased after changing the batch size to 32

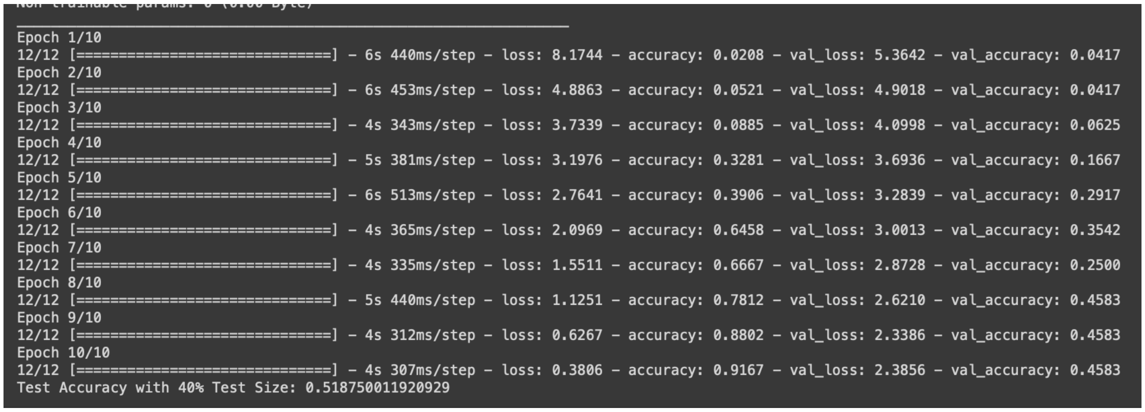


Question 9: For the best network architecture change the number of epochs to 5 and 20 and share the final validation accuracy for 5, 10 and 20 epochs. What do the results highlight?

Batch size	Epoch number	Training accuracy	Validation accuracy
16	5	69.14%	48.43%
16	10	92.97%	57.81%
16	20	100%	65.62%

Question 10: For the best network architecture and batch size =16 and epochs =10, change the test data size to 40% and share what is the effect on validation accuracy of the algorithm?

The validation accuracy decreased to 51.87% after changing the test size to 40%



```
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
Epoch 1/10
12/12 [==============================] - 6s 440ms/step - loss: 8.1744 - accuracy: 0.0208 - val_loss: 5.3642 - val_accuracy: 0.0417
Epoch 2/10
12/12 [==============================] - 6s 453ms/step - loss: 4.8863 - accuracy: 0.0521 - val_loss: 4.9018 - val_accuracy: 0.0417
Epoch 3/10
12/12 [==============================] - 4s 343ms/step - loss: 3.7339 - accuracy: 0.0885 - val_loss: 4.0998 - val_accuracy: 0.0625
Epoch 4/10
12/12 [==============================] - 5s 381ms/step - loss: 3.1976 - accuracy: 0.3281 - val_loss: 3.6936 - val_accuracy: 0.1667
Epoch 5/10
12/12 [==============================] - 6s 513ms/step - loss: 2.7641 - accuracy: 0.3906 - val_loss: 3.2839 - val_accuracy: 0.2917
Epoch 6/10
12/12 [==============================] - 4s 365ms/step - loss: 2.0969 - accuracy: 0.6458 - val_loss: 3.0013 - val_accuracy: 0.3542
Epoch 7/10
12/12 [==============================] - 4s 335ms/step - loss: 1.5511 - accuracy: 0.6667 - val_loss: 2.8728 - val_accuracy: 0.2500
Epoch 8/10
12/12 [==============================] - 5s 440ms/step - loss: 1.1251 - accuracy: 0.7812 - val_loss: 2.6210 - val_accuracy: 0.4583
Epoch 9/10
12/12 [==============================] - 4s 312ms/step - loss: 0.6267 - accuracy: 0.8802 - val_loss: 2.3386 - val_accuracy: 0.4583
Epoch 10/10
12/12 [==============================] - 4s 307ms/step - loss: 0.3806 - accuracy: 0.9167 - val_loss: 2.3856 - val_accuracy: 0.4583
Test Accuracy with 40% Test Size: 0.518750011920929
```