DSC650-T301 Big Data (2235-1)

4/18/2023

Joshua Greenert

6.1

In [23]:
```
from keras import layers
from keras import models

model = models.Sequential()
model.add(layers.Conv2D(32, (3, 3), activation = 'relu', input_shape = (28, 28, 1)))
model.add(layers.MaxPooling2D((2, 2)))
model.add(layers.Conv2D(64, (3,3), activation = 'relu'))
model.add(layers.MaxPooling2D((2, 2)))
model.add(layers.Conv2D(64, (3,3), activation = 'relu'))
```

In [24]:
```
# Display the architecture of the convnet so far.
model.summary()
```

```
Model: "sequential_3"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d_9 (Conv2D)           (None, 26, 26, 32)        320

 max_pooling2d_6 (MaxPooling  (None, 13, 13, 32)       0
 2D)

 conv2d_10 (Conv2D)          (None, 11, 11, 64)        18496

 max_pooling2d_7 (MaxPooling  (None, 5, 5, 64)         0
 2D)

 conv2d_11 (Conv2D)          (None, 3, 3, 64)          36928

=================================================================
Total params: 55,744
Trainable params: 55,744
Non-trainable params: 0
_________________________________________________________________
```

In [25]:
```
# Adding a classifier on top of the convnet
model.add(layers.Flatten())
model.add(layers.Dense(64, activation = 'relu'))
model.add(layers.Dense(10, activation = 'softmax'))
```

In [26]:
```
# View the summary
model.summary()
```

```
Model: "sequential_3"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d_9 (Conv2D)           (None, 26, 26, 32)        320

 max_pooling2d_6 (MaxPooling  (None, 13, 13, 32)       0
```

```
 2D)

 conv2d_10 (Conv2D)          (None, 11, 11, 64)        18496

 max_pooling2d_7 (MaxPooling  (None, 5, 5, 64)         0
 2D)

 conv2d_11 (Conv2D)          (None, 3, 3, 64)          36928

 flatten_3 (Flatten)         (None, 576)               0

 dense_6 (Dense)             (None, 64)                36928

 dense_7 (Dense)             (None, 10)                650

=================================================================
Total params: 93,322
Trainable params: 93,322
Non-trainable params: 0
_________________________________________________________________
```

In [27]:

```python
# Training the convnet on MNIST images
from keras.datasets import mnist
from keras.utils import to_categorical
from keras import optimizers

# set an SGD optimizer with a static learning rate.
# opt = optimizers.SGD(learning_rate=0.01)

(train_images, train_labels), (test_images, test_labels) = mnist.load_data()

train_images = train_images.reshape((60000, 28, 28, 1))
train_images = train_images.astype('float32') / 255

test_images = test_images.reshape((10000, 28, 28, 1))
test_images = test_images.astype('float32') / 255

train_labels = to_categorical(train_labels)
test_labels = to_categorical(test_labels)

model.compile(optimizer = 'rmsprop', loss = 'categorical_crossentropy', metrics = ['acc'
model.fit(train_images, train_labels, epochs=5, batch_size=64)
```

```
Epoch 1/5
938/938 [==============================] - 18s 19ms/step - loss: 0.1644 - acc: 0.9482
Epoch 2/5
938/938 [==============================] - 18s 19ms/step - loss: 0.0461 - acc: 0.9856
Epoch 3/5
938/938 [==============================] - 18s 19ms/step - loss: 0.0320 - acc: 0.9901
Epoch 4/5
938/938 [==============================] - 18s 19ms/step - loss: 0.0247 - acc: 0.9926
Epoch 5/5
938/938 [==============================] - 18s 19ms/step - loss: 0.0195 - acc: 0.9941
```

Out[27]:

```
<keras.callbacks.History at 0x251a2974d90>
```

In [28]:

```python
# Evaluate the model
test_loss, test_acc = model.evaluate(test_images, test_labels)
test_acc
```

```
313/313 [==============================] - 1s 4ms/step - loss: 0.0275 - acc: 0.9914
```

Out[28]:

```
0.9914000034332275
```

In [29]:

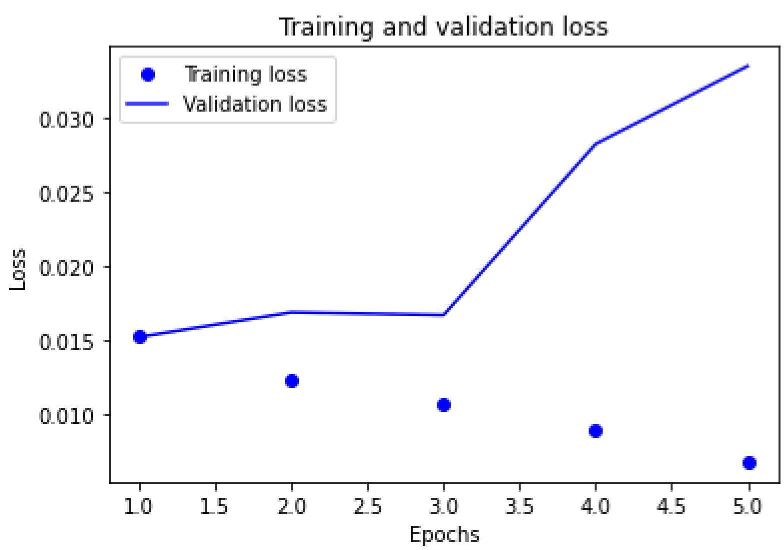
```python
from sklearn.model_selection import train_test_split
```

```
# Create a train-test split.
train_images, val_images, train_labels, val_labels = train_test_split(train_images, trai

# Create a history object to use for plotting.
history = model.fit(train_images, train_labels, epochs=5, batch_size=64, validation_data
```

```
Epoch 1/5
750/750 [==============================] - 16s 21ms/step - loss: 0.0153 - acc: 0.9953 -
val_loss: 0.0152 - val_acc: 0.9952
Epoch 2/5
750/750 [==============================] - 16s 21ms/step - loss: 0.0122 - acc: 0.9963 -
val_loss: 0.0169 - val_acc: 0.9947
Epoch 3/5
750/750 [==============================] - 16s 21ms/step - loss: 0.0106 - acc: 0.9968 -
val_loss: 0.0167 - val_acc: 0.9946
Epoch 4/5
750/750 [==============================] - 16s 21ms/step - loss: 0.0088 - acc: 0.9974 -
val_loss: 0.0283 - val_acc: 0.9933
Epoch 5/5
750/750 [==============================] - 16s 21ms/step - loss: 0.0067 - acc: 0.9981 -
val_loss: 0.0335 - val_acc: 0.9927
```

In [30]:

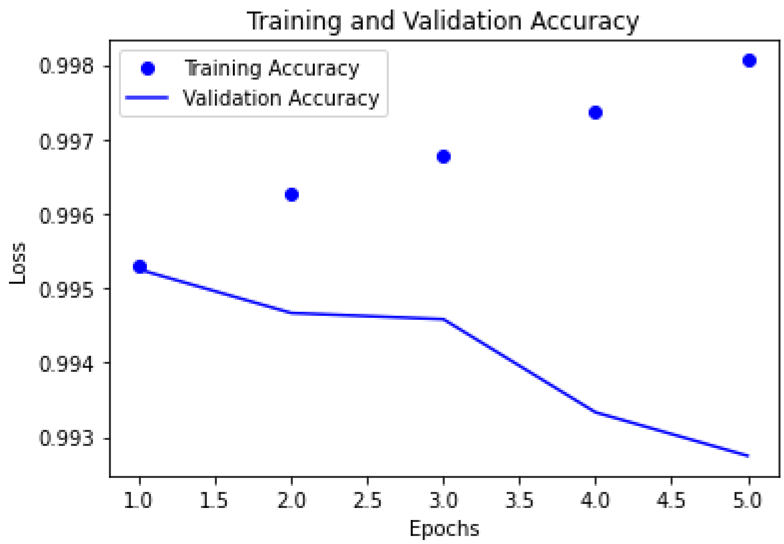
```
# Create a plot to view the training and validation loss
import matplotlib.pyplot as plt

history_dict = history.history
loss_values = history_dict['loss']
val_loss_values = history_dict['val_loss']

epochs = range(1, len(loss_values) + 1)

plt.plot(epochs, loss_values, 'bo', label='Training loss')
plt.plot(epochs, val_loss_values, 'b', label='Validation loss')
plt.title('Training and validation loss')
plt.xlabel('Epochs')
plt.ylabel('Loss')
plt.legend()

plt.show()
```



In [33]:

```
# Plot the train data and validation accuracy.
plt.clf()
acc_values = history_dict['acc']
val_acc_values = history_dict['val_acc']

plt.plot(epochs, acc_values, 'bo', label= 'Training Accuracy')
plt.plot(epochs, val_acc_values, 'b', label = "Validation Accuracy")
```

```
plt.title('Training and Validation Accuracy')
plt.xlabel('Epochs')
plt.ylabel('Loss')
plt.legend()

plt.show()
```



In [35]:

```
# Save the results
import os
import numpy as np

# Use the model to make predictions
predictions = model.predict(test_images)
predicted_labels = np.argmax(predictions, axis=1)

# Save the model to the results folder
model.save('results/model-6-1.h5')
plt.savefig('results/validation-6-1.png')
np.savetxt('results/predictions-6-1.txt', predicted_labels, fmt='%d')
np.savetxt('results/metrics-6-1.txt', np.column_stack((loss_values, acc_values, val_loss
```

```
313/313 [==============================] - 1s 4ms/step
<Figure size 432x288 with 0 Axes>
```

6.2.a

In [37]:

```
from keras.datasets import cifar10
from keras.preprocessing.image import ImageDataGenerator

(x_train, y_train), (x_test, y_test) = cifar10.load_data()

print(x_train.shape == (50000, 32, 32, 3))
print(x_test.shape == (10000, 32, 32, 3))
print(y_train.shape == (50000, 1))
print(y_test.shape == (10000, 1))

# Update y values to categorical.
num_classes = 10
y_train = to_categorical(y_train, num_classes)
y_test = to_categorical(y_test, num_classes)

x_train = x_train / 255
x_test = x_test / 255
```

```
Downloading data from https://www.cs.toronto.edu/~kriz/cifar-10-python.tar.gz
170498071/170498071 [==============================] - 24s 0us/step
```

```
True
True
True
True
```

In [38]:
```python
# Set up the model
model = models.Sequential()

model.add(layers.Conv2D(32, (3, 3), activation='relu', kernel_initializer='he_uniform', 
model.add(layers.Conv2D(32, (3, 3), activation='relu', kernel_initializer='he_uniform', 
model.add(layers.MaxPooling2D((2, 2)))
model.add(layers.Conv2D(64, (3, 3), activation='relu', kernel_initializer='he_uniform', 
model.add(layers.Conv2D(64, (3, 3), activation='relu', kernel_initializer='he_uniform', 
model.add(layers.MaxPooling2D((2, 2)))
model.add(layers.Conv2D(128, (3, 3), activation='relu', kernel_initializer='he_uniform',
model.add(layers.Conv2D(128, (3, 3), activation='relu', kernel_initializer='he_uniform',
model.add(layers.MaxPooling2D((2, 2)))
model.add(layers.Flatten())
model.add(layers.Dense(128, activation='relu', kernel_initializer='he_uniform'))
model.add(layers.Dense(10, activation='softmax'))
```

In [39]:
```python
model.summary()
```

```
Model: "sequential_4"
_________________________________________________________________
 Layer (type)                Output Shape              Param #   
=================================================================
 conv2d_12 (Conv2D)          (None, 32, 32, 32)        896       
                                                                 
 conv2d_13 (Conv2D)          (None, 32, 32, 32)        9248      
                                                                 
 max_pooling2d_8 (MaxPooling  (None, 16, 16, 32)       0         
 2D)                                                             
                                                                 
 conv2d_14 (Conv2D)          (None, 16, 16, 64)        18496     
                                                                 
 conv2d_15 (Conv2D)          (None, 16, 16, 64)        36928     
                                                                 
 max_pooling2d_9 (MaxPooling  (None, 8, 8, 64)         0         
 2D)                                                             
                                                                 
 conv2d_16 (Conv2D)          (None, 8, 8, 128)         73856     
                                                                 
 conv2d_17 (Conv2D)          (None, 8, 8, 128)         147584    
                                                                 
 max_pooling2d_10 (MaxPoolin  (None, 4, 4, 128)        0         
 g2D)                                                            
                                                                 
 flatten_4 (Flatten)         (None, 2048)              0         
                                                                 
 dense_8 (Dense)             (None, 128)               262272    
                                                                 
 dense_9 (Dense)             (None, 10)                1290      
                                                                 
=================================================================
Total params: 550,570
Trainable params: 550,570
Non-trainable params: 0
_________________________________________________________________
```

In [40]:
```python
# Compile the model.
model.compile(loss='categorical_crossentropy', optimizer=optimizers.RMSprop(learning_rat
```

In [41]:
```python
train_datagen = ImageDataGenerator()
test_datagen = ImageDataGenerator()
```

```
train_datagen.fit(x_train)
test_datagen.fit(x_test)

train_generator = train_datagen.flow(x_train, y_train, batch_size=64)

validation_generator = test_datagen.flow(x_test, y_test, batch_size=64)
```

In [42]:
```
history = model.fit(train_generator, steps_per_epoch=64, epochs=120, validation_data=(va
```

```
Epoch 1/120
64/64 [==============================] - 11s 154ms/step - loss: 2.2837 - acc: 0.1895 - v
al_loss: 1.9515 - val_acc: 0.3257
Epoch 2/120
64/64 [==============================] - 10s 153ms/step - loss: 1.8973 - acc: 0.3306 - v
al_loss: 1.8593 - val_acc: 0.3015
Epoch 3/120
64/64 [==============================] - 10s 150ms/step - loss: 1.6945 - acc: 0.3777 - v
al_loss: 1.6732 - val_acc: 0.3889
Epoch 4/120
64/64 [==============================] - 10s 149ms/step - loss: 1.5987 - acc: 0.4109 - v
al_loss: 2.1765 - val_acc: 0.3167
Epoch 5/120
64/64 [==============================] - 10s 156ms/step - loss: 1.5276 - acc: 0.4546 - v
al_loss: 1.4524 - val_acc: 0.4719
Epoch 6/120
64/64 [==============================] - 10s 151ms/step - loss: 1.4368 - acc: 0.4758 - v
al_loss: 1.4061 - val_acc: 0.5034
Epoch 7/120
64/64 [==============================] - 10s 152ms/step - loss: 1.3680 - acc: 0.5127 - v
al_loss: 1.3567 - val_acc: 0.5269
Epoch 8/120
64/64 [==============================] - 10s 153ms/step - loss: 1.3020 - acc: 0.5349 - v
al_loss: 1.2888 - val_acc: 0.5503
Epoch 9/120
64/64 [==============================] - 10s 155ms/step - loss: 1.2520 - acc: 0.5583 - v
al_loss: 1.2380 - val_acc: 0.5598
Epoch 10/120
64/64 [==============================] - 10s 154ms/step - loss: 1.1982 - acc: 0.5813 - v
al_loss: 1.2168 - val_acc: 0.5698
Epoch 11/120
64/64 [==============================] - 10s 149ms/step - loss: 1.1308 - acc: 0.6089 - v
al_loss: 1.0942 - val_acc: 0.6133
Epoch 12/120
64/64 [==============================] - 10s 150ms/step - loss: 1.1109 - acc: 0.6082 - v
al_loss: 1.1188 - val_acc: 0.6084
Epoch 13/120
64/64 [==============================] - 10s 154ms/step - loss: 1.0409 - acc: 0.6379 - v
al_loss: 1.1353 - val_acc: 0.6128
Epoch 14/120
64/64 [==============================] - 10s 151ms/step - loss: 1.0248 - acc: 0.6475 - v
al_loss: 1.0926 - val_acc: 0.6116
Epoch 15/120
64/64 [==============================] - 10s 151ms/step - loss: 1.0090 - acc: 0.6350 - v
al_loss: 1.0168 - val_acc: 0.6484
Epoch 16/120
64/64 [==============================] - 9s 149ms/step - loss: 0.9364 - acc: 0.6699 - va
l_loss: 1.0175 - val_acc: 0.6418
Epoch 17/120
64/64 [==============================] - 9s 150ms/step - loss: 0.9459 - acc: 0.6658 - va
l_loss: 0.9917 - val_acc: 0.6487
Epoch 18/120
64/64 [==============================] - 10s 156ms/step - loss: 0.8975 - acc: 0.6848 - v
al_loss: 0.9368 - val_acc: 0.6653
Epoch 19/120
```

```
64/64 [==============================] - 10s 158ms/step - loss: 0.8754 - acc: 0.6910 - v
al_loss: 1.0231 - val_acc: 0.6335
Epoch 20/120
64/64 [==============================] - 10s 153ms/step - loss: 0.8601 - acc: 0.7031 - v
al_loss: 0.9920 - val_acc: 0.6631
Epoch 21/120
64/64 [==============================] - 10s 152ms/step - loss: 0.8494 - acc: 0.7104 - v
al_loss: 0.8980 - val_acc: 0.6792
Epoch 22/120
64/64 [==============================] - 10s 161ms/step - loss: 0.8061 - acc: 0.7214 - v
al_loss: 0.9026 - val_acc: 0.6843
Epoch 23/120
64/64 [==============================] - 10s 156ms/step - loss: 0.7801 - acc: 0.7195 - v
al_loss: 0.8985 - val_acc: 0.7036
Epoch 24/120
64/64 [==============================] - 10s 149ms/step - loss: 0.7679 - acc: 0.7375 - v
al_loss: 0.8631 - val_acc: 0.7048
Epoch 25/120
64/64 [==============================] - 10s 163ms/step - loss: 0.7833 - acc: 0.7327 - v
al_loss: 0.8943 - val_acc: 0.6882
Epoch 26/120
64/64 [==============================] - 10s 156ms/step - loss: 0.7621 - acc: 0.7380 - v
al_loss: 0.8385 - val_acc: 0.7122
Epoch 27/120
64/64 [==============================] - 10s 156ms/step - loss: 0.7246 - acc: 0.7466 - v
al_loss: 1.2042 - val_acc: 0.6296
Epoch 28/120
64/64 [==============================] - 10s 157ms/step - loss: 0.6895 - acc: 0.7676 - v
al_loss: 0.8292 - val_acc: 0.7166
Epoch 29/120
64/64 [==============================] - 10s 154ms/step - loss: 0.6838 - acc: 0.7578 - v
al_loss: 0.9671 - val_acc: 0.6895
Epoch 30/120
64/64 [==============================] - 10s 153ms/step - loss: 0.6572 - acc: 0.7747 - v
al_loss: 0.8252 - val_acc: 0.7192
Epoch 31/120
64/64 [==============================] - 10s 158ms/step - loss: 0.6456 - acc: 0.7803 - v
al_loss: 0.8263 - val_acc: 0.7231
Epoch 32/120
64/64 [==============================] - 10s 156ms/step - loss: 0.6524 - acc: 0.7771 - v
al_loss: 0.8505 - val_acc: 0.7109
Epoch 33/120
64/64 [==============================] - 10s 154ms/step - loss: 0.6433 - acc: 0.7829 - v
al_loss: 0.7530 - val_acc: 0.7427
Epoch 34/120
64/64 [==============================] - 10s 154ms/step - loss: 0.6314 - acc: 0.7793 - v
al_loss: 0.7854 - val_acc: 0.7283
Epoch 35/120
64/64 [==============================] - 10s 158ms/step - loss: 0.6182 - acc: 0.7908 - v
al_loss: 0.8039 - val_acc: 0.7231
Epoch 36/120
64/64 [==============================] - 10s 151ms/step - loss: 0.6186 - acc: 0.7903 - v
al_loss: 0.8091 - val_acc: 0.7314
Epoch 37/120
64/64 [==============================] - 10s 158ms/step - loss: 0.5950 - acc: 0.7905 - v
al_loss: 0.7812 - val_acc: 0.7373
Epoch 38/120
64/64 [==============================] - 10s 159ms/step - loss: 0.5514 - acc: 0.8062 - v
al_loss: 0.7647 - val_acc: 0.7429
Epoch 39/120
64/64 [==============================] - 10s 158ms/step - loss: 0.5610 - acc: 0.8105 - v
al_loss: 0.8395 - val_acc: 0.7200
Epoch 40/120
64/64 [==============================] - 10s 156ms/step - loss: 0.5751 - acc: 0.8069 - v
al_loss: 0.7671 - val_acc: 0.7332
Epoch 41/120
```

```
64/64 [==============================] - 10s 155ms/step - loss: 0.5608 - acc: 0.8120 - v
al_loss: 0.7766 - val_acc: 0.7351
Epoch 42/120
64/64 [==============================] - 10s 150ms/step - loss: 0.5325 - acc: 0.8252 - v
al_loss: 0.7740 - val_acc: 0.7422
Epoch 43/120
64/64 [==============================] - 10s 157ms/step - loss: 0.5441 - acc: 0.8164 - v
al_loss: 0.7639 - val_acc: 0.7354
Epoch 44/120
64/64 [==============================] - 10s 163ms/step - loss: 0.5151 - acc: 0.8264 - v
al_loss: 0.8101 - val_acc: 0.7390
Epoch 45/120
64/64 [==============================] - 10s 155ms/step - loss: 0.5061 - acc: 0.8286 - v
al_loss: 0.8462 - val_acc: 0.7383
Epoch 46/120
64/64 [==============================] - 10s 157ms/step - loss: 0.5091 - acc: 0.8267 - v
al_loss: 0.8277 - val_acc: 0.7388
Epoch 47/120
64/64 [==============================] - 10s 155ms/step - loss: 0.4940 - acc: 0.8330 - v
al_loss: 0.7510 - val_acc: 0.7598
Epoch 48/120
64/64 [==============================] - 10s 163ms/step - loss: 0.4944 - acc: 0.8328 - v
al_loss: 0.7719 - val_acc: 0.7490
Epoch 49/120
64/64 [==============================] - 10s 162ms/step - loss: 0.4744 - acc: 0.8391 - v
al_loss: 0.7593 - val_acc: 0.7539
Epoch 50/120
64/64 [==============================] - 10s 155ms/step - loss: 0.4624 - acc: 0.8406 - v
al_loss: 0.7569 - val_acc: 0.7529
Epoch 51/120
64/64 [==============================] - 10s 156ms/step - loss: 0.4634 - acc: 0.8433 - v
al_loss: 0.7709 - val_acc: 0.7571
Epoch 52/120
64/64 [==============================] - 10s 163ms/step - loss: 0.4507 - acc: 0.8506 - v
al_loss: 0.8849 - val_acc: 0.7317
Epoch 53/120
64/64 [==============================] - 10s 160ms/step - loss: 0.4782 - acc: 0.8328 - v
al_loss: 0.7487 - val_acc: 0.7634
Epoch 54/120
64/64 [==============================] - 10s 160ms/step - loss: 0.4332 - acc: 0.8538 - v
al_loss: 0.8197 - val_acc: 0.7441
Epoch 55/120
64/64 [==============================] - 11s 167ms/step - loss: 0.4354 - acc: 0.8555 - v
al_loss: 0.7916 - val_acc: 0.7437
Epoch 56/120
64/64 [==============================] - 10s 156ms/step - loss: 0.4205 - acc: 0.8564 - v
al_loss: 0.8167 - val_acc: 0.7468
Epoch 57/120
64/64 [==============================] - 10s 158ms/step - loss: 0.3990 - acc: 0.8638 - v
al_loss: 0.9146 - val_acc: 0.7312
Epoch 58/120
64/64 [==============================] - 10s 161ms/step - loss: 0.4155 - acc: 0.8628 - v
al_loss: 0.8283 - val_acc: 0.7361
Epoch 59/120
64/64 [==============================] - 10s 157ms/step - loss: 0.3889 - acc: 0.8687 - v
al_loss: 0.7530 - val_acc: 0.7766
Epoch 60/120
64/64 [==============================] - 10s 159ms/step - loss: 0.3714 - acc: 0.8762 - v
al_loss: 0.8598 - val_acc: 0.7368
Epoch 61/120
64/64 [==============================] - 10s 163ms/step - loss: 0.4081 - acc: 0.8655 - v
al_loss: 0.8168 - val_acc: 0.7546
Epoch 62/120
64/64 [==============================] - 10s 154ms/step - loss: 0.4066 - acc: 0.8662 - v
al_loss: 0.8689 - val_acc: 0.7378
Epoch 63/120
```

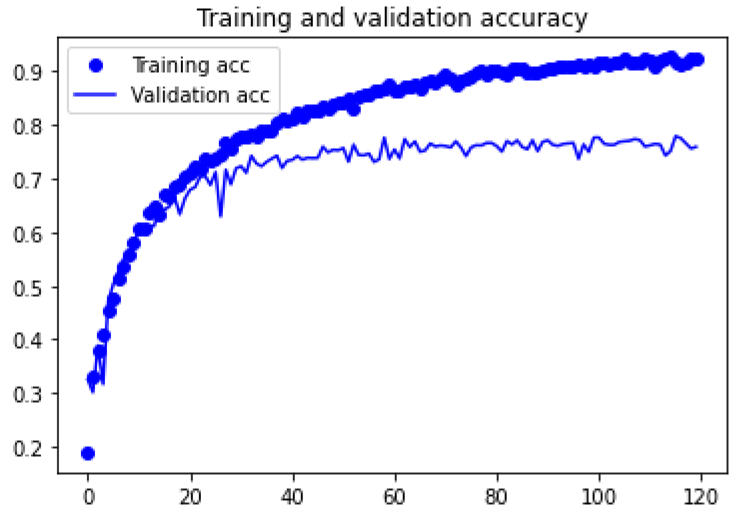
```
64/64 [==============================] - 10s 157ms/step - loss: 0.3699 - acc: 0.8730 - v
al_loss: 0.8052 - val_acc: 0.7739
Epoch 64/120
64/64 [==============================] - 10s 156ms/step - loss: 0.3867 - acc: 0.8738 - v
al_loss: 0.9079 - val_acc: 0.7590
Epoch 65/120
64/64 [==============================] - 10s 155ms/step - loss: 0.3718 - acc: 0.8760 - v
al_loss: 0.7583 - val_acc: 0.7695
Epoch 66/120
64/64 [==============================] - 10s 162ms/step - loss: 0.3929 - acc: 0.8689 - v
al_loss: 0.8614 - val_acc: 0.7502
Epoch 67/120
64/64 [==============================] - 10s 160ms/step - loss: 0.3528 - acc: 0.8809 - v
al_loss: 0.8858 - val_acc: 0.7527
Epoch 68/120
64/64 [==============================] - 10s 158ms/step - loss: 0.3462 - acc: 0.8821 - v
al_loss: 0.7771 - val_acc: 0.7656
Epoch 69/120
64/64 [==============================] - 11s 166ms/step - loss: 0.3370 - acc: 0.8806 - v
al_loss: 0.9363 - val_acc: 0.7600
Epoch 70/120
64/64 [==============================] - 10s 154ms/step - loss: 0.3575 - acc: 0.8850 - v
al_loss: 0.8636 - val_acc: 0.7620
Epoch 71/120
64/64 [==============================] - 10s 160ms/step - loss: 0.3295 - acc: 0.8948 - v
al_loss: 0.8711 - val_acc: 0.7607
Epoch 72/120
64/64 [==============================] - 10s 164ms/step - loss: 0.3508 - acc: 0.8855 - v
al_loss: 0.7938 - val_acc: 0.7583
Epoch 73/120
64/64 [==============================] - 10s 162ms/step - loss: 0.3600 - acc: 0.8757 - v
al_loss: 0.8092 - val_acc: 0.7698
Epoch 74/120
64/64 [==============================] - 10s 160ms/step - loss: 0.3419 - acc: 0.8818 - v
al_loss: 0.8552 - val_acc: 0.7583
Epoch 75/120
64/64 [==============================] - 10s 159ms/step - loss: 0.3350 - acc: 0.8867 - v
al_loss: 0.9377 - val_acc: 0.7422
Epoch 76/120
64/64 [==============================] - 10s 157ms/step - loss: 0.3168 - acc: 0.8923 - v
al_loss: 0.9443 - val_acc: 0.7585
Epoch 77/120
64/64 [==============================] - 10s 160ms/step - loss: 0.3301 - acc: 0.8967 - v
al_loss: 0.8843 - val_acc: 0.7625
Epoch 78/120
64/64 [==============================] - 10s 155ms/step - loss: 0.3028 - acc: 0.9004 - v
al_loss: 0.8726 - val_acc: 0.7620
Epoch 79/120
64/64 [==============================] - 10s 159ms/step - loss: 0.3299 - acc: 0.8960 - v
al_loss: 0.7875 - val_acc: 0.7666
Epoch 80/120
64/64 [==============================] - 10s 160ms/step - loss: 0.3059 - acc: 0.9011 - v
al_loss: 0.8610 - val_acc: 0.7666
Epoch 81/120
64/64 [==============================] - 10s 156ms/step - loss: 0.3053 - acc: 0.9014 - v
al_loss: 0.8430 - val_acc: 0.7607
Epoch 82/120
64/64 [==============================] - 10s 158ms/step - loss: 0.3092 - acc: 0.8972 - v
al_loss: 0.8416 - val_acc: 0.7505
Epoch 83/120
64/64 [==============================] - 10s 157ms/step - loss: 0.3159 - acc: 0.8928 - v
al_loss: 0.8408 - val_acc: 0.7703
Epoch 84/120
64/64 [==============================] - 10s 156ms/step - loss: 0.2961 - acc: 0.9043 - v
al_loss: 0.8574 - val_acc: 0.7639
Epoch 85/120
```

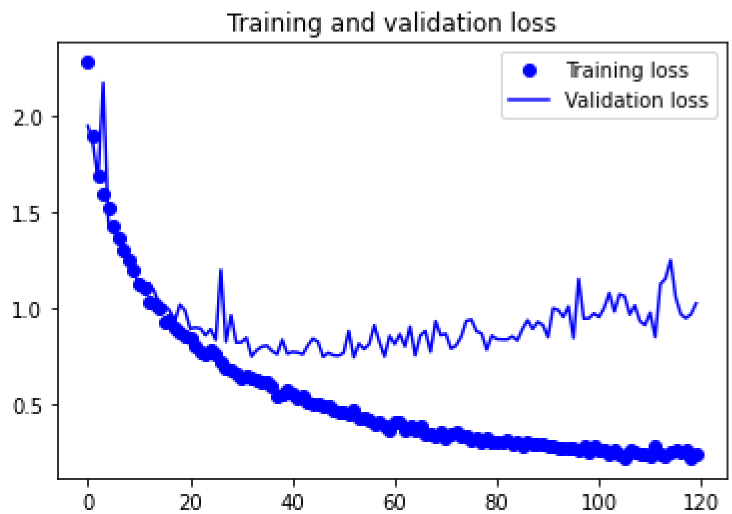
```
64/64 [==============================] - 10s 161ms/step - loss: 0.3052 - acc: 0.9038 - v
al_loss: 0.8362 - val_acc: 0.7715
Epoch 86/120
64/64 [==============================] - 10s 156ms/step - loss: 0.2907 - acc: 0.9058 - v
al_loss: 0.8960 - val_acc: 0.7590
Epoch 87/120
64/64 [==============================] - 10s 157ms/step - loss: 0.3054 - acc: 0.8972 - v
al_loss: 0.9416 - val_acc: 0.7549
Epoch 88/120
64/64 [==============================] - 10s 159ms/step - loss: 0.2978 - acc: 0.8999 - v
al_loss: 0.8958 - val_acc: 0.7717
Epoch 89/120
64/64 [==============================] - 10s 158ms/step - loss: 0.2980 - acc: 0.8992 - v
al_loss: 0.9326 - val_acc: 0.7515
Epoch 90/120
64/64 [==============================] - 10s 164ms/step - loss: 0.2945 - acc: 0.9031 - v
al_loss: 0.9165 - val_acc: 0.7688
Epoch 91/120
64/64 [==============================] - 10s 162ms/step - loss: 0.2853 - acc: 0.9059 - v
al_loss: 0.8531 - val_acc: 0.7715
Epoch 92/120
64/64 [==============================] - 10s 158ms/step - loss: 0.2878 - acc: 0.9048 - v
al_loss: 1.0040 - val_acc: 0.7629
Epoch 93/120
64/64 [==============================] - 10s 158ms/step - loss: 0.2744 - acc: 0.9099 - v
al_loss: 0.9948 - val_acc: 0.7612
Epoch 94/120
64/64 [==============================] - 10s 151ms/step - loss: 0.2796 - acc: 0.9104 - v
al_loss: 0.9586 - val_acc: 0.7646
Epoch 95/120
64/64 [==============================] - 9s 148ms/step - loss: 0.2763 - acc: 0.9106 - va
l_loss: 1.0135 - val_acc: 0.7659
Epoch 96/120
64/64 [==============================] - 10s 151ms/step - loss: 0.2768 - acc: 0.9106 - v
al_loss: 0.8484 - val_acc: 0.7666
Epoch 97/120
64/64 [==============================] - 10s 157ms/step - loss: 0.2649 - acc: 0.9131 - v
al_loss: 1.1568 - val_acc: 0.7371
Epoch 98/120
64/64 [==============================] - 10s 155ms/step - loss: 0.2826 - acc: 0.9092 - v
al_loss: 0.9502 - val_acc: 0.7639
Epoch 99/120
64/64 [==============================] - 10s 152ms/step - loss: 0.2598 - acc: 0.9124 - v
al_loss: 0.9511 - val_acc: 0.7500
Epoch 100/120
64/64 [==============================] - 11s 167ms/step - loss: 0.2888 - acc: 0.9087 - v
al_loss: 0.9767 - val_acc: 0.7766
Epoch 101/120
64/64 [==============================] - 11s 169ms/step - loss: 0.2657 - acc: 0.9158 - v
al_loss: 0.9581 - val_acc: 0.7769
Epoch 102/120
64/64 [==============================] - 10s 157ms/step - loss: 0.2639 - acc: 0.9153 - v
al_loss: 1.0056 - val_acc: 0.7651
Epoch 103/120
64/64 [==============================] - 10s 155ms/step - loss: 0.2489 - acc: 0.9126 - v
al_loss: 1.0832 - val_acc: 0.7632
Epoch 104/120
64/64 [==============================] - 10s 155ms/step - loss: 0.2660 - acc: 0.9185 - v
al_loss: 0.9866 - val_acc: 0.7632
Epoch 105/120
64/64 [==============================] - 10s 153ms/step - loss: 0.2427 - acc: 0.9170 - v
al_loss: 1.0767 - val_acc: 0.7686
Epoch 106/120
64/64 [==============================] - 10s 154ms/step - loss: 0.2288 - acc: 0.9236 - v
al_loss: 1.0658 - val_acc: 0.7698
Epoch 107/120
```

```
64/64 [==============================] - 10s 157ms/step - loss: 0.2632 - acc: 0.9165 - v
al_loss: 0.9711 - val_acc: 0.7708
Epoch 108/120
64/64 [==============================] - 11s 167ms/step - loss: 0.2534 - acc: 0.9150 - v
al_loss: 1.0187 - val_acc: 0.7739
Epoch 109/120
64/64 [==============================] - 11s 164ms/step - loss: 0.2475 - acc: 0.9177 - v
al_loss: 0.9365 - val_acc: 0.7720
Epoch 110/120
64/64 [==============================] - 10s 158ms/step - loss: 0.2487 - acc: 0.9229 - v
al_loss: 0.9160 - val_acc: 0.7600
Epoch 111/120
64/64 [==============================] - 11s 168ms/step - loss: 0.2352 - acc: 0.9197 - v
al_loss: 0.9809 - val_acc: 0.7622
Epoch 112/120
64/64 [==============================] - 10s 157ms/step - loss: 0.2838 - acc: 0.9075 - v
al_loss: 0.8533 - val_acc: 0.7649
Epoch 113/120
64/64 [==============================] - 10s 152ms/step - loss: 0.2439 - acc: 0.9216 - v
al_loss: 1.1290 - val_acc: 0.7634
Epoch 114/120
64/64 [==============================] - 10s 159ms/step - loss: 0.2296 - acc: 0.9258 - v
al_loss: 1.1578 - val_acc: 0.7432
Epoch 115/120
64/64 [==============================] - 10s 161ms/step - loss: 0.2505 - acc: 0.9270 - v
al_loss: 1.2551 - val_acc: 0.7515
Epoch 116/120
64/64 [==============================] - 10s 153ms/step - loss: 0.2622 - acc: 0.9175 - v
al_loss: 1.0616 - val_acc: 0.7793
Epoch 117/120
64/64 [==============================] - 10s 160ms/step - loss: 0.2575 - acc: 0.9131 - v
al_loss: 0.9726 - val_acc: 0.7756
Epoch 118/120
64/64 [==============================] - 10s 162ms/step - loss: 0.2656 - acc: 0.9170 - v
al_loss: 0.9519 - val_acc: 0.7659
Epoch 119/120
64/64 [==============================] - 10s 156ms/step - loss: 0.2246 - acc: 0.9238 - v
al_loss: 0.9725 - val_acc: 0.7563
Epoch 120/120
64/64 [==============================] - 11s 172ms/step - loss: 0.2456 - acc: 0.9246 - v
al_loss: 1.0291 - val_acc: 0.7593
```

In [43]:

```
# Display the curves of lass and accuracy during training

acc = history.history['acc']
val_acc = history.history['val_acc']
loss = history.history['loss']
val_loss = history.history['val_loss']

epochs = range(len(acc))

plt.plot(epochs, acc, 'bo', label='Training acc')
plt.plot(epochs, val_acc, 'b', label='Validation acc')
plt.title('Training and validation accuracy')
plt.legend()

plt.figure()

plt.plot(epochs, loss, 'bo', label='Training loss')
plt.plot(epochs, val_loss, 'b', label='Validation loss')
plt.title('Training and validation loss')
plt.legend()

plt.show()
```





In [44]:

```
# Use the model to make predictions
predictions = model.predict(x_test)
predicted_labels = np.argmax(predictions, axis=1)

# Save the model to the results folder
model.save('results/model-6-2-a.h5')
plt.savefig('results/validation-6-2-a.png')
np.savetxt('results/predictions-6-2-a.txt', predicted_labels, fmt='%d')
np.savetxt('results/metrics-6-2-a.txt', np.column_stack((loss_values, acc_values, val_lo
```

```
313/313 [==============================] - 6s 20ms/step
<Figure size 432x288 with 0 Axes>
```

6.2.b

In [45]:

```
(x_train, y_train), (x_test, y_test) = cifar10.load_data()

print(x_train.shape == (50000, 32, 32, 3))
print(x_test.shape == (10000, 32, 32, 3))
print(y_train.shape == (50000, 1))
print(y_test.shape == (10000, 1))

# Update y values to categorical.
num_classes = 10
y_train = to_categorical(y_train, num_classes)
y_test = to_categorical(y_test, num_classes)

x_train = x_train / 255
x_test = x_test / 255
```

```
True
True
True
True
```

In [46]:

```
# Set up the model
model = models.Sequential()

model.add(layers.Conv2D(32, (3, 3), activation='relu', kernel_initializer='he_uniform', 
model.add(layers.BatchNormalization())
model.add(layers.Conv2D(32, (3, 3), activation='relu', kernel_initializer='he_uniform', 
model.add(layers.MaxPooling2D((2, 2)))
model.add(layers.Dropout(0.2))
model.add(layers.Conv2D(64, (3, 3), activation='relu', kernel_initializer='he_uniform', 
model.add(layers.BatchNormalization())
model.add(layers.Conv2D(64, (3, 3), activation='relu', kernel_initializer='he_uniform', 
model.add(layers.MaxPooling2D((2, 2)))
model.add(layers.Dropout(0.3))
model.add(layers.Conv2D(128, (3, 3), activation='relu', kernel_initializer='he_uniform',
model.add(layers.BatchNormalization())
model.add(layers.Conv2D(128, (3, 3), activation='relu', kernel_initializer='he_uniform',
model.add(layers.MaxPooling2D((2, 2)))
model.add(layers.Dropout(0.4))
model.add(layers.Flatten())
model.add(layers.Dense(128, activation='relu', kernel_initializer='he_uniform'))
model.add(layers.Dropout(0.5))
model.add(layers.Dense(10, activation='softmax'))
```

In [47]:

```
model.summary()
```

```
Model: "sequential_5"
_________________________________________________________________
 Layer (type)                Output Shape              Param #   
=================================================================
 conv2d_18 (Conv2D)          (None, 32, 32, 32)        896       
                                                                 
 batch_normalization (BatchN  (None, 32, 32, 32)       128       
 ormalization)                                                   
                                                                 
 conv2d_19 (Conv2D)          (None, 32, 32, 32)        9248      
                                                                 
 max_pooling2d_11 (MaxPoolin  (None, 16, 16, 32)       0         
 g2D)                                                            
                                                                 
 dropout (Dropout)           (None, 16, 16, 32)        0         
                                                                 
 conv2d_20 (Conv2D)          (None, 16, 16, 64)        18496     
                                                                 
 batch_normalization_1 (Batc  (None, 16, 16, 64)       256       
 hNormalization)                                                 
                                                                 
 conv2d_21 (Conv2D)          (None, 16, 16, 64)        36928     
                                                                 
 max_pooling2d_12 (MaxPoolin  (None, 8, 8, 64)         0         
 g2D)                                                            
                                                                 
 dropout_1 (Dropout)         (None, 8, 8, 64)          0         
                                                                 
 conv2d_22 (Conv2D)          (None, 8, 8, 128)         73856     
                                                                 
 batch_normalization_2 (Batc  (None, 8, 8, 128)        512       
 hNormalization)                                                 
                                                                 
 conv2d_23 (Conv2D)          (None, 8, 8, 128)         147584    
```

```
 max_pooling2d_13 (MaxPoolin  (None, 4, 4, 128)         0
 g2D)

 dropout_2 (Dropout)         (None, 4, 4, 128)         0

 flatten_5 (Flatten)         (None, 2048)              0

 dense_10 (Dense)            (None, 128)               262272

 dropout_3 (Dropout)         (None, 128)               0

 dense_11 (Dense)            (None, 10)                1290

=================================================================
Total params: 551,466
Trainable params: 551,018
Non-trainable params: 448
_________________________________________________________________
```

In [48]:
```
model.compile(loss='categorical_crossentropy', optimizer='rmsprop', metrics=['acc'])
```

In [49]:
```
train_datagen = ImageDataGenerator(
            rescale=1./255,
            rotation_range=40,
            width_shift_range=0.2,
            height_shift_range=0.2,
            shear_range=0.2,
            zoom_range=0.2,
            horizontal_flip=True)

test_datagen = ImageDataGenerator(rescale=1./255)

train_datagen.fit(x_train)
test_datagen.fit(x_test)

train_generator = train_datagen.flow(x_train, y_train, batch_size=64)

validation_generator = test_datagen.flow(x_test, y_test, batch_size=64)
```

In [50]:
```
history = model.fit(train_generator, steps_per_epoch=64, epochs=120, validation_data=(va
```

```
Epoch 1/120
64/64 [==============================] - 14s 197ms/step - loss: 2.6795 - acc: 0.1140 - v
al_loss: 2.3342 - val_acc: 0.0972
Epoch 2/120
64/64 [==============================] - 12s 192ms/step - loss: 2.2685 - acc: 0.1357 - v
al_loss: 2.3555 - val_acc: 0.1060
Epoch 3/120
64/64 [==============================] - 12s 191ms/step - loss: 2.2394 - acc: 0.1548 - v
al_loss: 2.5423 - val_acc: 0.1016
Epoch 4/120
64/64 [==============================] - 13s 197ms/step - loss: 2.1930 - acc: 0.1614 - v
al_loss: 2.7448 - val_acc: 0.0947
Epoch 5/120
64/64 [==============================] - 13s 199ms/step - loss: 2.2122 - acc: 0.1692 - v
al_loss: 3.0198 - val_acc: 0.0955
Epoch 6/120
64/64 [==============================] - 13s 202ms/step - loss: 2.1620 - acc: 0.1814 - v
al_loss: 2.9739 - val_acc: 0.0955
Epoch 7/120
64/64 [==============================] - 12s 195ms/step - loss: 2.1528 - acc: 0.1914 - v
al_loss: 2.7050 - val_acc: 0.1060
Epoch 8/120
64/64 [==============================] - 13s 203ms/step - loss: 2.1492 - acc: 0.1904 - v
al_loss: 2.6081 - val_acc: 0.1038
```

```
Epoch 9/120
64/64 [==============================] - 13s 194ms/step - loss: 2.1124 - acc: 0.2068 - v
al_loss: 3.4202 - val_acc: 0.1033
Epoch 10/120
64/64 [==============================] - 12s 189ms/step - loss: 2.0969 - acc: 0.1990 - v
al_loss: 5.6036 - val_acc: 0.0989
Epoch 11/120
64/64 [==============================] - 12s 185ms/step - loss: 2.0631 - acc: 0.2324 - v
al_loss: 2.5543 - val_acc: 0.0991
Epoch 12/120
64/64 [==============================] - 12s 185ms/step - loss: 2.0876 - acc: 0.2021 - v
al_loss: 2.1136 - val_acc: 0.1685
Epoch 13/120
64/64 [==============================] - 12s 190ms/step - loss: 2.0699 - acc: 0.2085 - v
al_loss: 4.4744 - val_acc: 0.0994
Epoch 14/120
64/64 [==============================] - 12s 194ms/step - loss: 2.0321 - acc: 0.2292 - v
al_loss: 2.6532 - val_acc: 0.1001
Epoch 15/120
64/64 [==============================] - 12s 184ms/step - loss: 2.0289 - acc: 0.2405 - v
al_loss: 3.6351 - val_acc: 0.1035
Epoch 16/120
64/64 [==============================] - 12s 184ms/step - loss: 2.0079 - acc: 0.2261 - v
al_loss: 4.6778 - val_acc: 0.0999
Epoch 17/120
64/64 [==============================] - 12s 185ms/step - loss: 2.0173 - acc: 0.2475 - v
al_loss: 2.2745 - val_acc: 0.1248
Epoch 18/120
64/64 [==============================] - 12s 185ms/step - loss: 1.9848 - acc: 0.2429 - v
al_loss: 25.6225 - val_acc: 0.1052
Epoch 19/120
64/64 [==============================] - 12s 188ms/step - loss: 1.9324 - acc: 0.2881 - v
al_loss: 22.8345 - val_acc: 0.0959
Epoch 20/120
64/64 [==============================] - 12s 184ms/step - loss: 1.9391 - acc: 0.2773 - v
al_loss: 9.2863 - val_acc: 0.1084
Epoch 21/120
64/64 [==============================] - 12s 187ms/step - loss: 1.9207 - acc: 0.2878 - v
al_loss: 21.8551 - val_acc: 0.0979
Epoch 22/120
64/64 [==============================] - 12s 184ms/step - loss: 1.8960 - acc: 0.3052 - v
al_loss: 2.2366 - val_acc: 0.1467
Epoch 23/120
64/64 [==============================] - 12s 185ms/step - loss: 1.8779 - acc: 0.3215 - v
al_loss: 2.6222 - val_acc: 0.1204
Epoch 24/120
64/64 [==============================] - 12s 191ms/step - loss: 1.8561 - acc: 0.3237 - v
al_loss: 2.6817 - val_acc: 0.1099
Epoch 25/120
64/64 [==============================] - 12s 180ms/step - loss: 1.8620 - acc: 0.3203 - v
al_loss: 18.8771 - val_acc: 0.0994
Epoch 26/120
64/64 [==============================] - 12s 184ms/step - loss: 1.8556 - acc: 0.3252 - v
al_loss: 4.3199 - val_acc: 0.1116
Epoch 27/120
64/64 [==============================] - 12s 181ms/step - loss: 1.8231 - acc: 0.3362 - v
al_loss: 6.9402 - val_acc: 0.1458
Epoch 28/120
64/64 [==============================] - 11s 179ms/step - loss: 1.8334 - acc: 0.3406 - v
al_loss: 2.6702 - val_acc: 0.1472
Epoch 29/120
64/64 [==============================] - 12s 183ms/step - loss: 1.7951 - acc: 0.3569 - v
al_loss: 5.2655 - val_acc: 0.1235
Epoch 30/120
64/64 [==============================] - 12s 181ms/step - loss: 1.7803 - acc: 0.3665 - v
al_loss: 6.7972 - val_acc: 0.1021
```

```
Epoch 31/120
64/64 [==============================] - 12s 180ms/step - loss: 1.7462 - acc: 0.3760 - v
al_loss: 8.1150 - val_acc: 0.0984
Epoch 32/120
64/64 [==============================] - 12s 180ms/step - loss: 1.7658 - acc: 0.3618 - v
al_loss: 5.2472 - val_acc: 0.0984
Epoch 33/120
64/64 [==============================] - 12s 180ms/step - loss: 1.7609 - acc: 0.3762 - v
al_loss: 8.7620 - val_acc: 0.0991
Epoch 34/120
64/64 [==============================] - 11s 179ms/step - loss: 1.7300 - acc: 0.3689 - v
al_loss: 3.3961 - val_acc: 0.1387
Epoch 35/120
64/64 [==============================] - 12s 184ms/step - loss: 1.7325 - acc: 0.3833 - v
al_loss: 3.0554 - val_acc: 0.1208
Epoch 36/120
64/64 [==============================] - 12s 181ms/step - loss: 1.7176 - acc: 0.3826 - v
al_loss: 22.0960 - val_acc: 0.1030
Epoch 37/120
64/64 [==============================] - 12s 180ms/step - loss: 1.7091 - acc: 0.3735 - v
al_loss: 4.7590 - val_acc: 0.1028
Epoch 38/120
64/64 [==============================] - 11s 179ms/step - loss: 1.7134 - acc: 0.3872 - v
al_loss: 3.7109 - val_acc: 0.1453
Epoch 39/120
64/64 [==============================] - 11s 178ms/step - loss: 1.6972 - acc: 0.3953 - v
al_loss: 2.5095 - val_acc: 0.1038
Epoch 40/120
64/64 [==============================] - 12s 185ms/step - loss: 1.6602 - acc: 0.4033 - v
al_loss: 2.2329 - val_acc: 0.2437
Epoch 41/120
64/64 [==============================] - 11s 178ms/step - loss: 1.7036 - acc: 0.3865 - v
al_loss: 3.7691 - val_acc: 0.1426
Epoch 42/120
64/64 [==============================] - 11s 177ms/step - loss: 1.6525 - acc: 0.4116 - v
al_loss: 5.3318 - val_acc: 0.1094
Epoch 43/120
64/64 [==============================] - 12s 181ms/step - loss: 1.6694 - acc: 0.4016 - v
al_loss: 6.9960 - val_acc: 0.1035
Epoch 44/120
64/64 [==============================] - 12s 180ms/step - loss: 1.6931 - acc: 0.3984 - v
al_loss: 18.6203 - val_acc: 0.0950
Epoch 45/120
64/64 [==============================] - 12s 184ms/step - loss: 1.6627 - acc: 0.4048 - v
al_loss: 6.1255 - val_acc: 0.0986
Epoch 46/120
64/64 [==============================] - 12s 181ms/step - loss: 1.6580 - acc: 0.4121 - v
al_loss: 8.1010 - val_acc: 0.1047
Epoch 47/120
64/64 [==============================] - 12s 182ms/step - loss: 1.6298 - acc: 0.4258 - v
al_loss: 4.7100 - val_acc: 0.1057
Epoch 48/120
64/64 [==============================] - 11s 179ms/step - loss: 1.6720 - acc: 0.4048 - v
al_loss: 24.1461 - val_acc: 0.1030
Epoch 49/120
64/64 [==============================] - 12s 180ms/step - loss: 1.6235 - acc: 0.4187 - v
al_loss: 2.2656 - val_acc: 0.1619
Epoch 50/120
64/64 [==============================] - 12s 186ms/step - loss: 1.6531 - acc: 0.4243 - v
al_loss: 16.9186 - val_acc: 0.1006
Epoch 51/120
64/64 [==============================] - 12s 181ms/step - loss: 1.6370 - acc: 0.4111 - v
al_loss: 2.6770 - val_acc: 0.1086
Epoch 52/120
64/64 [==============================] - 12s 180ms/step - loss: 1.6228 - acc: 0.4189 - v
al_loss: 2.2710 - val_acc: 0.1655
```

```
Epoch 53/120
64/64 [==============================] - 12s 182ms/step - loss: 1.6007 - acc: 0.4373 - v
al_loss: 13.9716 - val_acc: 0.1411
Epoch 54/120
64/64 [==============================] - 12s 185ms/step - loss: 1.6402 - acc: 0.4148 - v
al_loss: 2.6467 - val_acc: 0.1382
Epoch 55/120
64/64 [==============================] - 12s 187ms/step - loss: 1.6080 - acc: 0.4290 - v
al_loss: 2.4264 - val_acc: 0.1279
Epoch 56/120
64/64 [==============================] - 12s 183ms/step - loss: 1.6197 - acc: 0.4143 - v
al_loss: 3.2394 - val_acc: 0.1030
Epoch 57/120
64/64 [==============================] - 12s 184ms/step - loss: 1.6221 - acc: 0.4270 - v
al_loss: 5.5914 - val_acc: 0.1001
Epoch 58/120
64/64 [==============================] - 12s 186ms/step - loss: 1.5795 - acc: 0.4395 - v
al_loss: 15.0164 - val_acc: 0.1384
Epoch 59/120
64/64 [==============================] - 12s 182ms/step - loss: 1.5850 - acc: 0.4387 - v
al_loss: 60.7235 - val_acc: 0.1003
Epoch 60/120
64/64 [==============================] - 12s 185ms/step - loss: 1.5874 - acc: 0.4199 - v
al_loss: 35.0550 - val_acc: 0.1067
Epoch 61/120
64/64 [==============================] - 12s 186ms/step - loss: 1.5785 - acc: 0.4487 - v
al_loss: 45.2798 - val_acc: 0.1040
Epoch 62/120
64/64 [==============================] - 12s 186ms/step - loss: 1.6174 - acc: 0.4258 - v
al_loss: 24.6839 - val_acc: 0.1016
Epoch 63/120
64/64 [==============================] - 12s 181ms/step - loss: 1.5860 - acc: 0.4397 - v
al_loss: 35.6904 - val_acc: 0.1028
Epoch 64/120
64/64 [==============================] - 12s 183ms/step - loss: 1.6031 - acc: 0.4465 - v
al_loss: 2.4387 - val_acc: 0.1328
Epoch 65/120
64/64 [==============================] - 12s 188ms/step - loss: 1.5828 - acc: 0.4343 - v
al_loss: 4.1101 - val_acc: 0.1050
Epoch 66/120
64/64 [==============================] - 12s 189ms/step - loss: 1.6119 - acc: 0.4246 - v
al_loss: 12.7003 - val_acc: 0.0972
Epoch 67/120
64/64 [==============================] - 12s 183ms/step - loss: 1.5821 - acc: 0.4412 - v
al_loss: 38.7816 - val_acc: 0.1062
Epoch 68/120
64/64 [==============================] - 12s 184ms/step - loss: 1.5740 - acc: 0.4480 - v
al_loss: 10.6478 - val_acc: 0.0991
Epoch 69/120
64/64 [==============================] - 12s 184ms/step - loss: 1.5839 - acc: 0.4397 - v
al_loss: 45.5764 - val_acc: 0.1409
Epoch 70/120
64/64 [==============================] - 12s 187ms/step - loss: 1.5674 - acc: 0.4529 - v
al_loss: 1.7534 - val_acc: 0.4211
Epoch 71/120
64/64 [==============================] - 12s 184ms/step - loss: 1.5477 - acc: 0.4417 - v
al_loss: 2.5413 - val_acc: 0.0991
Epoch 72/120
64/64 [==============================] - 12s 184ms/step - loss: 1.5697 - acc: 0.4436 - v
al_loss: 9.7079 - val_acc: 0.1042
Epoch 73/120
64/64 [==============================] - 12s 185ms/step - loss: 1.5700 - acc: 0.4453 - v
al_loss: 33.3889 - val_acc: 0.1006
Epoch 74/120
64/64 [==============================] - 12s 187ms/step - loss: 1.5663 - acc: 0.4487 - v
al_loss: 2.8972 - val_acc: 0.1106
```

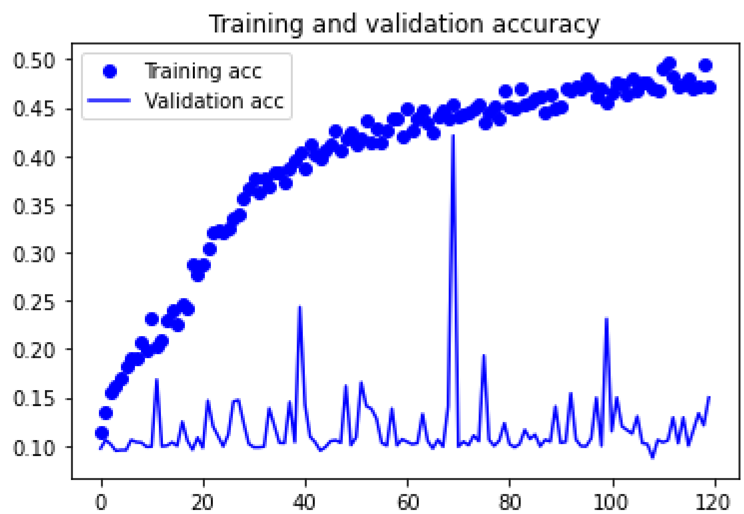
```
Epoch 75/120
64/64 [==============================] - 12s 186ms/step - loss: 1.5501 - acc: 0.4526 - v
al_loss: 16.0890 - val_acc: 0.1047
Epoch 76/120
64/64 [==============================] - 12s 188ms/step - loss: 1.5885 - acc: 0.4348 - v
al_loss: 3.2735 - val_acc: 0.1934
Epoch 77/120
64/64 [==============================] - 12s 186ms/step - loss: 1.5815 - acc: 0.4461 - v
al_loss: 8.4401 - val_acc: 0.1062
Epoch 78/120
64/64 [==============================] - 12s 187ms/step - loss: 1.5617 - acc: 0.4513 - v
al_loss: 19.7906 - val_acc: 0.0999
Epoch 79/120
64/64 [==============================] - 12s 184ms/step - loss: 1.5753 - acc: 0.4392 - v
al_loss: 48.7871 - val_acc: 0.1052
Epoch 80/120
64/64 [==============================] - 12s 185ms/step - loss: 1.5155 - acc: 0.4688 - v
al_loss: 2.3215 - val_acc: 0.1233
Epoch 81/120
64/64 [==============================] - 12s 195ms/step - loss: 1.5680 - acc: 0.4517 - v
al_loss: 4.1019 - val_acc: 0.1018
Epoch 82/120
64/64 [==============================] - 12s 188ms/step - loss: 1.5720 - acc: 0.4497 - v
al_loss: 3.8959 - val_acc: 0.0984
Epoch 83/120
64/64 [==============================] - 12s 195ms/step - loss: 1.5282 - acc: 0.4707 - v
al_loss: 24.6894 - val_acc: 0.1018
Epoch 84/120
64/64 [==============================] - 12s 185ms/step - loss: 1.5480 - acc: 0.4539 - v
al_loss: 10.4061 - val_acc: 0.1165
Epoch 85/120
64/64 [==============================] - 12s 184ms/step - loss: 1.5227 - acc: 0.4551 - v
al_loss: 6.1698 - val_acc: 0.1069
Epoch 86/120
64/64 [==============================] - 13s 205ms/step - loss: 1.5728 - acc: 0.4592 - v
al_loss: 18.3342 - val_acc: 0.1116
Epoch 87/120
64/64 [==============================] - 12s 185ms/step - loss: 1.5443 - acc: 0.4617 - v
al_loss: 39.6418 - val_acc: 0.0994
Epoch 88/120
64/64 [==============================] - 11s 179ms/step - loss: 1.5719 - acc: 0.4458 - v
al_loss: 22.6992 - val_acc: 0.1062
Epoch 89/120
64/64 [==============================] - 11s 179ms/step - loss: 1.5308 - acc: 0.4629 - v
al_loss: 4.3248 - val_acc: 0.1042
Epoch 90/120
64/64 [==============================] - 12s 181ms/step - loss: 1.5665 - acc: 0.4502 - v
al_loss: 2.2610 - val_acc: 0.1406
Epoch 91/120
64/64 [==============================] - 11s 179ms/step - loss: 1.5561 - acc: 0.4523 - v
al_loss: 3.7280 - val_acc: 0.1030
Epoch 92/120
64/64 [==============================] - 12s 182ms/step - loss: 1.5209 - acc: 0.4690 - v
al_loss: 62.4323 - val_acc: 0.1040
Epoch 93/120
64/64 [==============================] - 12s 183ms/step - loss: 1.5190 - acc: 0.4670 - v
al_loss: 4.3345 - val_acc: 0.1543
Epoch 94/120
64/64 [==============================] - 12s 182ms/step - loss: 1.5025 - acc: 0.4722 - v
al_loss: 5.4102 - val_acc: 0.1064
Epoch 95/120
64/64 [==============================] - 12s 182ms/step - loss: 1.4923 - acc: 0.4700 - v
al_loss: 29.8517 - val_acc: 0.0996
Epoch 96/120
64/64 [==============================] - 12s 187ms/step - loss: 1.5059 - acc: 0.4795 - v
al_loss: 2.6820 - val_acc: 0.0994
```

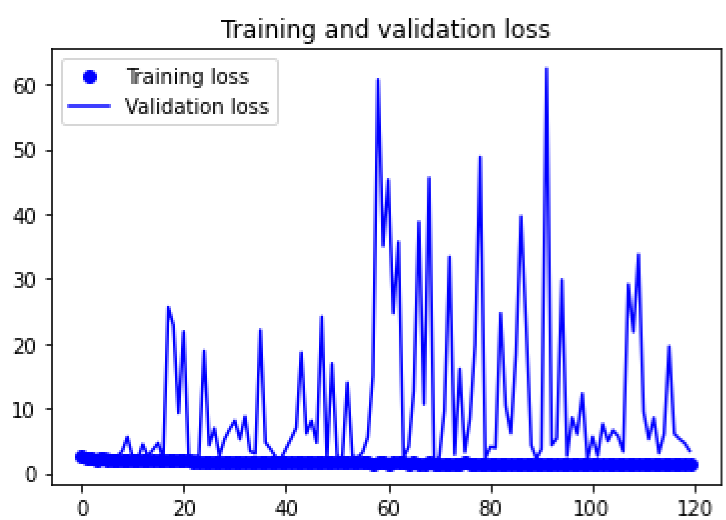
```
Epoch 97/120
64/64 [==============================] - 12s 183ms/step - loss: 1.5120 - acc: 0.4746 - v
al_loss: 8.5809 - val_acc: 0.1077
Epoch 98/120
64/64 [==============================] - 11s 180ms/step - loss: 1.5237 - acc: 0.4609 - v
al_loss: 6.0476 - val_acc: 0.1497
Epoch 99/120
64/64 [==============================] - 11s 178ms/step - loss: 1.4986 - acc: 0.4699 - v
al_loss: 12.2934 - val_acc: 0.0999
Epoch 100/120
64/64 [==============================] - 12s 181ms/step - loss: 1.5363 - acc: 0.4560 - v
al_loss: 2.1246 - val_acc: 0.2314
Epoch 101/120
64/64 [==============================] - 12s 186ms/step - loss: 1.4937 - acc: 0.4653 - v
al_loss: 5.6431 - val_acc: 0.1150
Epoch 102/120
64/64 [==============================] - 11s 179ms/step - loss: 1.4845 - acc: 0.4770 - v
al_loss: 2.6797 - val_acc: 0.1499
Epoch 103/120
64/64 [==============================] - 12s 182ms/step - loss: 1.4896 - acc: 0.4736 - v
al_loss: 7.6049 - val_acc: 0.1201
Epoch 104/120
64/64 [==============================] - 12s 184ms/step - loss: 1.5169 - acc: 0.4634 - v
al_loss: 4.9664 - val_acc: 0.1162
Epoch 105/120
64/64 [==============================] - 12s 181ms/step - loss: 1.4544 - acc: 0.4810 - v
al_loss: 6.6424 - val_acc: 0.1125
Epoch 106/120
64/64 [==============================] - 12s 184ms/step - loss: 1.5117 - acc: 0.4680 - v
al_loss: 5.7358 - val_acc: 0.1304
Epoch 107/120
64/64 [==============================] - 12s 182ms/step - loss: 1.4901 - acc: 0.4761 - v
al_loss: 3.3516 - val_acc: 0.1028
Epoch 108/120
64/64 [==============================] - 12s 187ms/step - loss: 1.4897 - acc: 0.4771 - v
al_loss: 29.1532 - val_acc: 0.1016
Epoch 109/120
64/64 [==============================] - 12s 181ms/step - loss: 1.5243 - acc: 0.4705 - v
al_loss: 21.8093 - val_acc: 0.0874
Epoch 110/120
64/64 [==============================] - 12s 182ms/step - loss: 1.5117 - acc: 0.4685 - v
al_loss: 33.7608 - val_acc: 0.1060
Epoch 111/120
64/64 [==============================] - 12s 188ms/step - loss: 1.4673 - acc: 0.4915 - v
al_loss: 9.4681 - val_acc: 0.1035
Epoch 112/120
64/64 [==============================] - 12s 189ms/step - loss: 1.4845 - acc: 0.4963 - v
al_loss: 5.2646 - val_acc: 0.1055
Epoch 113/120
64/64 [==============================] - 12s 184ms/step - loss: 1.4931 - acc: 0.4827 - v
al_loss: 8.5561 - val_acc: 0.1292
Epoch 114/120
64/64 [==============================] - 12s 182ms/step - loss: 1.4944 - acc: 0.4719 - v
al_loss: 3.1014 - val_acc: 0.1025
Epoch 115/120
64/64 [==============================] - 12s 186ms/step - loss: 1.5078 - acc: 0.4749 - v
al_loss: 6.0575 - val_acc: 0.1292
Epoch 116/120
64/64 [==============================] - 12s 191ms/step - loss: 1.5056 - acc: 0.4800 - v
al_loss: 19.5801 - val_acc: 0.1001
Epoch 117/120
64/64 [==============================] - 12s 184ms/step - loss: 1.5153 - acc: 0.4705 - v
al_loss: 6.1042 - val_acc: 0.1165
Epoch 118/120
64/64 [==============================] - 12s 187ms/step - loss: 1.5024 - acc: 0.4712 - v
al_loss: 5.3027 - val_acc: 0.1335
```

```
Epoch 119/120
64/64 [==============================] - 12s 187ms/step - loss: 1.4704 - acc: 0.4956 - v
al_loss: 4.6801 - val_acc: 0.1211
Epoch 120/120
64/64 [==============================] - 12s 188ms/step - loss: 1.4875 - acc: 0.4727 - v
al_loss: 3.4648 - val_acc: 0.1499
```

In [51]:

```
# Plot the data.

acc = history.history['acc']
val_acc = history.history['val_acc']
loss = history.history['loss']
val_loss = history.history['val_loss']

epochs = range(len(acc))

plt.plot(epochs, acc, 'bo', label='Training acc')
plt.plot(epochs, val_acc, 'b', label='Validation acc')
plt.title('Training and validation accuracy')
plt.legend()

plt.figure()

plt.plot(epochs, loss, 'bo', label='Training loss')
plt.plot(epochs, val_loss, 'b', label='Validation loss')
plt.title('Training and validation loss')
plt.legend()

plt.show()
```





In [52]:

```
# Use the model to make predictions
```

```
predictions = model.predict(x_test)
predicted_labels = np.argmax(predictions, axis=1)

# Save the model to the results folder
model.save('results/model-6-2-b.h5')
plt.savefig('results/validation-6-2-b.png')
np.savetxt('results/predictions-6-2-b.txt', predicted_labels, fmt='%d')
np.savetxt('results/metrics-6-2-b.txt', np.column_stack((loss_values, acc_values, val_lo
```

```
313/313 [==============================] - 8s 24ms/step
<Figure size 432x288 with 0 Axes>
```

6.3

In [54]:

```
from tensorflow.keras.applications.resnet50 import ResNet50
from tensorflow.keras.preprocessing import image
from tensorflow.keras.applications.resnet50 import preprocess_input, decode_predictions

model = ResNet50(weights='imagenet')
filepath = 'images'
output = []

# Loop over each image to make predictions
for file in os.listdir(filepath):
    image_path = os.path.join(filepath, file)
    img = image.load_img(image_path, target_size=(224, 224))

    x = image.img_to_array(img)
    x = np.expand_dims(x, axis = 0)
    x = preprocess_input(x)

    # Make the predictions
    predictions = model.predict(x)
    prediction_name = str(image_path)
    info = str('Predicted: ' + str(decode_predictions(predictions, top=5)[0]))
    output.append(prediction_name)
    output.append(info)
    print(f'{prediction_name} \n {info}')
```

```
1/1 [==============================] - 1s 1s/step
Downloading data from https://storage.googleapis.com/download.tensorflow.org/data/imagen
et_class_index.json
35363/35363 [==============================] - 0s 1us/step
images\amy-hirschi-szrJ3wjzOMg-unsplash.jpg
 Predicted: [('n03291819', 'envelope', 0.21261807), ('n03642806', 'laptop', 0.08052259),
('n04118776', 'rule', 0.080337815), ('n03179701', 'desk', 0.07715189), ('n04548280', 'wa
ll_clock', 0.07612319)]
1/1 [==============================] - 0s 117ms/step
images\daniel-bonilla-MVZlkv_G4zQ-unsplash.jpg
 Predicted: [('n03803284', 'muzzle', 0.9223493), ('n02403003', 'ox', 0.031144816), ('n03
868863', 'oxygen_mask', 0.0288808), ('n03424325', 'gasmask', 0.0042149327), ('n0209233
9', 'Weimaraner', 0.0014808961)]
1/1 [==============================] - 0s 103ms/step
images\dushawn-jovic-B3fgTrrgsiI-unsplash.jpg
 Predicted: [('n02807133', 'bathing_cap', 0.4402964), ('n03710637', 'maillot', 0.1761608
1), ('n02892767', 'brassiere', 0.14404574), ('n03710721', 'maillot', 0.13406157), ('n028
37789', 'bikini', 0.07064589)]
1/1 [==============================] - 0s 103ms/step
images\erik-mclean-ZRns2R5azu0-unsplash.jpg
 Predicted: [('n03594945', 'jeep', 0.29830098), ('n04467665', 'trailer_truck', 0.2078867
1), ('n04037443', 'racer', 0.12850937), ('n03770679', 'minivan', 0.05470578), ('n0393063
0', 'pickup', 0.04988277)]
1/1 [==============================] - 0s 101ms/step
images\intricate-explorer-ndmaGsIr6E4-unsplash.jpg
```

```
 Predicted: [('n13040303', 'stinkhorn', 0.13910884), ('n12985857', 'coral_fungus', 0.038
33431), ('n01773549', 'barn_spider', 0.03268129), ('n01950731', 'sea_slug', 0.03092150
8), ('n04604644', 'worm_fence', 0.02760182)]
1/1 [==============================] - 0s 101ms/step
images\leio-mclaren-FwdZYz0yc9g-unsplash.jpg
 Predicted: [('n03888257', 'parachute', 0.44306922), ('n03773504', 'missile', 0.1711833
8), ('n09472597', 'volcano', 0.10201463), ('n04552348', 'warplane', 0.10144949), ('n0400
8634', 'projectile', 0.056138314)]
1/1 [==============================] - 0s 102ms/step
images\manja-vitolic-gKXKBY-C-Dk-unsplash.jpg
 Predicted: [('n02124075', 'Egyptian_cat', 0.61982495), ('n02123597', 'Siamese_cat', 0.1
3296269), ('n02909870', 'bucket', 0.052781727), ('n02123045', 'tabby', 0.03871195), ('n0
3958227', 'plastic_bag', 0.020926394)]
1/1 [==============================] - 0s 101ms/step
images\obie-fernandez-0GFNAelMPZA-unsplash.jpg
 Predicted: [('n03250847', 'drumstick', 0.2297712), ('n04584207', 'wig', 0.129819), ('n0
3594734', 'jean', 0.066456355), ('n02963159', 'cardigan', 0.05127348), ('n03770439', 'mi
niskirt', 0.04637946)]
1/1 [==============================] - 0s 100ms/step
images\quino-al-8gWEAAXJjtI-unsplash.jpg
 Predicted: [('n03187595', 'dial_telephone', 0.9591948), ('n03902125', 'pay-phone', 0.03
9810065), ('n04328186', 'stopwatch', 0.000522505), ('n04548280', 'wall_clock', 0.0001079
9182), ('n04069434', 'reflex_camera', 7.302062e-05)]
1/1 [==============================] - 0s 102ms/step
images\richard-brutyo-Sg3XwuEpybU-unsplash.jpg
 Predicted: [('n02099601', 'golden_retriever', 0.9678861), ('n02099712', 'Labrador_retri
ever', 0.007136036), ('n02102318', 'cocker_spaniel', 0.0039223228), ('n02085620', 'Chihu
ahua', 0.0025483514), ('n02113624', 'toy_poodle', 0.0023548473)]
```

In [56]:

```
# Save predictions to a file
if not os.path.exists('results/predictions/resnet50'):
    os.makedirs('results/predictions/resnet50')

outfile = 'results/predictions/resnet50/predictions-6-3.txt'
with open(outfile, 'w') as f:
    for line in output:
        f.write("%s\n" % line)
```