monkey_species

May 13, 2019

1 Udacity Project

1.0.1 By Mayank Bhatnagar

This project is about 'Fine-Grained Categorization'. The technique is advance compared to object identification problems. We can use such techniques to recognize different species of flowers, birds or any other species. It has wider use in identifying and grouping objects and even evaluating whether we are finding a new species next existing or explored earlier.

1.1 1. Understanding Input Data and Loading Data Set

Key Import Statements

```
In [1]: ## Main Import statements
        import numpy as np
        import pandas as pd
        import cv2
        import matplotlib.pyplot as plt
        import time
        %matplotlib inline

        ## Loading data set
        from sklearn.datasets import load_files
        from keras.utils import np_utils
        from glob import glob
```

```
Using TensorFlow backend.
```

Summary of Data Set used in the project

```
In [2]: ## Define function to read content of a text file
        def load_file(name):
            rows = pd.read_csv(name, sep=",")
            return rows
```

```
In [3]: ## Load the information for all the loaded data set
        rows = load_file('monkey_labels.csv')
```

```
## Print all the training and validation data summary
print (rows)
```

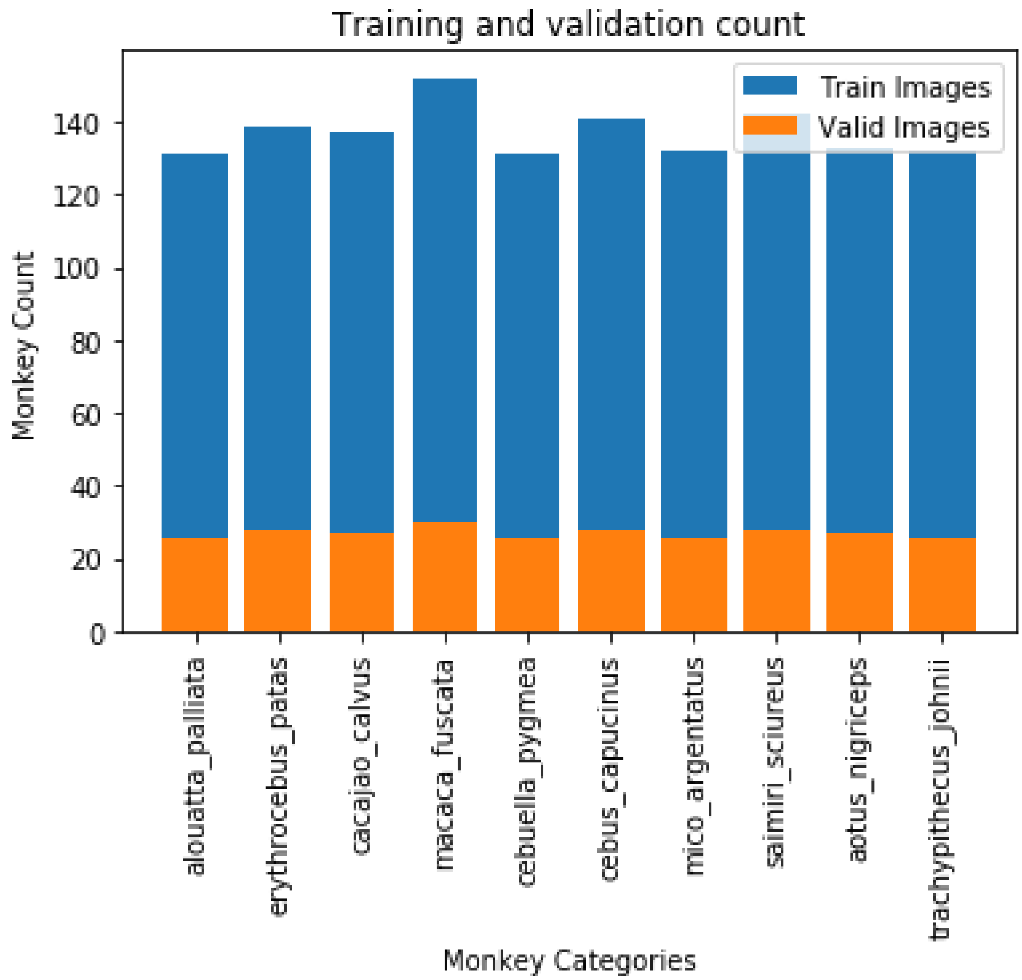
```
  Label              Latin Name                Common Name  Train Images  \
0    n0      alouatta_palliata              mantled_howler           131
1    n1     erythrocebus_patas                patas_monkey           139
2    n2         cacajao_calvus                  bald_uakari           137
3    n3         macaca_fuscata             japanese_macaque           152
4    n4        cebuella_pygmea               pygmy_marmoset           131
5    n5       cebus_capucinus         white_headed_capuchin           141
6    n6        mico_argentatus            silvery_marmoset           132
7    n7      saimiri_sciureus        common_squirrel_monkey           142
8    n8       aotus_nigriceps     black_headed_night_monkey           133
9    n9  trachypithecus_johnii                nilgiri_langur           132

   Validation Images
0                 26
1                 28
2                 27
3                 30
4                 26
5                 28
6                 26
7                 28
8                 27
9                 26
```

In [4]:
```
## Presenting the data in a graphical format

plt.title('Training and validation count')
index = np.arange(len(rows))
plt.bar(index, rows['Train Images'], label='Train Images')
plt.bar(index, rows['Validation Images'], label='Valid Images')
plt.xlabel('Monkey Categories', fontsize=10)
plt.ylabel('Monkey Count', fontsize=10)
plt.xticks(index, rows['Latin Name'], fontsize=10, rotation=90)
plt.legend(loc=1)

plt.show()
```



Load of Data Set

```
In [5]: ## define function to load training and validation datasets
        def load_dataset(path):
            data = load_files(path)
            mon_files = np.array(data['filenames'])
            ## print(mon_files)

            """
            y = data['target']
            n = y.shape[0]
            print(y,n)
            cate = np.zeros((272,11),dtype=np.int)
            print(cate)
            print(len(cate))
            print(np.arange((n)))
```

```
            print(cate[np.arange(n), y])
            cate[np.arange(n), y] = 0
            """

            ## Check Below
            mon_targets = np_utils.to_categorical(np.array(data['target']),11)
            return mon_files, mon_targets
```

In [6]:
```
## load train and validation datasets
train_files, train_targets = load_dataset('./training')
valid_files, valid_targets = load_dataset('./validation')

## load list of monkey names
mon_cate = [item[11:-1] for item in sorted(glob("./training/*/"))]
mon_name = []

for idx, item in enumerate(mon_cate):
    ## print(rows.loc[rows['Label'] == item, 'Latin Name'].iloc[0])
    mon_name.append(rows[rows['Label']== item]['Latin Name'].iloc[0])

## print statistics about the dataset
print('There are %d total monkey categories.\n' % len(mon_cate))
print('%s \n' % rows['Latin Name'].iloc[0:len(mon_cate)])
print('There are %s total monkey images.' % len(np.hstack([train_files, valid_files])))
print('There are %d training monkey images.' % len(train_files))
print('There are %d validation monkey images.' % len(valid_files))
```

```
There are 10 total monkey categories.

0        alouatta_palliata
1       erythrocebus_patas
2           cacajao_calvus
3           macaca_fuscata
4          cebuella_pygmea
5          cebus_capucinus
6          mico_argentatus
7         saimiri_sciureus
8          aotus_nigriceps
9     trachypithecus_johnii
Name: Latin Name, dtype: object

There are 1365 total monkey images.
There are 1093 training monkey images.
There are 272 validation monkey images.
```

1.2 2. Pre-Processing of Data

```
In [7]: from keras.preprocessing import image
        from tqdm import tqdm

        def path_to_tensor(img_path):
            ## loads RGB image as PIL.Image.Image type
            img = image.load_img(img_path, target_size=(224, 224))
            ## convert PIL.Image.Image type to 3D tensor with shape (224, 224, 3)
            x = image.img_to_array(img)
            ## convert 3D tensor to 4D tensor with shape (1, 224, 224, 3) and return 4D tensor
            return np.expand_dims(x, axis=0)

        def paths_to_tensor(img_paths):
            list_of_tensors = [path_to_tensor(img_path) for img_path in tqdm(img_paths)]
            return np.vstack(list_of_tensors)
```

```
In [ ]: ## display the image, along with bounding box
        ## plt.imshow(cv_rgb)
        ## plt.show()
```

```
In [8]: ## Rescaling the images between 0 and 1
        from PIL import ImageFile
        ImageFile.LOAD_TRUNCATED_IMAGES = True

        ## pre-process the data for Keras
        train_tensors = paths_to_tensor(train_files).astype('float32')/255
        valid_tensors = paths_to_tensor(valid_files).astype('float32')/255
```

```
100%|| 1093/1093 [00:31<00:00, 34.20it/s]
100%|| 272/272 [00:08<00:00, 30.74it/s]
```

1.3 3. Creating a Benchmark ResNet50 and VGG16 model

```
In [9]: ## Common functions for Creating Benchmarking

        from keras.layers import Conv2D, MaxPooling2D, GlobalAveragePooling2D
        from keras.layers import Dropout, Flatten, Dense
        from keras.models import Sequential
        from keras import backend

        from keras.models import Sequential, Model
        from keras.optimizers import Adam, SGD

        from keras.applications.resnet50 import ResNet50
        from keras.applications.vgg16 import VGG16
```

1.3.1 Creating ResNET50 model using three modularized functions

1. ResNet50_mod => Create ResNet50 model
2. train_model => Train the created model
3. create_graph => Plot Accuracy and Loss functions

In [10]:
```
## Creating a ResNet50 Model after clearing session
def ResNet50_mod():

    ## To clear memory from the model. Remove the comment and run.
    backend.clear_session()

    ## To create a new model
    ResNet50_model = ResNet50(weights='imagenet',include_top=False,input_shape=(224,224
    x = ResNet50_model.output
    x = Dense(512, activation='relu')(x)
    x = Dropout(0.5)(x)
    x = Dense(512, activation='relu')(x)
    x = Dropout(0.5)(x)
    output = Dense(11, activation='softmax', name='custom_output')(x) ## Need to check
    ResNet50_T_model = Model(inputs=ResNet50_model.input, outputs = output)

    ## Suppressing retraining of already trained model
    for layer in ResNet50_model.layers:
        layer.trainable = False

    ## Compiling the model
    ResNet50_T_model.compile(optimizer='rmsprop', loss='categorical_crossentropy', metr
    ## Checking how does the complie process work with Adm optimizer
    ## ResNet50_T_model.compile(Adam(lr=0.001), loss='categorical_crossentropy', metric
    return ResNet50_T_model
```

In [11]:
```
## This is an additional logic
def train_model(model, epochs_val, batch_val):

    acc = []
    val_acc = []
    loss = []
    val_loss = []
    test_accu = []
    time_taken = []

    for e_val in epochs_val:
        for b_val in batch_val:
            print (e_val,' ',b_val)

            ## Calling function to create the model
            #monkey_model = monkey_mod()
```

```
            ## Start tracking the time
            start_time = time.time()

            ## Training the model and capturing the history
            model_history = model.fit(train_tensors, train_targets,
                                     validation_data=(valid_tensors, valid_targets),
                                     epochs=e_val, batch_size=b_val, callbacks=None, verbose


            ## Stop tracking the time
            end_time = time.time()
            time_taken.append(end_time - start_time)

            ## Capture accuracy and loss information
            acc.append(model_history.history['acc'])
            val_acc.append(model_history.history['val_acc'])
            loss.append(model_history.history['loss'])
            val_loss.append(model_history.history['val_loss'])

            ## Getting prediction and accuracy details
            mon_predictions = [np.argmax(model.predict(np.expand_dims(tensor, axis=0)))

            ## report test accuracy
            test_accuracy = 100*np.sum(np.array(mon_predictions)==np.argmax(valid_targe
            print('Test accuracy: %.4f%%' % test_accuracy)
            test_accu.append(test_accuracy)

            ## Printing the training time taken
            print('Training Time: %s sec' % time_taken)

    ## Create Plot for the model
    ## create_graph(epochs_val, batch_val, acc, val_acc, loss, val_loss, test_accu)
    return (acc, val_acc, loss, val_loss, test_accu, time_taken)
```

In [12]:
```
## Plotting all the graphs together as a generic function
def create_graph(epochs_val, batch_val, acc, val_acc, loss, val_loss, test_acc, time_ta

    x = 0
    ## Iterate for all the values for each epochs
    for e_val in epochs_val:
        for b_val in batch_val:

            plt.title('Training and validation accuracy: Epoch (' + str(e_val) + '), Ba
            plt.plot(range(1,e_val+1), acc[x], 'red', label='Training acc')
            plt.plot(range(1,e_val+1), val_acc[x], 'blue', label='Validation acc')
            plt.xlabel('No. of Epochs')
            plt.ylabel('Accuracy Value')
            plt.legend()
```

```
                plt.figure()
                plt.title('Training and validation loss: Epoch (' + str(e_val) + '), Batch
                plt.plot(range(1,e_val+1), loss[x], 'red', label='Training loss')
                plt.plot(range(1,e_val+1), val_loss[x], 'blue', label='Validation loss')
                plt.xlabel('No. of Epochs')
                plt.ylabel('Loss Value')
                plt.legend()
                plt.show()

                ## Printing accuracy information for each batch run
                print('Test Accuracy: %.4f%%' % test_accu[x])

                ## Printing the training time taken
                print('Training Time: %s sec' % time_taken[x])

                x += 1
```

In [13]:
```
### Create model and train

## Submitting with below
epochs_val = [10,20,40]
batch_val = [16,32]

## Test with below
#epochs_val = [10]
#batch_val = [32]

## Create and train the model
model = ResNet50_mod()
(acc, val_acc, loss, val_loss, test_accu, time_taken) = train_model(model, epochs_val,
```

```
Downloading data from https://github.com/fchollet/deep-learning-models/releases/download/v0.2/re
94658560/94653016 [==============================] - 2s 0us/step
10   16
Train on 1093 samples, validate on 272 samples
Epoch 1/10
1093/1093 [==============================] - 20s 18ms/step - loss: 2.2134 - acc: 0.3641 - val_lo
Epoch 2/10
1093/1093 [==============================] - 17s 15ms/step - loss: 0.9300 - acc: 0.6962 - val_lo
Epoch 3/10
1093/1093 [==============================] - 17s 15ms/step - loss: 0.6200 - acc: 0.8060 - val_lo
Epoch 4/10
1093/1093 [==============================] - 17s 15ms/step - loss: 0.4452 - acc: 0.8609 - val_lo
Epoch 5/10
1093/1093 [==============================] - 17s 15ms/step - loss: 0.4610 - acc: 0.8591 - val_lo
Epoch 6/10
1093/1093 [==============================] - 17s 15ms/step - loss: 0.3650 - acc: 0.8838 - val_lo
```

```
Epoch 7/10
1093/1093 [==============================] - 17s 15ms/step - loss: 0.3368 - acc: 0.9058 - val_lo
Epoch 8/10
1093/1093 [==============================] - 17s 15ms/step - loss: 0.3253 - acc: 0.9067 - val_lo
Epoch 9/10
1093/1093 [==============================] - 17s 15ms/step - loss: 0.3015 - acc: 0.9250 - val_lo
Epoch 10/10
1093/1093 [==============================] - 17s 15ms/step - loss: 0.3120 - acc: 0.9149 - val_lo
Test accuracy: 26.8382%
Training Time: [170.2390956878662] sec
10   32
Train on 1093 samples, validate on 272 samples
Epoch 1/10
1093/1093 [==============================] - 14s 13ms/step - loss: 0.2222 - acc: 0.9396 - val_lo
Epoch 2/10
1093/1093 [==============================] - 14s 13ms/step - loss: 0.1388 - acc: 0.9579 - val_lo
Epoch 3/10
1093/1093 [==============================] - 14s 13ms/step - loss: 0.1159 - acc: 0.9716 - val_lo
Epoch 4/10
1093/1093 [==============================] - 14s 13ms/step - loss: 0.1317 - acc: 0.9561 - val_lo
Epoch 5/10
1093/1093 [==============================] - 14s 13ms/step - loss: 0.1202 - acc: 0.9643 - val_lo
Epoch 6/10
1093/1093 [==============================] - 14s 13ms/step - loss: 0.1210 - acc: 0.9643 - val_lo
Epoch 7/10
1093/1093 [==============================] - 14s 13ms/step - loss: 0.1010 - acc: 0.9771 - val_lo
Epoch 8/10
1093/1093 [==============================] - 14s 13ms/step - loss: 0.1459 - acc: 0.9652 - val_lo
Epoch 9/10
1093/1093 [==============================] - 14s 13ms/step - loss: 0.0896 - acc: 0.9753 - val_lo
Epoch 10/10
1093/1093 [==============================] - 14s 13ms/step - loss: 0.1389 - acc: 0.9689 - val_lo
Test accuracy: 94.4853%
Training Time: [170.2390956878662, 142.29511785507202] sec
20   16
Train on 1093 samples, validate on 272 samples
Epoch 1/20
1093/1093 [==============================] - 17s 15ms/step - loss: 0.2715 - acc: 0.9360 - val_lo
Epoch 2/20
1093/1093 [==============================] - 17s 15ms/step - loss: 0.2990 - acc: 0.9433 - val_lo
Epoch 3/20
1093/1093 [==============================] - 17s 15ms/step - loss: 0.2058 - acc: 0.9588 - val_lo
Epoch 4/20
1093/1093 [==============================] - 17s 15ms/step - loss: 0.2282 - acc: 0.9478 - val_lo
Epoch 5/20
1093/1093 [==============================] - 17s 15ms/step - loss: 0.2044 - acc: 0.9515 - val_lo
Epoch 6/20
1093/1093 [==============================] - 17s 15ms/step - loss: 0.2038 - acc: 0.9543 - val_lo
```

```
Epoch 7/20
1093/1093 [==============================] - 17s 15ms/step - loss: 0.1734 - acc: 0.9515 - val_lo
Epoch 8/20
1093/1093 [==============================] - 17s 15ms/step - loss: 0.2757 - acc: 0.9469 - val_lo
Epoch 9/20
1093/1093 [==============================] - 17s 15ms/step - loss: 0.2379 - acc: 0.9524 - val_lo
Epoch 10/20
1093/1093 [==============================] - 17s 15ms/step - loss: 0.1662 - acc: 0.9661 - val_lo
Epoch 11/20
1093/1093 [==============================] - 17s 15ms/step - loss: 0.2266 - acc: 0.9552 - val_lo
Epoch 12/20
1093/1093 [==============================] - 17s 15ms/step - loss: 0.2701 - acc: 0.9478 - val_lo
Epoch 13/20
1093/1093 [==============================] - 17s 15ms/step - loss: 0.2461 - acc: 0.9570 - val_lo
Epoch 14/20
1093/1093 [==============================] - 17s 15ms/step - loss: 0.2505 - acc: 0.9506 - val_lo
Epoch 15/20
1093/1093 [==============================] - 17s 15ms/step - loss: 0.1603 - acc: 0.9643 - val_lo
Epoch 16/20
1093/1093 [==============================] - 17s 15ms/step - loss: 0.0906 - acc: 0.9744 - val_lo
Epoch 17/20
1093/1093 [==============================] - 17s 15ms/step - loss: 0.1712 - acc: 0.9634 - val_lo
Epoch 18/20
1093/1093 [==============================] - 17s 15ms/step - loss: 0.2550 - acc: 0.9543 - val_lo
Epoch 19/20
1093/1093 [==============================] - 17s 15ms/step - loss: 0.1801 - acc: 0.9652 - val_lo
Epoch 20/20
1093/1093 [==============================] - 17s 15ms/step - loss: 0.2187 - acc: 0.9671 - val_lo
Test accuracy: 95.9559%
Training Time: [170.2390956878662, 142.29511785507202, 333.946396112442] sec
20   32
Train on 1093 samples, validate on 272 samples
Epoch 1/20
1093/1093 [==============================] - 14s 13ms/step - loss: 0.0742 - acc: 0.9826 - val_lo
Epoch 2/20
1093/1093 [==============================] - 14s 13ms/step - loss: 0.1111 - acc: 0.9790 - val_lo
Epoch 3/20
1093/1093 [==============================] - 14s 13ms/step - loss: 0.0931 - acc: 0.9817 - val_lo
Epoch 4/20
1093/1093 [==============================] - 14s 13ms/step - loss: 0.1245 - acc: 0.9780 - val_lo
Epoch 5/20
1093/1093 [==============================] - 14s 13ms/step - loss: 0.1183 - acc: 0.9726 - val_lo
Epoch 6/20
1093/1093 [==============================] - 14s 13ms/step - loss: 0.0668 - acc: 0.9854 - val_lo
Epoch 7/20
1093/1093 [==============================] - 14s 13ms/step - loss: 0.1044 - acc: 0.9808 - val_lo
Epoch 8/20
1093/1093 [==============================] - 14s 13ms/step - loss: 0.0390 - acc: 0.9909 - val_lo
```

```
Epoch 9/20
1093/1093 [==============================] - 14s 13ms/step - loss: 0.1060 - acc: 0.9762 - val_lo
Epoch 10/20
1093/1093 [==============================] - 14s 13ms/step - loss: 0.0621 - acc: 0.9854 - val_lo
Epoch 11/20
1093/1093 [==============================] - 14s 13ms/step - loss: 0.0656 - acc: 0.9872 - val_lo
Epoch 12/20
1093/1093 [==============================] - 14s 13ms/step - loss: 0.0763 - acc: 0.9780 - val_lo
Epoch 13/20
1093/1093 [==============================] - 14s 13ms/step - loss: 0.0665 - acc: 0.9881 - val_lo
Epoch 14/20
1093/1093 [==============================] - 14s 13ms/step - loss: 0.0956 - acc: 0.9808 - val_lo
Epoch 15/20
1093/1093 [==============================] - 14s 13ms/step - loss: 0.1055 - acc: 0.9817 - val_lo
Epoch 16/20
1093/1093 [==============================] - 14s 13ms/step - loss: 0.0614 - acc: 0.9835 - val_lo
Epoch 17/20
1093/1093 [==============================] - 14s 13ms/step - loss: 0.0215 - acc: 0.9936 - val_lo
Epoch 18/20
1093/1093 [==============================] - 14s 13ms/step - loss: 0.0904 - acc: 0.9817 - val_lo
Epoch 19/20
1093/1093 [==============================] - 14s 13ms/step - loss: 0.0227 - acc: 0.9936 - val_lo
Epoch 20/20
1093/1093 [==============================] - 14s 13ms/step - loss: 0.1597 - acc: 0.9744 - val_lo
Test accuracy: 95.5882%
Training Time: [170.2390956878662, 142.29511785507202, 333.946396112442, 284.02110266685486] sec
40   16
Train on 1093 samples, validate on 272 samples
Epoch 1/40
1093/1093 [==============================] - 17s 15ms/step - loss: 0.1403 - acc: 0.9671 - val_lo
Epoch 2/40
1093/1093 [==============================] - 17s 15ms/step - loss: 0.2596 - acc: 0.9625 - val_lo
Epoch 3/40
1093/1093 [==============================] - 17s 15ms/step - loss: 0.1169 - acc: 0.9744 - val_lo
Epoch 4/40
1093/1093 [==============================] - 17s 15ms/step - loss: 0.1478 - acc: 0.9707 - val_lo
Epoch 5/40
1093/1093 [==============================] - 17s 15ms/step - loss: 0.2241 - acc: 0.9698 - val_lo
Epoch 6/40
1093/1093 [==============================] - 17s 15ms/step - loss: 0.1359 - acc: 0.9716 - val_lo
Epoch 7/40
1093/1093 [==============================] - 17s 15ms/step - loss: 0.1044 - acc: 0.9780 - val_lo
Epoch 8/40
1093/1093 [==============================] - 17s 15ms/step - loss: 0.1829 - acc: 0.9680 - val_lo
Epoch 9/40
1093/1093 [==============================] - 17s 15ms/step - loss: 0.1831 - acc: 0.9707 - val_lo
Epoch 10/40
1093/1093 [==============================] - 17s 15ms/step - loss: 0.2316 - acc: 0.9680 - val_lo
```

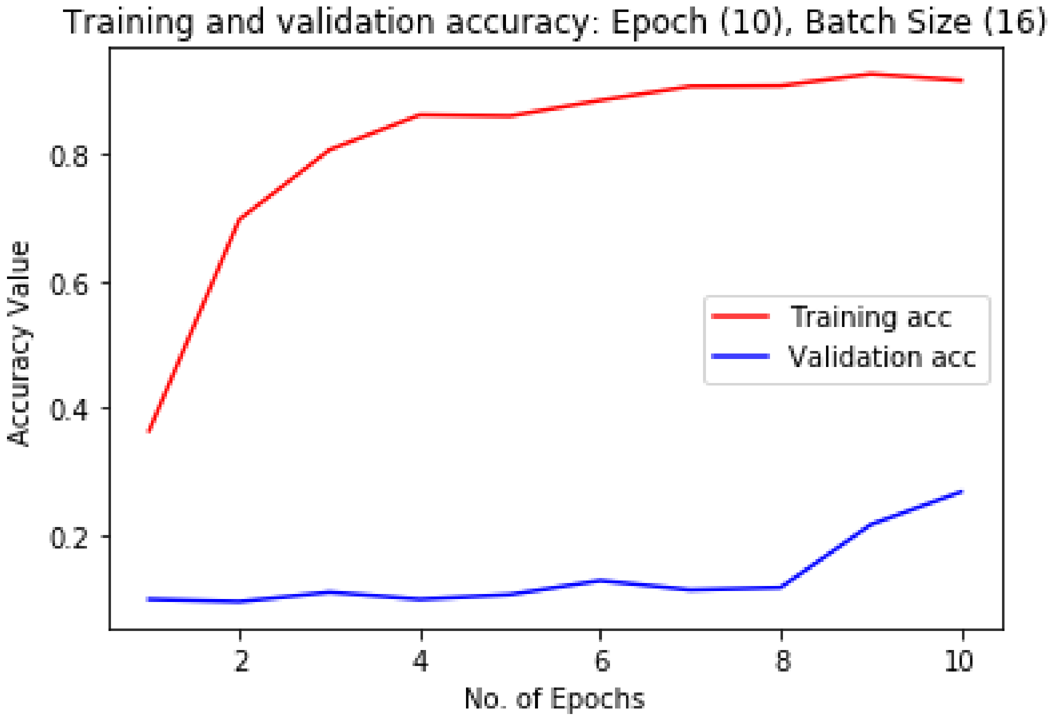
```
Epoch 11/40
1093/1093 [==============================] - 17s 15ms/step - loss: 0.2436 - acc: 0.9661 - val_lo
Epoch 12/40
1093/1093 [==============================] - 17s 15ms/step - loss: 0.0909 - acc: 0.9780 - val_lo
Epoch 13/40
1093/1093 [==============================] - 17s 15ms/step - loss: 0.2399 - acc: 0.9643 - val_lo
Epoch 14/40
1093/1093 [==============================] - 17s 15ms/step - loss: 0.1634 - acc: 0.9753 - val_lo
Epoch 15/40
1093/1093 [==============================] - 17s 15ms/step - loss: 0.1396 - acc: 0.9726 - val_lo
Epoch 16/40
1093/1093 [==============================] - 17s 15ms/step - loss: 0.1774 - acc: 0.9680 - val_lo
Epoch 17/40
1093/1093 [==============================] - 17s 15ms/step - loss: 0.2535 - acc: 0.9652 - val_lo
Epoch 18/40
1093/1093 [==============================] - 17s 15ms/step - loss: 0.2087 - acc: 0.9698 - val_lo
Epoch 19/40
1093/1093 [==============================] - 17s 15ms/step - loss: 0.1631 - acc: 0.9735 - val_lo
Epoch 20/40
1093/1093 [==============================] - 17s 15ms/step - loss: 0.0744 - acc: 0.9799 - val_lo
Epoch 21/40
1093/1093 [==============================] - 17s 15ms/step - loss: 0.1324 - acc: 0.9780 - val_lo
Epoch 22/40
1093/1093 [==============================] - 17s 15ms/step - loss: 0.1832 - acc: 0.9744 - val_lo
Epoch 23/40
1093/1093 [==============================] - 17s 15ms/step - loss: 0.1897 - acc: 0.9790 - val_lo
Epoch 24/40
1093/1093 [==============================] - 17s 15ms/step - loss: 0.1414 - acc: 0.9808 - val_lo
Epoch 25/40
1093/1093 [==============================] - 17s 15ms/step - loss: 0.1253 - acc: 0.9817 - val_lo
Epoch 26/40
1093/1093 [==============================] - 17s 15ms/step - loss: 0.1407 - acc: 0.9771 - val_lo
Epoch 27/40
1093/1093 [==============================] - 17s 15ms/step - loss: 0.1695 - acc: 0.9707 - val_lo
Epoch 28/40
1093/1093 [==============================] - 17s 15ms/step - loss: 0.1501 - acc: 0.9780 - val_lo
Epoch 29/40
1093/1093 [==============================] - 17s 15ms/step - loss: 0.1972 - acc: 0.9643 - val_lo
Epoch 30/40
1093/1093 [==============================] - 17s 15ms/step - loss: 0.1759 - acc: 0.9753 - val_lo
Epoch 31/40
1093/1093 [==============================] - 17s 15ms/step - loss: 0.1885 - acc: 0.9716 - val_lo
Epoch 32/40
1093/1093 [==============================] - 17s 15ms/step - loss: 0.1144 - acc: 0.9826 - val_lo
Epoch 33/40
1093/1093 [==============================] - 17s 15ms/step - loss: 0.2026 - acc: 0.9716 - val_lo
Epoch 34/40
1093/1093 [==============================] - 17s 15ms/step - loss: 0.2090 - acc: 0.9744 - val_lo
```

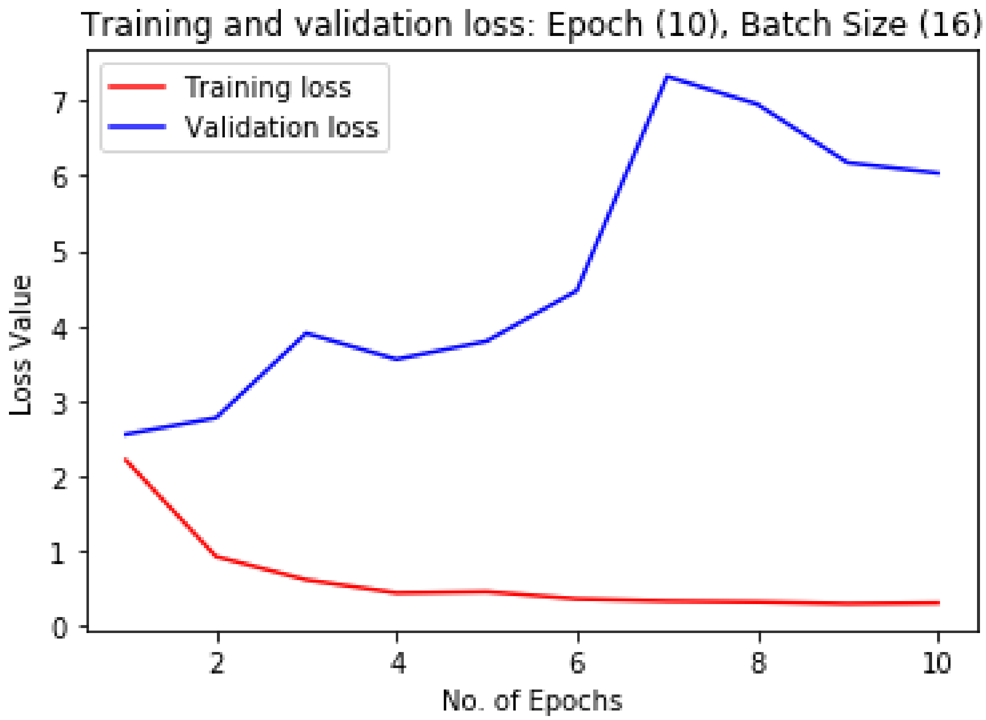
```
Epoch 35/40
1093/1093 [==============================] - 17s 15ms/step - loss: 0.1773 - acc: 0.9771 - val_lo
Epoch 36/40
1093/1093 [==============================] - 17s 15ms/step - loss: 0.1923 - acc: 0.9716 - val_lo
Epoch 37/40
1093/1093 [==============================] - 17s 15ms/step - loss: 0.2406 - acc: 0.9652 - val_lo
Epoch 38/40
1093/1093 [==============================] - 17s 15ms/step - loss: 0.1357 - acc: 0.9780 - val_lo
Epoch 39/40
1093/1093 [==============================] - 17s 15ms/step - loss: 0.1834 - acc: 0.9726 - val_lo
Epoch 40/40
1093/1093 [==============================] - 16s 15ms/step - loss: 0.1728 - acc: 0.9780 - val_lo
Test accuracy: 95.2206%
Training Time: [170.2390956878662, 142.29511785507202, 333.946396112442, 284.02110266685486, 666
40   32
Train on 1093 samples, validate on 272 samples
Epoch 1/40
1093/1093 [==============================] - 14s 13ms/step - loss: 0.0550 - acc: 0.9899 - val_lo
Epoch 2/40
1093/1093 [==============================] - 14s 13ms/step - loss: 0.0898 - acc: 0.9899 - val_lo
Epoch 3/40
1093/1093 [==============================] - 14s 13ms/step - loss: 0.0807 - acc: 0.9890 - val_lo
Epoch 4/40
1093/1093 [==============================] - 14s 13ms/step - loss: 0.0355 - acc: 0.9936 - val_lo
Epoch 5/40
1093/1093 [==============================] - 14s 13ms/step - loss: 0.0797 - acc: 0.9872 - val_lo
Epoch 6/40
1093/1093 [==============================] - 14s 13ms/step - loss: 0.0788 - acc: 0.9844 - val_lo
Epoch 7/40
1093/1093 [==============================] - 14s 13ms/step - loss: 0.0874 - acc: 0.9872 - val_lo
Epoch 8/40
1093/1093 [==============================] - 14s 13ms/step - loss: 0.0641 - acc: 0.9899 - val_lo
Epoch 9/40
1093/1093 [==============================] - 14s 13ms/step - loss: 0.0660 - acc: 0.9881 - val_lo
Epoch 10/40
1093/1093 [==============================] - 14s 13ms/step - loss: 0.0755 - acc: 0.9854 - val_lo
Epoch 11/40
1093/1093 [==============================] - 14s 13ms/step - loss: 0.0808 - acc: 0.9890 - val_lo
Epoch 12/40
1093/1093 [==============================] - 14s 13ms/step - loss: 0.0696 - acc: 0.9863 - val_lo
Epoch 13/40
1093/1093 [==============================] - 14s 13ms/step - loss: 0.0526 - acc: 0.9890 - val_lo
Epoch 14/40
1093/1093 [==============================] - 14s 13ms/step - loss: 0.0625 - acc: 0.9881 - val_lo
Epoch 15/40
1093/1093 [==============================] - 14s 13ms/step - loss: 0.0046 - acc: 0.9963 - val_lo
Epoch 16/40
1093/1093 [==============================] - 14s 13ms/step - loss: 0.0843 - acc: 0.9909 - val_lo
```

```
Epoch 17/40
1093/1093 [==============================] - 14s 13ms/step - loss: 0.0698 - acc: 0.9890 - val_lo
Epoch 18/40
1093/1093 [==============================] - 14s 13ms/step - loss: 0.0474 - acc: 0.9918 - val_lo
Epoch 19/40
1093/1093 [==============================] - 14s 13ms/step - loss: 0.1094 - acc: 0.9826 - val_lo
Epoch 20/40
1093/1093 [==============================] - 14s 13ms/step - loss: 0.0179 - acc: 0.9945 - val_lo
Epoch 21/40
1093/1093 [==============================] - 14s 13ms/step - loss: 0.0966 - acc: 0.9899 - val_lo
Epoch 22/40
1093/1093 [==============================] - 14s 13ms/step - loss: 0.0773 - acc: 0.9899 - val_lo
Epoch 23/40
1093/1093 [==============================] - 14s 13ms/step - loss: 0.0670 - acc: 0.9863 - val_lo
Epoch 24/40
1093/1093 [==============================] - 14s 13ms/step - loss: 0.0985 - acc: 0.9835 - val_lo
Epoch 25/40
1093/1093 [==============================] - 14s 13ms/step - loss: 0.0826 - acc: 0.9909 - val_lo
Epoch 26/40
1093/1093 [==============================] - 14s 13ms/step - loss: 0.1627 - acc: 0.9771 - val_lo
Epoch 27/40
1093/1093 [==============================] - 14s 13ms/step - loss: 0.0583 - acc: 0.9899 - val_lo
Epoch 28/40
1093/1093 [==============================] - 14s 13ms/step - loss: 0.0402 - acc: 0.9936 - val_lo
Epoch 29/40
1093/1093 [==============================] - 14s 13ms/step - loss: 0.1801 - acc: 0.9826 - val_lo
Epoch 30/40
1093/1093 [==============================] - 14s 13ms/step - loss: 0.0346 - acc: 0.9936 - val_lo
Epoch 31/40
1093/1093 [==============================] - 14s 13ms/step - loss: 0.0768 - acc: 0.9899 - val_lo
Epoch 32/40
1093/1093 [==============================] - 14s 13ms/step - loss: 0.1019 - acc: 0.9854 - val_lo
Epoch 33/40
1093/1093 [==============================] - 14s 13ms/step - loss: 0.0348 - acc: 0.9927 - val_lo
Epoch 34/40
1093/1093 [==============================] - 14s 13ms/step - loss: 0.1269 - acc: 0.9817 - val_lo
Epoch 35/40
1093/1093 [==============================] - 14s 13ms/step - loss: 0.0643 - acc: 0.9909 - val_lo
Epoch 36/40
1093/1093 [==============================] - 14s 13ms/step - loss: 0.1223 - acc: 0.9881 - val_lo
Epoch 37/40
1093/1093 [==============================] - 14s 13ms/step - loss: 0.0773 - acc: 0.9881 - val_lo
Epoch 38/40
1093/1093 [==============================] - 14s 13ms/step - loss: 0.0318 - acc: 0.9954 - val_lo
Epoch 39/40
1093/1093 [==============================] - 14s 13ms/step - loss: 0.0966 - acc: 0.9872 - val_lo
Epoch 40/40
1093/1093 [==============================] - 14s 13ms/step - loss: 0.0207 - acc: 0.9954 - val_lo
```

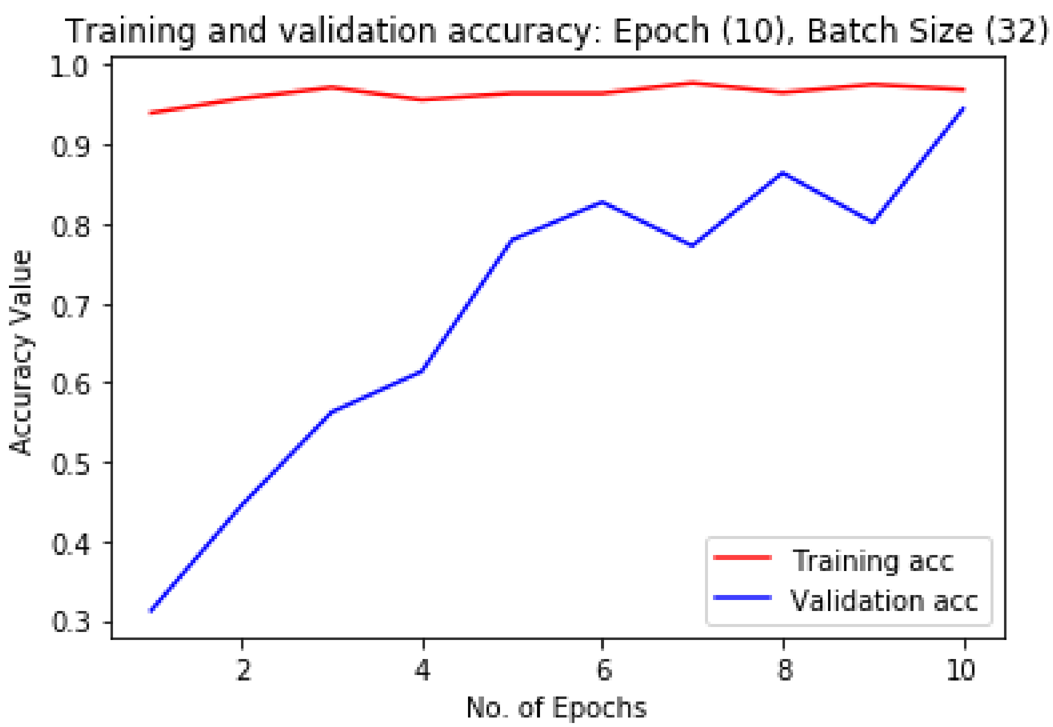
```
Test accuracy: 94.8529%
Training Time: [170.2390956878662, 142.29511785507202, 333.946396112442, 284.02110266685486, 666
```

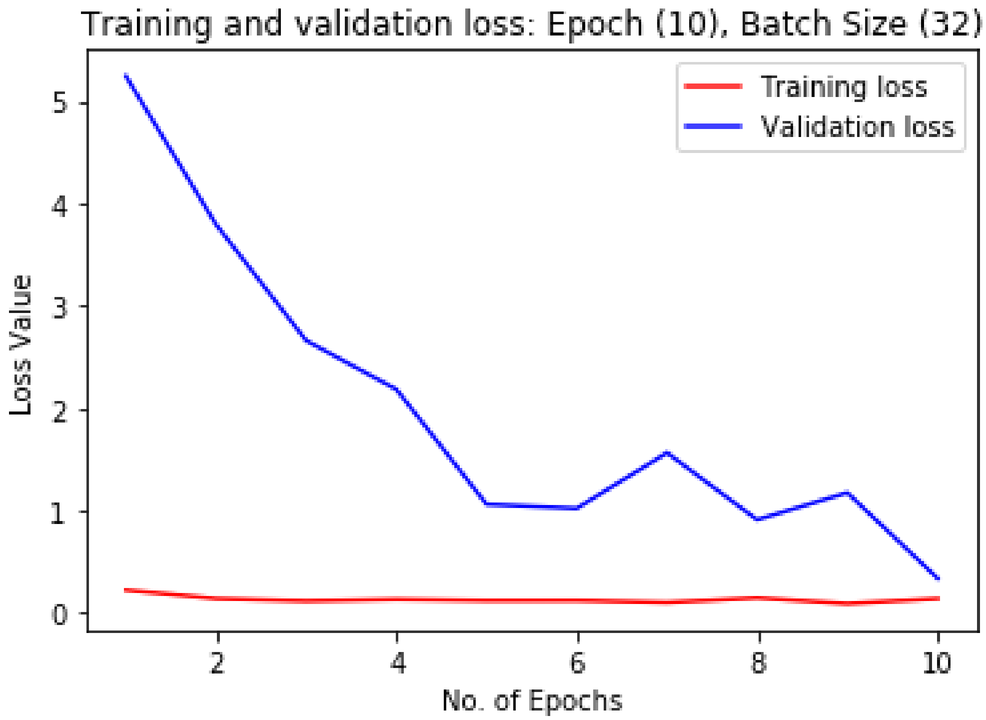
```
In [14]: ## Calling graph function to create graphs
         create_graph(epochs_val, batch_val, acc, val_acc, loss, val_loss, test_accu, time_taken
```



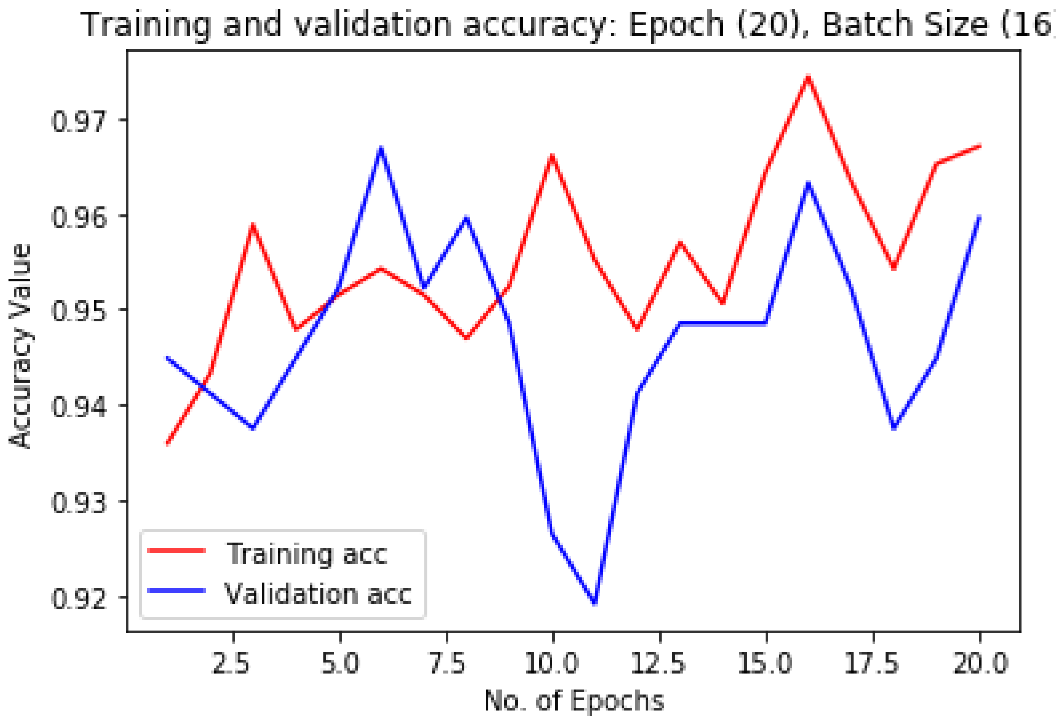


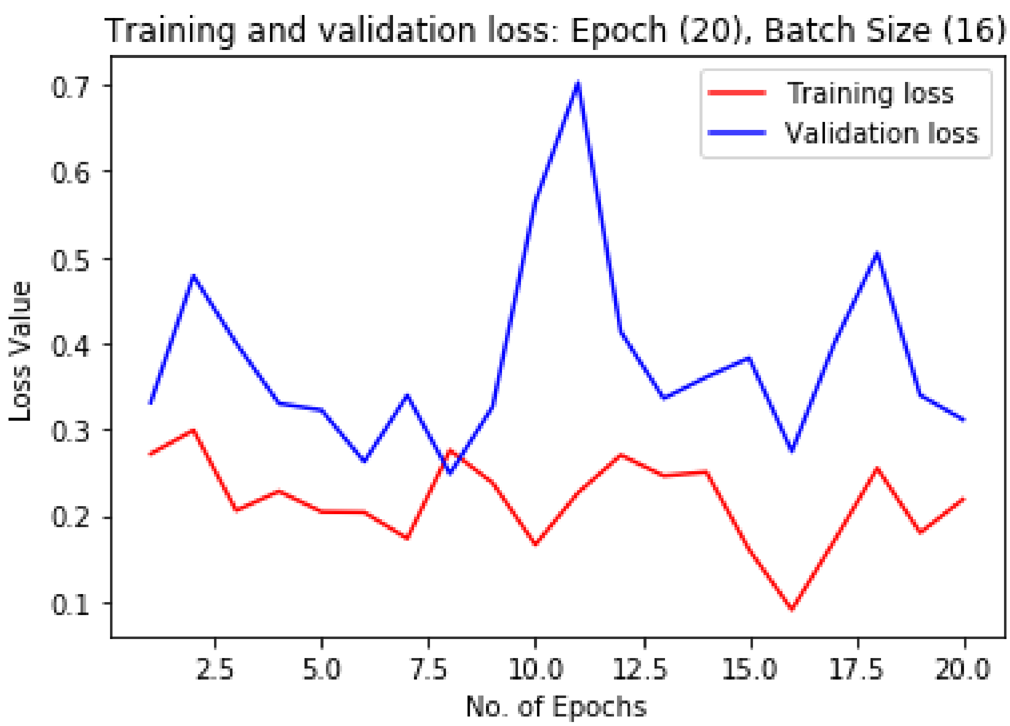

```
Test Accuracy: 26.8382%
Training Time: 170.2390956878662 sec
```



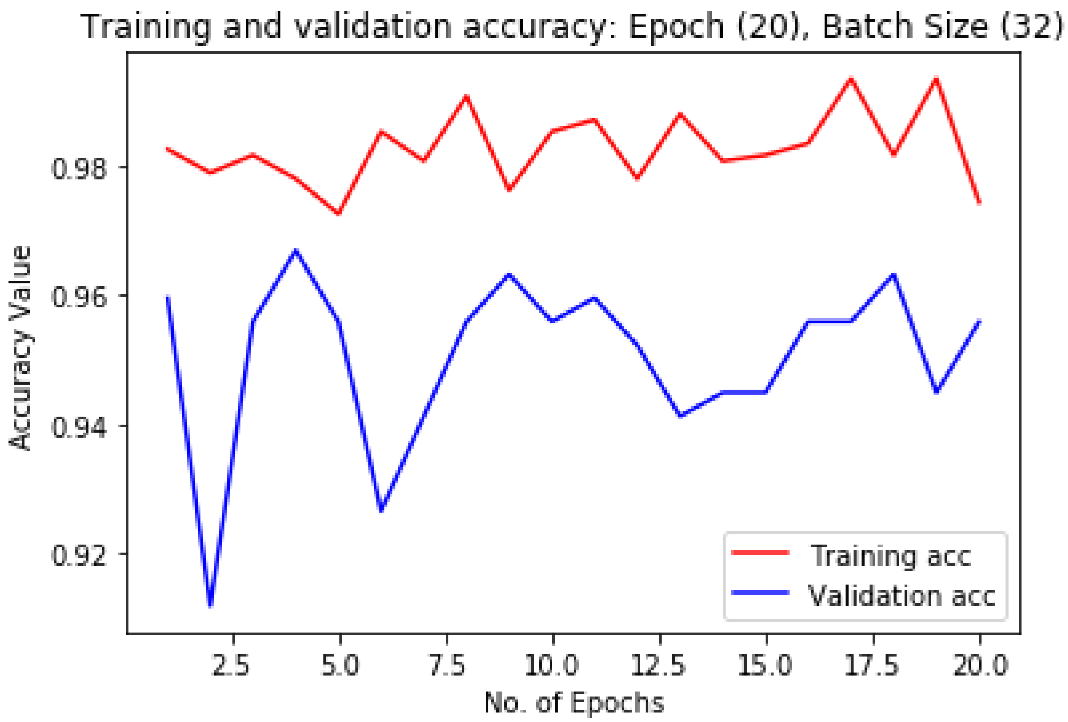


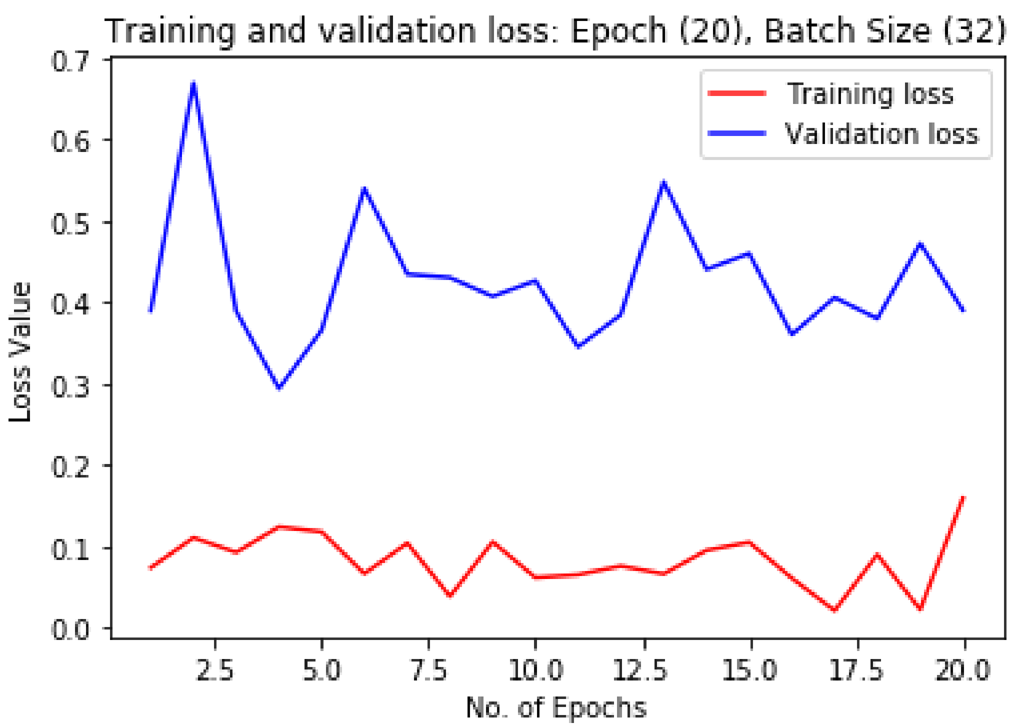

```
Test Accuracy: 94.4853%
Training Time: 142.29511785507202 sec
```


Training and validation accuracy: Epoch (20), Batch Size (16)
Accuracy Value
0.97
0.96
0.95
0.94
0.93
0.92
2.5
5.0
7.5
10.0
12.5
15.0
17.5
20.0
No. of Epochs
Training acc
Validation acc

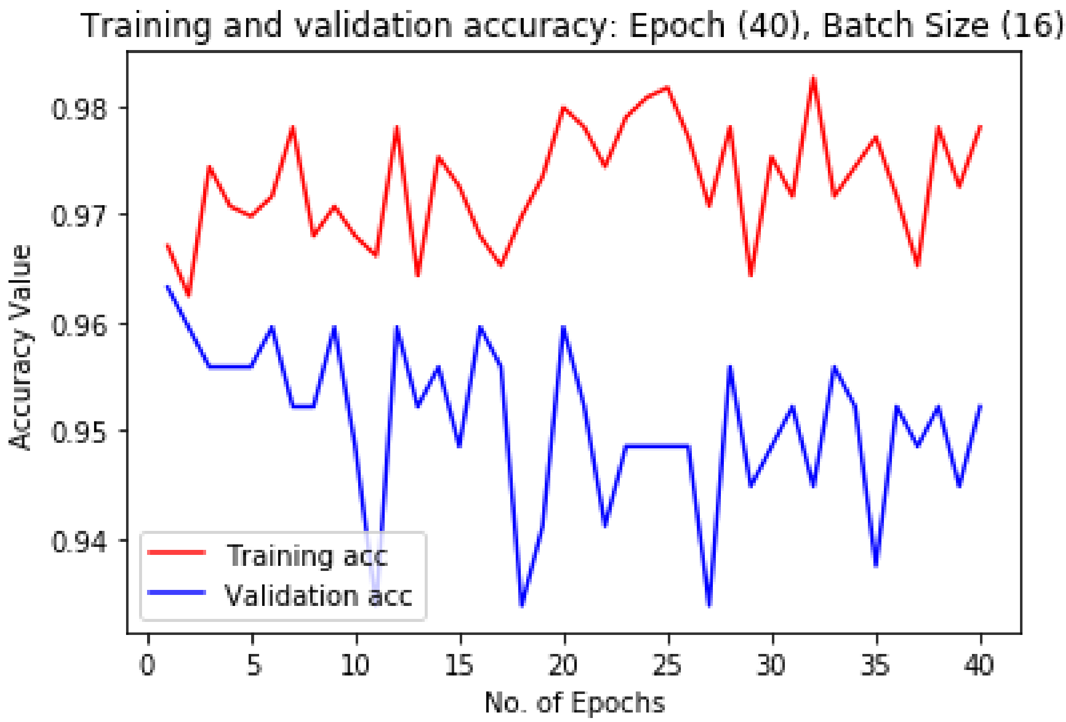

Training and validation loss: Epoch (20), Batch Size (16)
Loss Value
0.7
0.6
0.5
0.4
0.3
0.2
0.1
2.5
5.0
7.5
10.0
12.5
15.0
17.5
20.0
No. of Epochs
Training loss
Validation loss

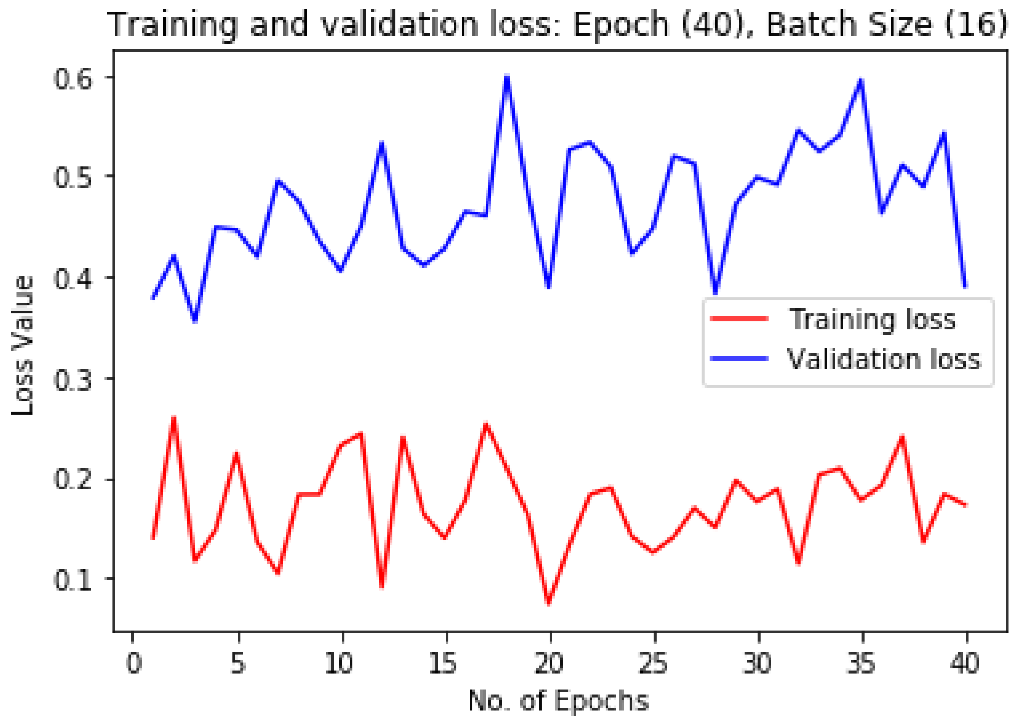
```
Test Accuracy: 95.9559%
Training Time: 333.946396112442 sec
```



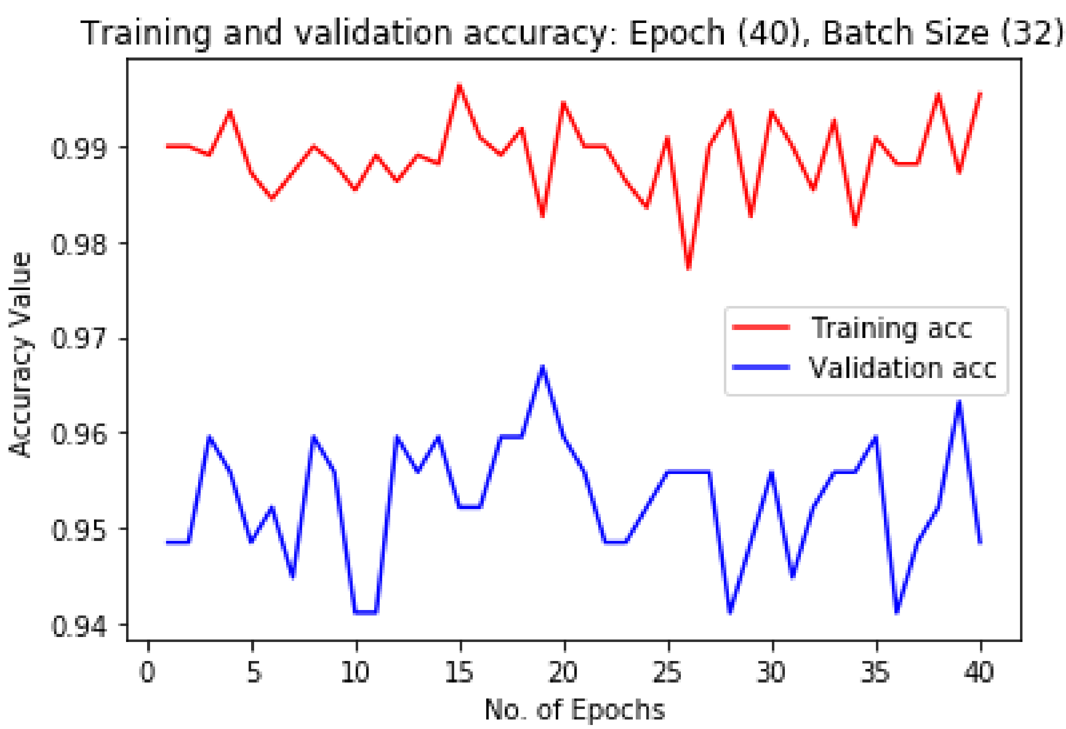


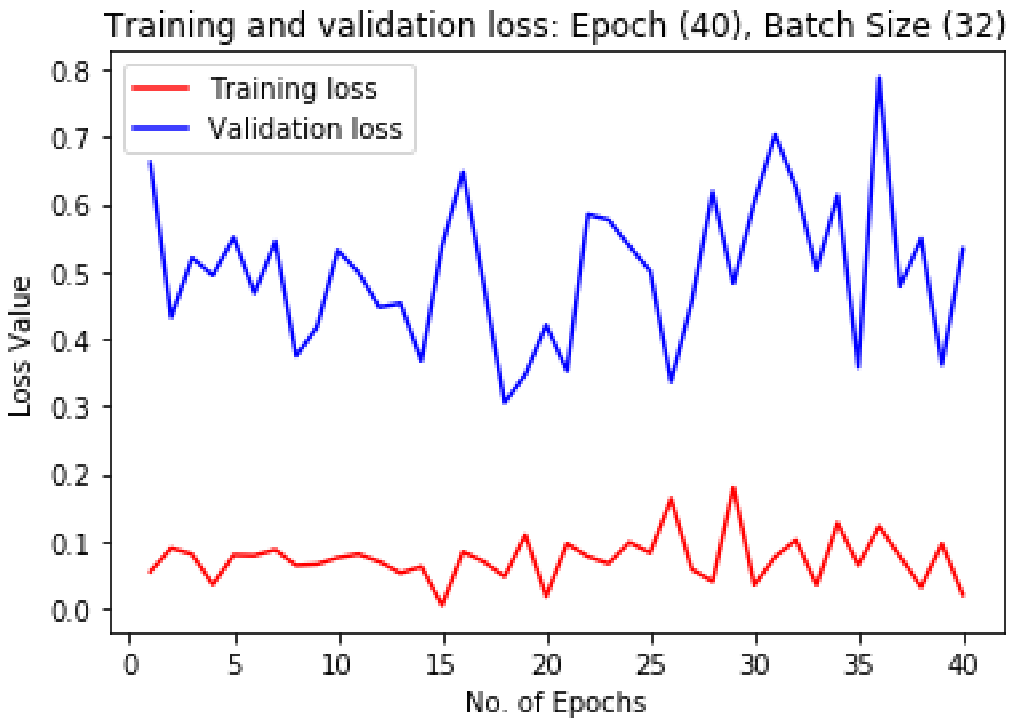
```
Test Accuracy: 95.5882%
Training Time: 284.02110266685486 sec
```






```
Test Accuracy: 95.2206%
Training Time: 666.5932734012604 sec
```






```
Test Accuracy: 94.8529%
Training Time: 566.6708428859711 sec
```

1.3.2 Creating VGG16 model using three modularized functions

Created New Function

1. VGG16_mod => Create VGG16 model

Used Existing Function

2. train_model => Train the created model
3. create_graph => Plot Accuracy and Loss functions

```
In [15]: ## Creating a ResNet50 Model after clearing session
         def VGG16_mod():

             ## To clear memory from the model. Remove the comment and run.
             backend.clear_session()

             ## To create a new model
```

```
        VGG16_model = VGG16(weights='imagenet',include_top=False,input_shape=(224,224,3),
        x = VGG16_model.output
        x = Dense(512, activation='relu')(x)
        x = Dropout(0.5)(x)
        x = Dense(512, activation='relu')(x)
        x = Dropout(0.5)(x)
        output = Dense(11, activation='softmax', name='custom_output')(x) ## Check why it c
        VGG16_T_model = Model(inputs=VGG16_model.input, outputs = output)

        ## Suppressing retraining of already trained model
        for layer in VGG16_model.layers:
            layer.trainable = False

        ## Compiling the model
        VGG16_T_model.compile(optimizer='rmsprop', loss='categorical_crossentropy', metrics
        ## Checking how does the complie process work with Adm optimizer
        ## ResNet50_T_model.compile(Adam(lr=0.001), loss='categorical_crossentropy', metric
        return VGG16_T_model
```

In [16]:
```
### Create model and train

## Submitting with below
epochs_val = [10,20,40]
batch_val = [16,32]

## Test with below
#epochs_val = [10]
#batch_val = [32]

## Create and train model
model = VGG16_mod()
(acc, val_acc, loss, val_loss, test_accu, time_taken) = train_model(model, epochs_val,
```

```
Downloading data from https://github.com/fchollet/deep-learning-models/releases/download/v0.1/vg
58892288/58889256 [==============================] - 1s 0us/step
10   16
Train on 1093 samples, validate on 272 samples
Epoch 1/10
1093/1093 [==============================] - 29s 26ms/step - loss: 2.3063 - acc: 0.1766 - val_lo
Epoch 2/10
1093/1093 [==============================] - 26s 23ms/step - loss: 1.9706 - acc: 0.2653 - val_lo
Epoch 3/10
1093/1093 [==============================] - 26s 24ms/step - loss: 1.7293 - acc: 0.3916 - val_lo
Epoch 4/10
1093/1093 [==============================] - 26s 24ms/step - loss: 1.5211 - acc: 0.4639 - val_lo
Epoch 5/10
1093/1093 [==============================] - 26s 24ms/step - loss: 1.3636 - acc: 0.5178 - val_lo
Epoch 6/10
```

```
1093/1093 [==============================] - 26s 23ms/step - loss: 1.2373 - acc: 0.5810 - val_lo
Epoch 7/10
1093/1093 [==============================] - 26s 24ms/step - loss: 1.2046 - acc: 0.5819 - val_lo
Epoch 8/10
1093/1093 [==============================] - 26s 24ms/step - loss: 1.0887 - acc: 0.6176 - val_lo
Epoch 9/10
1093/1093 [==============================] - 26s 24ms/step - loss: 1.0226 - acc: 0.6514 - val_lo
Epoch 10/10
1093/1093 [==============================] - 26s 23ms/step - loss: 0.9412 - acc: 0.6807 - val_lo
Test accuracy: 65.8088%
Training Time: [261.41396284103394] sec
10   32
Train on 1093 samples, validate on 272 samples
Epoch 1/10
1093/1093 [==============================] - 26s 24ms/step - loss: 0.8391 - acc: 0.7008 - val_lo
Epoch 2/10
1093/1093 [==============================] - 23s 21ms/step - loss: 0.7696 - acc: 0.7301 - val_lo
Epoch 3/10
1093/1093 [==============================] - 23s 21ms/step - loss: 0.7865 - acc: 0.7466 - val_lo
Epoch 4/10
1093/1093 [==============================] - 23s 21ms/step - loss: 0.7244 - acc: 0.7438 - val_lo
Epoch 5/10
1093/1093 [==============================] - 23s 21ms/step - loss: 0.7557 - acc: 0.7392 - val_lo
Epoch 6/10
1093/1093 [==============================] - 23s 21ms/step - loss: 0.7064 - acc: 0.7575 - val_lo
Epoch 7/10
1093/1093 [==============================] - 23s 21ms/step - loss: 0.6712 - acc: 0.7786 - val_lo
Epoch 8/10
1093/1093 [==============================] - 23s 21ms/step - loss: 0.6376 - acc: 0.7685 - val_lo
Epoch 9/10
1093/1093 [==============================] - 23s 21ms/step - loss: 0.5987 - acc: 0.7960 - val_lo
Epoch 10/10
1093/1093 [==============================] - 23s 21ms/step - loss: 0.5938 - acc: 0.7887 - val_lo
Test accuracy: 74.2647%
Training Time: [261.41396284103394, 230.02988648414612] sec
20   16
Train on 1093 samples, validate on 272 samples
Epoch 1/20
1093/1093 [==============================] - 26s 24ms/step - loss: 0.6138 - acc: 0.7722 - val_lo
Epoch 2/20
1093/1093 [==============================] - 26s 24ms/step - loss: 0.6065 - acc: 0.7969 - val_lo
Epoch 3/20
1093/1093 [==============================] - 26s 24ms/step - loss: 0.5858 - acc: 0.8033 - val_lo
Epoch 4/20
1093/1093 [==============================] - 26s 24ms/step - loss: 0.5936 - acc: 0.7996 - val_lo
Epoch 5/20
1093/1093 [==============================] - 26s 24ms/step - loss: 0.5622 - acc: 0.8079 - val_lo
Epoch 6/20
```

```
1093/1093 [==============================] - 26s 24ms/step - loss: 0.5263 - acc: 0.8152 - val_lo
Epoch 7/20
1093/1093 [==============================] - 26s 24ms/step - loss: 0.5381 - acc: 0.8024 - val_lo
Epoch 8/20
1093/1093 [==============================] - 26s 24ms/step - loss: 0.5017 - acc: 0.8234 - val_lo
Epoch 9/20
1093/1093 [==============================] - 26s 24ms/step - loss: 0.4843 - acc: 0.8298 - val_lo
Epoch 10/20
1093/1093 [==============================] - 26s 23ms/step - loss: 0.4609 - acc: 0.8381 - val_lo
Epoch 11/20
1093/1093 [==============================] - 26s 24ms/step - loss: 0.4985 - acc: 0.8326 - val_lo
Epoch 12/20
1093/1093 [==============================] - 26s 24ms/step - loss: 0.4520 - acc: 0.8371 - val_lo
Epoch 13/20
1093/1093 [==============================] - 26s 24ms/step - loss: 0.4245 - acc: 0.8536 - val_lo
Epoch 14/20
1093/1093 [==============================] - 26s 24ms/step - loss: 0.4584 - acc: 0.8554 - val_lo
Epoch 15/20
1093/1093 [==============================] - 26s 24ms/step - loss: 0.4081 - acc: 0.8554 - val_lo
Epoch 16/20
1093/1093 [==============================] - 26s 24ms/step - loss: 0.4004 - acc: 0.8683 - val_lo
Epoch 17/20
1093/1093 [==============================] - 26s 24ms/step - loss: 0.4092 - acc: 0.8710 - val_lo
Epoch 18/20
1093/1093 [==============================] - 26s 24ms/step - loss: 0.4158 - acc: 0.8527 - val_lo
Epoch 19/20
1093/1093 [==============================] - 26s 24ms/step - loss: 0.3479 - acc: 0.8792 - val_lo
Epoch 20/20
1093/1093 [==============================] - 26s 23ms/step - loss: 0.4269 - acc: 0.8564 - val_lo
Test accuracy: 81.2500%
Training Time: [261.41396284103394, 230.02988648414612, 516.8229987621307] sec
20   32
Train on 1093 samples, validate on 272 samples
Epoch 1/20
1093/1093 [==============================] - 23s 21ms/step - loss: 0.2813 - acc: 0.9030 - val_lo
Epoch 2/20
1093/1093 [==============================] - 23s 21ms/step - loss: 0.3098 - acc: 0.8911 - val_lo
Epoch 3/20
1093/1093 [==============================] - 23s 21ms/step - loss: 0.2686 - acc: 0.9076 - val_lo
Epoch 4/20
1093/1093 [==============================] - 23s 21ms/step - loss: 0.3187 - acc: 0.8875 - val_lo
Epoch 5/20
1093/1093 [==============================] - 23s 21ms/step - loss: 0.2798 - acc: 0.8994 - val_lo
Epoch 6/20
1093/1093 [==============================] - 23s 21ms/step - loss: 0.2418 - acc: 0.9186 - val_lo
Epoch 7/20
1093/1093 [==============================] - 23s 21ms/step - loss: 0.2585 - acc: 0.9122 - val_lo
Epoch 8/20
```

```
1093/1093 [==============================] - 23s 21ms/step - loss: 0.2792 - acc: 0.9103 - val_lo
Epoch 9/20
1093/1093 [==============================] - 23s 21ms/step - loss: 0.2382 - acc: 0.9222 - val_lo
Epoch 10/20
1093/1093 [==============================] - 23s 21ms/step - loss: 0.2317 - acc: 0.9222 - val_lo
Epoch 11/20
1093/1093 [==============================] - 23s 21ms/step - loss: 0.2732 - acc: 0.9058 - val_lo
Epoch 12/20
1093/1093 [==============================] - 23s 21ms/step - loss: 0.2198 - acc: 0.9231 - val_lo
Epoch 13/20
1093/1093 [==============================] - 23s 21ms/step - loss: 0.2215 - acc: 0.9222 - val_lo
Epoch 14/20
1093/1093 [==============================] - 23s 21ms/step - loss: 0.2444 - acc: 0.9149 - val_lo
Epoch 15/20
1093/1093 [==============================] - 23s 21ms/step - loss: 0.2057 - acc: 0.9360 - val_lo
Epoch 16/20
1093/1093 [==============================] - 23s 21ms/step - loss: 0.2331 - acc: 0.9231 - val_lo
Epoch 17/20
1093/1093 [==============================] - 23s 21ms/step - loss: 0.2108 - acc: 0.9268 - val_lo
Epoch 18/20
1093/1093 [==============================] - 23s 21ms/step - loss: 0.1867 - acc: 0.9369 - val_lo
Epoch 19/20
1093/1093 [==============================] - 23s 21ms/step - loss: 0.2096 - acc: 0.9231 - val_lo
Epoch 20/20
1093/1093 [==============================] - 23s 21ms/step - loss: 0.1985 - acc: 0.9341 - val_lo
Test accuracy: 83.0882%
Training Time: [261.41396284103394, 230.02988648414612, 516.8229987621307, 454.4341125488281] se
40   16
Train on 1093 samples, validate on 272 samples
Epoch 1/40
1093/1093 [==============================] - 26s 24ms/step - loss: 0.2867 - acc: 0.9021 - val_lo
Epoch 2/40
1093/1093 [==============================] - 26s 24ms/step - loss: 0.3169 - acc: 0.9030 - val_lo
Epoch 3/40
1093/1093 [==============================] - 26s 24ms/step - loss: 0.2701 - acc: 0.9103 - val_lo
Epoch 4/40
1093/1093 [==============================] - 26s 24ms/step - loss: 0.2784 - acc: 0.9103 - val_lo
Epoch 5/40
1093/1093 [==============================] - 26s 24ms/step - loss: 0.2616 - acc: 0.9158 - val_lo
Epoch 6/40
1093/1093 [==============================] - 26s 24ms/step - loss: 0.3051 - acc: 0.8994 - val_lo
Epoch 7/40
1093/1093 [==============================] - 26s 24ms/step - loss: 0.2609 - acc: 0.9259 - val_lo
Epoch 8/40
1093/1093 [==============================] - 26s 24ms/step - loss: 0.3301 - acc: 0.8975 - val_lo
Epoch 9/40
1093/1093 [==============================] - 26s 24ms/step - loss: 0.2622 - acc: 0.9149 - val_lo
Epoch 10/40
```

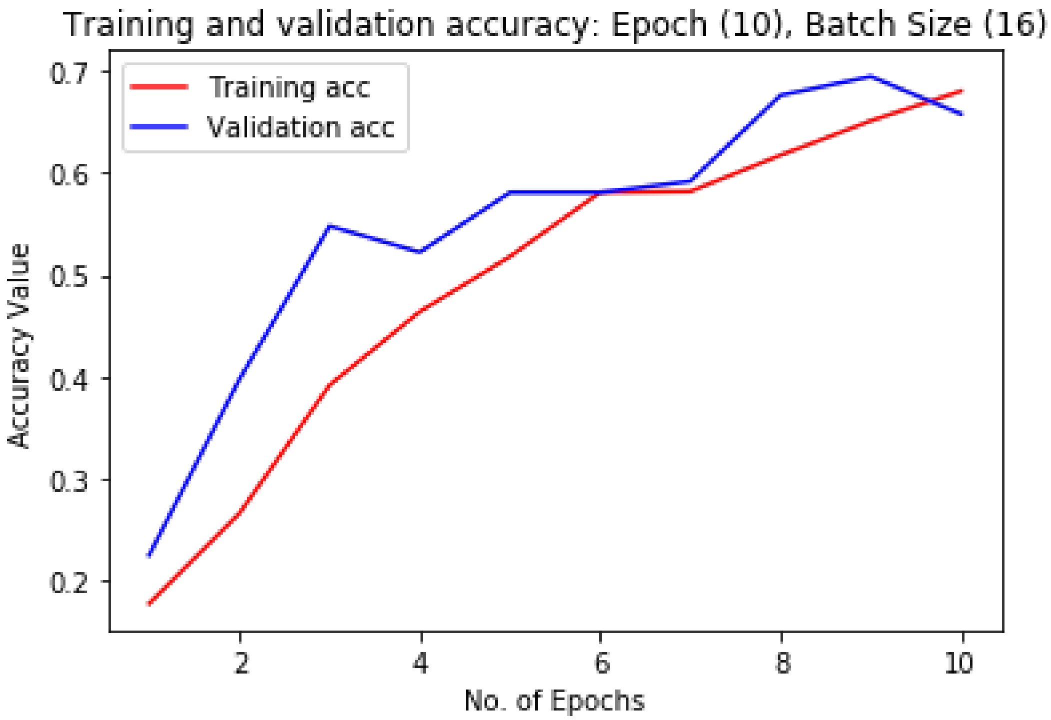
```
1093/1093 [==============================] - 26s 24ms/step - loss: 0.2177 - acc: 0.9204 - val_lo
Epoch 11/40
1093/1093 [==============================] - 26s 24ms/step - loss: 0.2457 - acc: 0.9131 - val_lo
Epoch 12/40
1093/1093 [==============================] - 26s 23ms/step - loss: 0.2187 - acc: 0.9305 - val_lo
Epoch 13/40
1093/1093 [==============================] - 26s 24ms/step - loss: 0.2488 - acc: 0.9195 - val_lo
Epoch 14/40
1093/1093 [==============================] - 26s 24ms/step - loss: 0.2267 - acc: 0.9241 - val_lo
Epoch 15/40
1093/1093 [==============================] - 26s 24ms/step - loss: 0.2813 - acc: 0.9058 - val_lo
Epoch 16/40
1093/1093 [==============================] - 26s 24ms/step - loss: 0.2325 - acc: 0.9149 - val_lo
Epoch 17/40
1093/1093 [==============================] - 26s 24ms/step - loss: 0.2343 - acc: 0.9222 - val_lo
Epoch 18/40
1093/1093 [==============================] - 26s 24ms/step - loss: 0.2616 - acc: 0.9195 - val_lo
Epoch 19/40
1093/1093 [==============================] - 26s 24ms/step - loss: 0.2117 - acc: 0.9314 - val_lo
Epoch 20/40
1093/1093 [==============================] - 26s 24ms/step - loss: 0.2509 - acc: 0.9231 - val_lo
Epoch 21/40
1093/1093 [==============================] - 26s 24ms/step - loss: 0.2412 - acc: 0.9305 - val_lo
Epoch 22/40
1093/1093 [==============================] - 26s 24ms/step - loss: 0.2145 - acc: 0.9360 - val_lo
Epoch 23/40
1093/1093 [==============================] - 26s 24ms/step - loss: 0.2132 - acc: 0.9277 - val_lo
Epoch 24/40
1093/1093 [==============================] - 26s 23ms/step - loss: 0.2293 - acc: 0.9350 - val_lo
Epoch 25/40
1093/1093 [==============================] - 26s 24ms/step - loss: 0.2188 - acc: 0.9177 - val_lo
Epoch 26/40
1093/1093 [==============================] - 26s 24ms/step - loss: 0.2036 - acc: 0.9360 - val_lo
Epoch 27/40
1093/1093 [==============================] - 26s 24ms/step - loss: 0.2568 - acc: 0.9167 - val_lo
Epoch 28/40
1093/1093 [==============================] - 26s 24ms/step - loss: 0.2001 - acc: 0.9350 - val_lo
Epoch 29/40
1093/1093 [==============================] - 26s 24ms/step - loss: 0.2029 - acc: 0.9332 - val_lo
Epoch 30/40
1093/1093 [==============================] - 26s 24ms/step - loss: 0.2277 - acc: 0.9314 - val_lo
Epoch 31/40
1093/1093 [==============================] - 26s 24ms/step - loss: 0.2312 - acc: 0.9277 - val_lo
Epoch 32/40
1093/1093 [==============================] - 26s 24ms/step - loss: 0.2423 - acc: 0.9222 - val_lo
Epoch 33/40
1093/1093 [==============================] - 26s 24ms/step - loss: 0.2505 - acc: 0.9213 - val_lo
Epoch 34/40
```

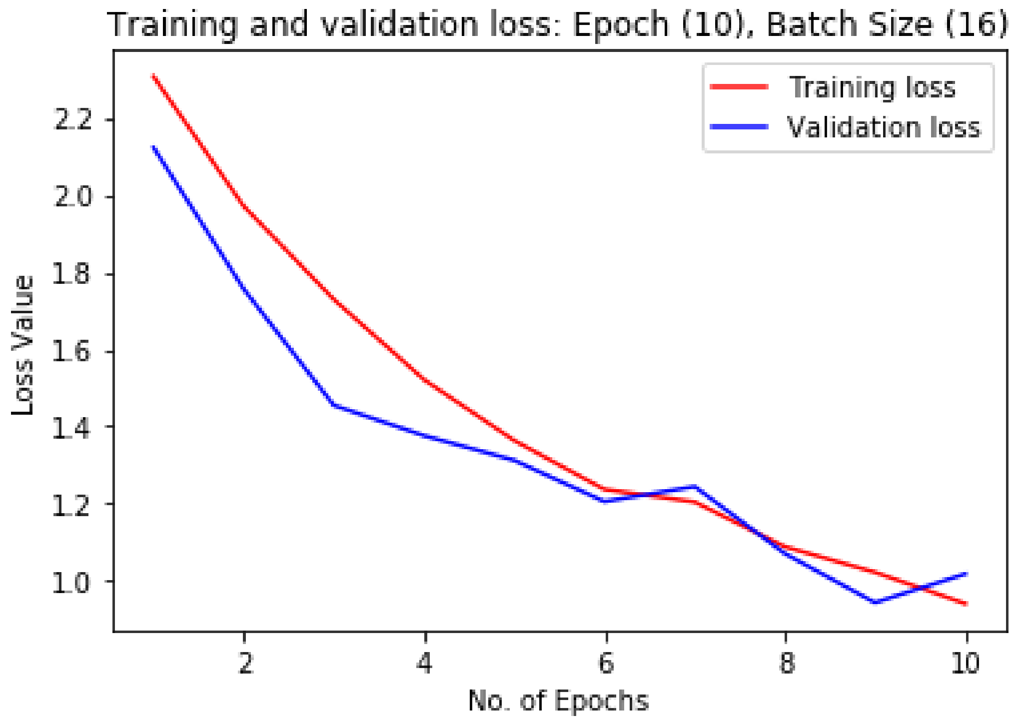
```
1093/1093 [==============================] - 26s 24ms/step - loss: 0.2390 - acc: 0.9259 - val_lo
Epoch 35/40
1093/1093 [==============================] - 26s 24ms/step - loss: 0.1544 - acc: 0.9478 - val_lo
Epoch 36/40
1093/1093 [==============================] - 26s 24ms/step - loss: 0.2215 - acc: 0.9341 - val_lo
Epoch 37/40
1093/1093 [==============================] - 26s 24ms/step - loss: 0.2445 - acc: 0.9296 - val_lo
Epoch 38/40
1093/1093 [==============================] - 26s 24ms/step - loss: 0.2355 - acc: 0.9268 - val_lo
Epoch 39/40
1093/1093 [==============================] - 26s 24ms/step - loss: 0.2313 - acc: 0.9231 - val_lo
Epoch 40/40
1093/1093 [==============================] - 26s 24ms/step - loss: 0.2459 - acc: 0.9305 - val_lo
Test accuracy: 85.2941%
Training Time: [261.41396284103394, 230.02988648414612, 516.8229987621307, 454.4341125488281, 10
40   32
Train on 1093 samples, validate on 272 samples
Epoch 1/40
1093/1093 [==============================] - 23s 21ms/step - loss: 0.1553 - acc: 0.9506 - val_lo
Epoch 2/40
1093/1093 [==============================] - 23s 21ms/step - loss: 0.1776 - acc: 0.9488 - val_lo
Epoch 3/40
1093/1093 [==============================] - 23s 21ms/step - loss: 0.1157 - acc: 0.9634 - val_lo
Epoch 4/40
1093/1093 [==============================] - 23s 21ms/step - loss: 0.1190 - acc: 0.9607 - val_lo
Epoch 5/40
1093/1093 [==============================] - 23s 21ms/step - loss: 0.1814 - acc: 0.9588 - val_lo
Epoch 6/40
1093/1093 [==============================] - 23s 21ms/step - loss: 0.1252 - acc: 0.9488 - val_lo
Epoch 7/40
1093/1093 [==============================] - 23s 21ms/step - loss: 0.1198 - acc: 0.9488 - val_lo
Epoch 8/40
1093/1093 [==============================] - 23s 21ms/step - loss: 0.1254 - acc: 0.9561 - val_lo
Epoch 9/40
1093/1093 [==============================] - 23s 21ms/step - loss: 0.1174 - acc: 0.9597 - val_lo
Epoch 10/40
1093/1093 [==============================] - 23s 21ms/step - loss: 0.1371 - acc: 0.9579 - val_lo
Epoch 11/40
1093/1093 [==============================] - 23s 21ms/step - loss: 0.1234 - acc: 0.9616 - val_lo
Epoch 12/40
1093/1093 [==============================] - 23s 21ms/step - loss: 0.1492 - acc: 0.9533 - val_lo
Epoch 13/40
1093/1093 [==============================] - 23s 21ms/step - loss: 0.1648 - acc: 0.9442 - val_lo
Epoch 14/40
1093/1093 [==============================] - 23s 21ms/step - loss: 0.1260 - acc: 0.9597 - val_lo
Epoch 15/40
1093/1093 [==============================] - 23s 21ms/step - loss: 0.1345 - acc: 0.9570 - val_lo
Epoch 16/40
```

```
1093/1093 [==============================] - 23s 21ms/step - loss: 0.1424 - acc: 0.9561 - val_lo
Epoch 17/40
1093/1093 [==============================] - 23s 21ms/step - loss: 0.1052 - acc: 0.9625 - val_lo
Epoch 18/40
1093/1093 [==============================] - 23s 21ms/step - loss: 0.1606 - acc: 0.9579 - val_lo
Epoch 19/40
1093/1093 [==============================] - 23s 21ms/step - loss: 0.1179 - acc: 0.9652 - val_lo
Epoch 20/40
1093/1093 [==============================] - 23s 21ms/step - loss: 0.0959 - acc: 0.9698 - val_lo
Epoch 21/40
1093/1093 [==============================] - 23s 21ms/step - loss: 0.1484 - acc: 0.9561 - val_lo
Epoch 22/40
1093/1093 [==============================] - 23s 21ms/step - loss: 0.1123 - acc: 0.9671 - val_lo
Epoch 23/40
1093/1093 [==============================] - 23s 21ms/step - loss: 0.1263 - acc: 0.9625 - val_lo
Epoch 24/40
1093/1093 [==============================] - 23s 21ms/step - loss: 0.1244 - acc: 0.9661 - val_lo
Epoch 25/40
1093/1093 [==============================] - 23s 21ms/step - loss: 0.1450 - acc: 0.9497 - val_lo
Epoch 26/40
1093/1093 [==============================] - 23s 21ms/step - loss: 0.1327 - acc: 0.9543 - val_lo
Epoch 27/40
1093/1093 [==============================] - 23s 21ms/step - loss: 0.0880 - acc: 0.9707 - val_lo
Epoch 28/40
1093/1093 [==============================] - 23s 21ms/step - loss: 0.1525 - acc: 0.9625 - val_lo
Epoch 29/40
1093/1093 [==============================] - 23s 21ms/step - loss: 0.1363 - acc: 0.9570 - val_lo
Epoch 30/40
1093/1093 [==============================] - 23s 21ms/step - loss: 0.1079 - acc: 0.9661 - val_lo
Epoch 31/40
1093/1093 [==============================] - 23s 21ms/step - loss: 0.1205 - acc: 0.9643 - val_lo
Epoch 32/40
1093/1093 [==============================] - 23s 21ms/step - loss: 0.1215 - acc: 0.9652 - val_lo
Epoch 33/40
1093/1093 [==============================] - 23s 21ms/step - loss: 0.0965 - acc: 0.9698 - val_lo
Epoch 34/40
1093/1093 [==============================] - 23s 21ms/step - loss: 0.0932 - acc: 0.9698 - val_lo
Epoch 35/40
1093/1093 [==============================] - 23s 21ms/step - loss: 0.1260 - acc: 0.9616 - val_lo
Epoch 36/40
1093/1093 [==============================] - 23s 21ms/step - loss: 0.1280 - acc: 0.9634 - val_lo
Epoch 37/40
1093/1093 [==============================] - 23s 21ms/step - loss: 0.1320 - acc: 0.9579 - val_lo
Epoch 38/40
1093/1093 [==============================] - 23s 21ms/step - loss: 0.1387 - acc: 0.9607 - val_lo
Epoch 39/40
1093/1093 [==============================] - 23s 21ms/step - loss: 0.1181 - acc: 0.9634 - val_lo
Epoch 40/40
```

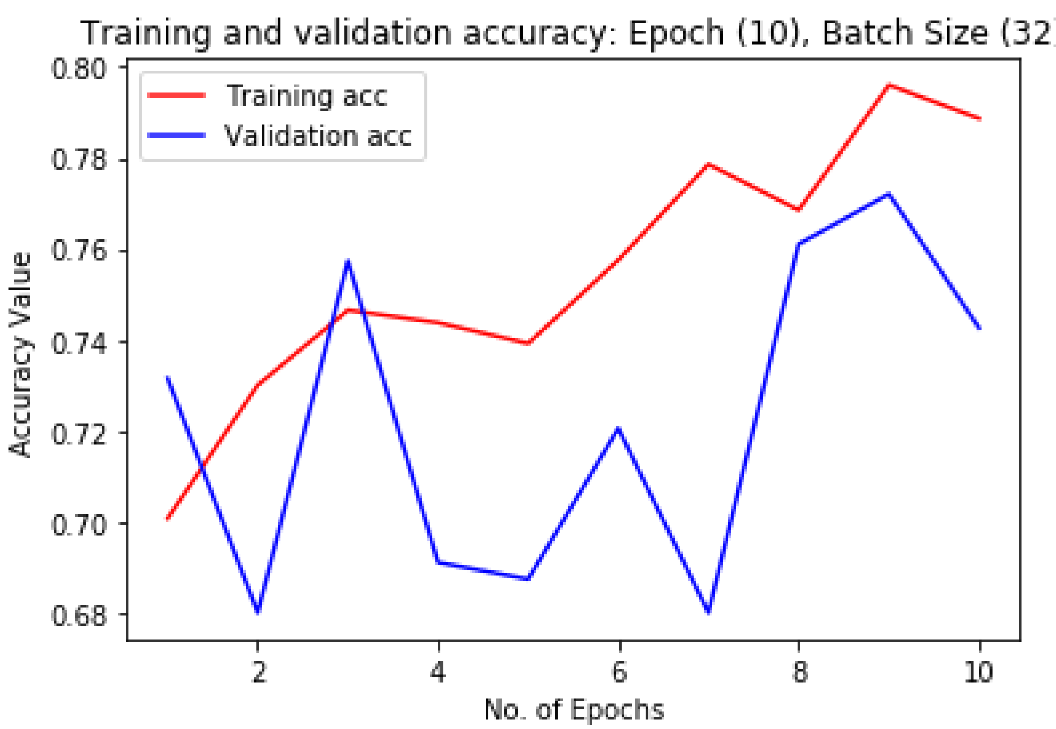
```
1093/1093 [==============================] - 23s 21ms/step - loss: 0.1095 - acc: 0.9707 - val_lo
Test accuracy: 85.6618%
Training Time: [261.41396284103394, 230.02988648414612, 516.8229987621307, 454.4341125488281, 10
```

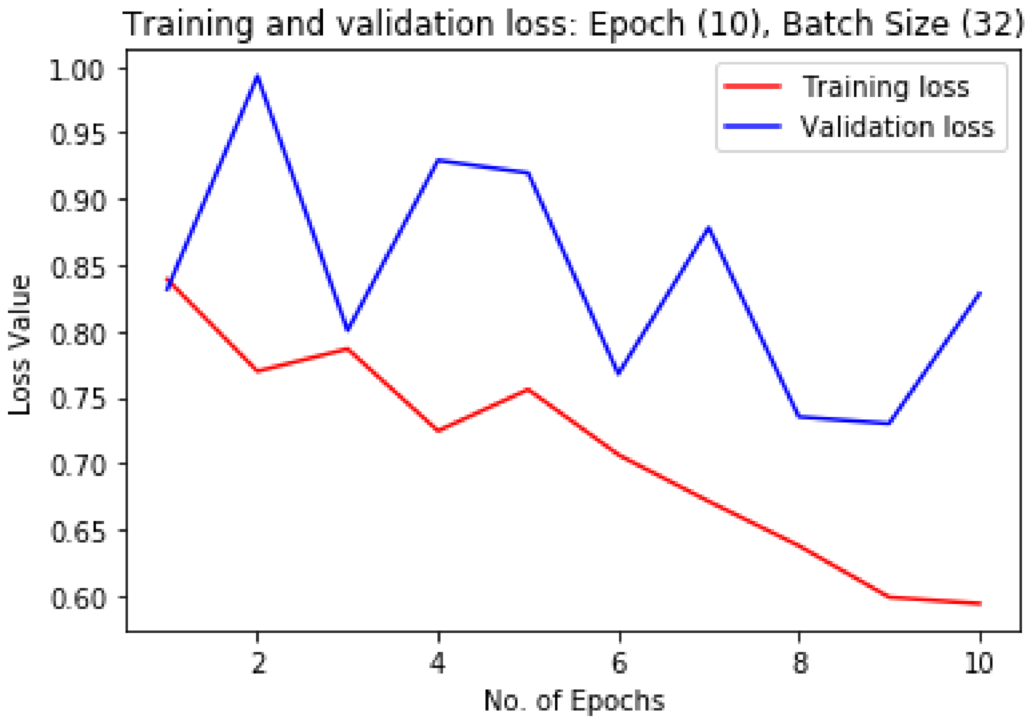
In [17]:
```
## Calling graph function to create graphs
create_graph(epochs_val, batch_val, acc, val_acc, loss, val_loss, test_accu, time_taken
```



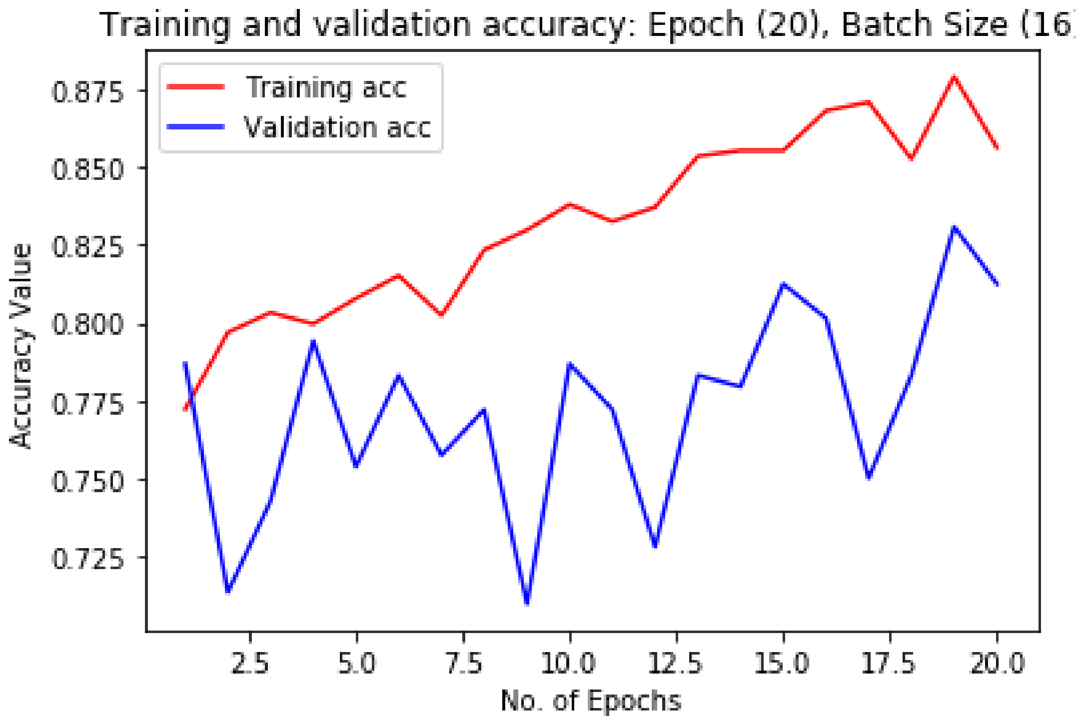


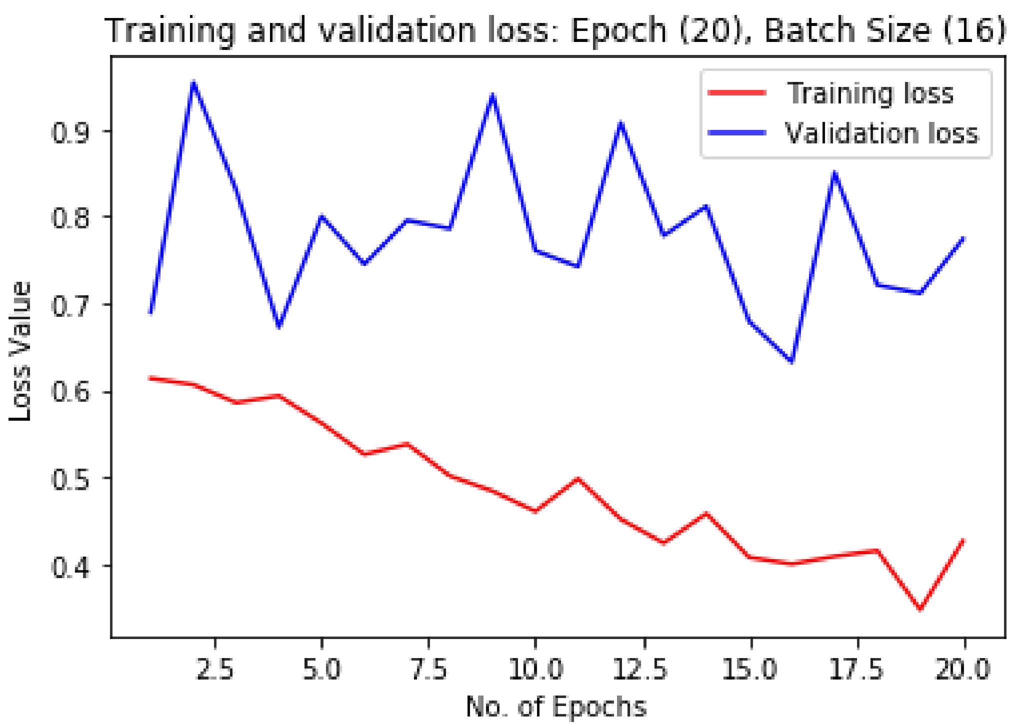

```
Test Accuracy: 65.8088%
Training Time: 261.41396284103394 sec
```



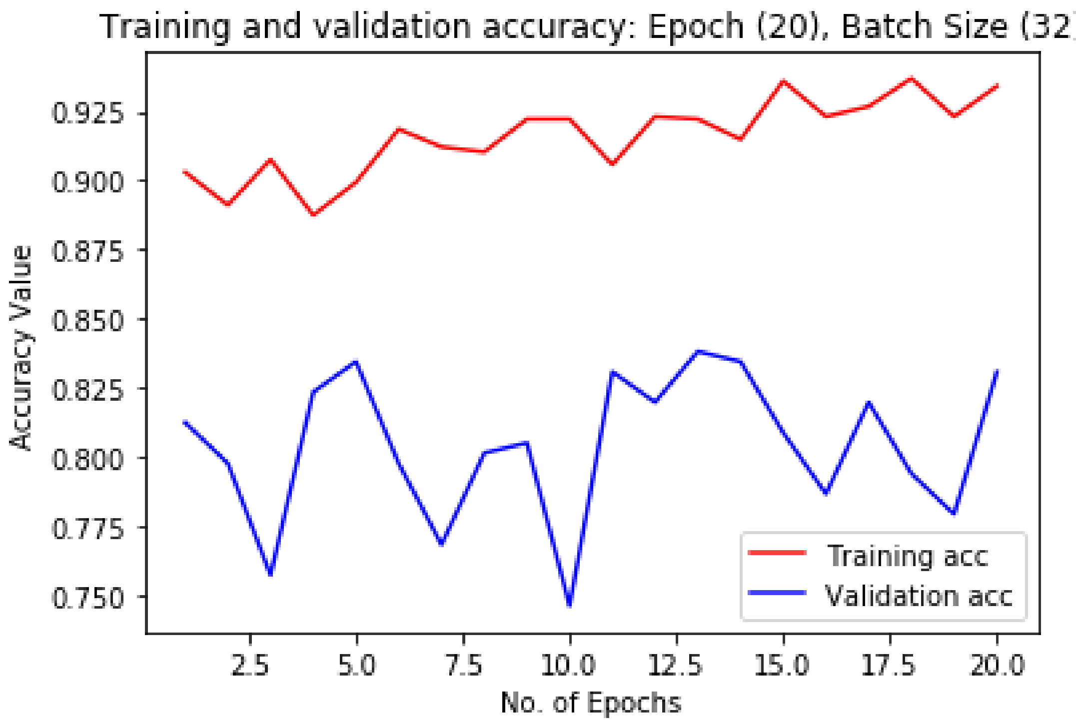


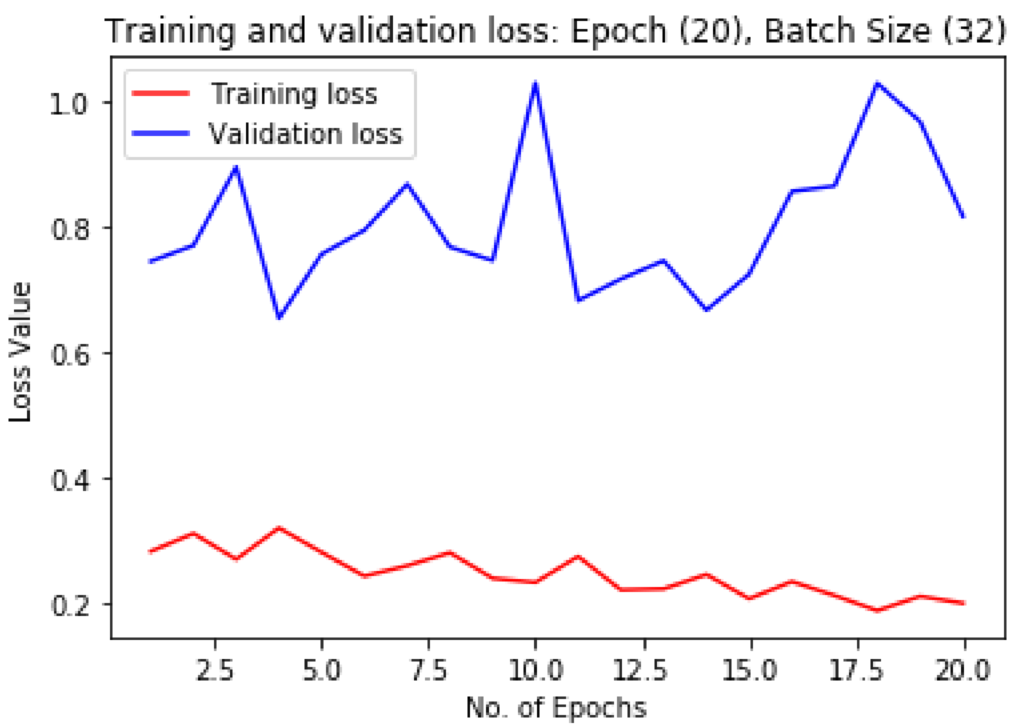

```
Test Accuracy: 74.2647%
Training Time: 230.02988648414612 sec
```


Training and validation accuracy: Epoch (20), Batch Size (16)
Training acc
Validation acc
Accuracy Value
0.875
0.850
0.825
0.800
0.775
0.750
0.725
2.5
5.0
7.5
10.0
12.5
15.0
17.5
20.0
No. of Epochs

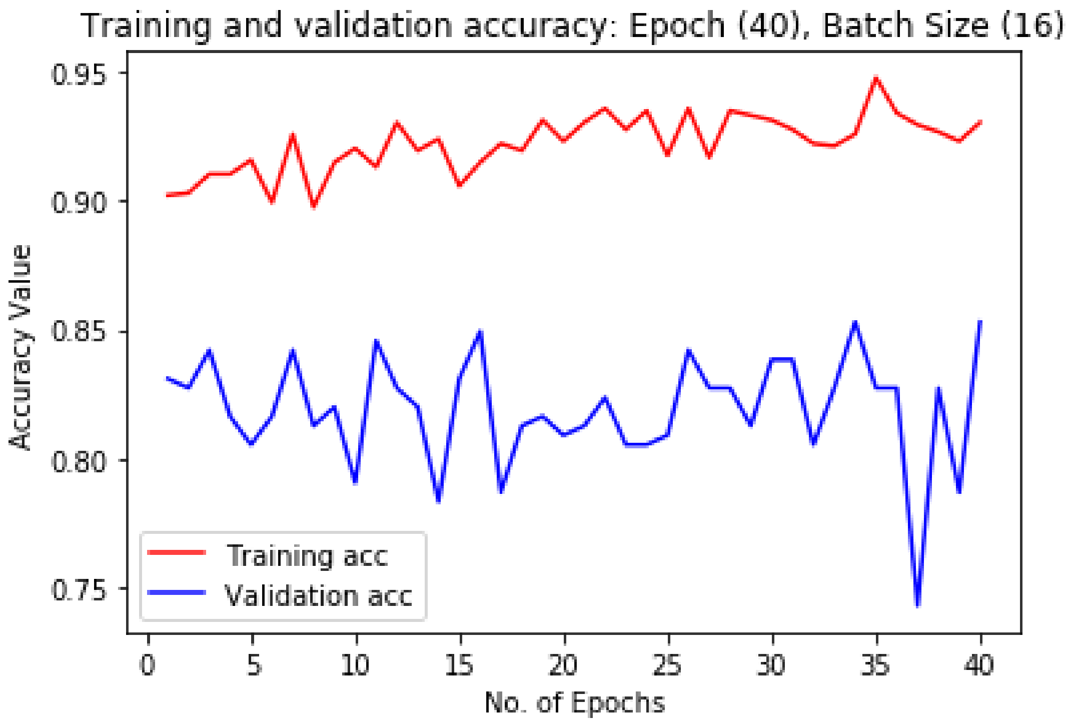

Training and validation loss: Epoch (20), Batch Size (16)
Training loss
Validation loss
Loss Value
0.9
0.8
0.7
0.6
0.5
0.4
2.5
5.0
7.5
10.0
12.5
15.0
17.5
20.0
No. of Epochs

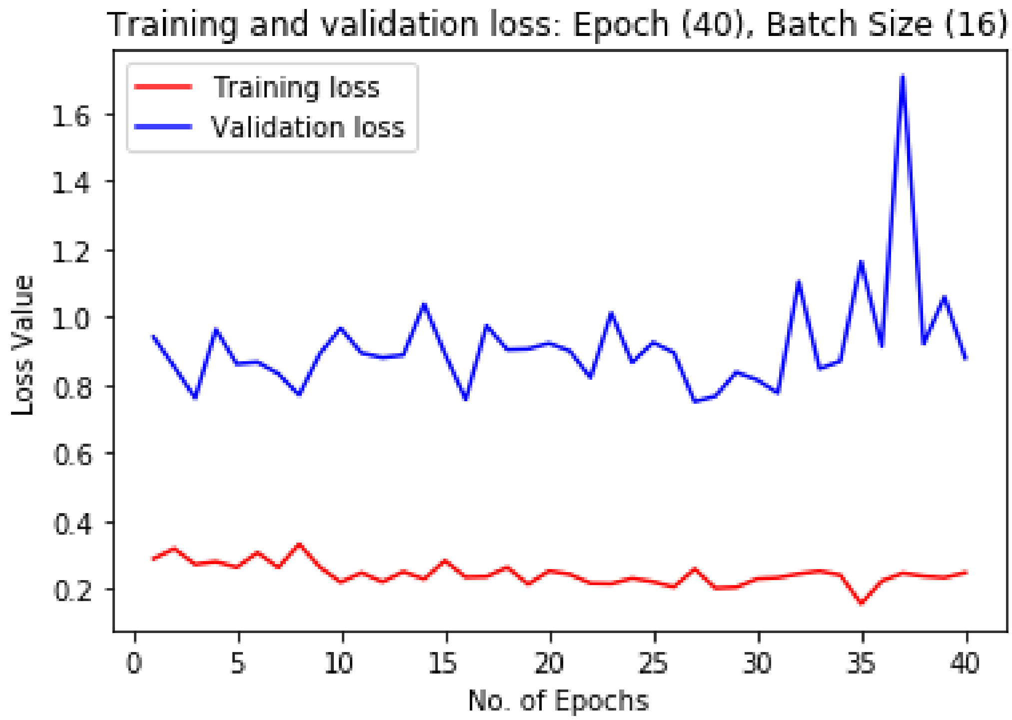
```
Test Accuracy: 81.2500%
Training Time: 516.8229987621307 sec
```



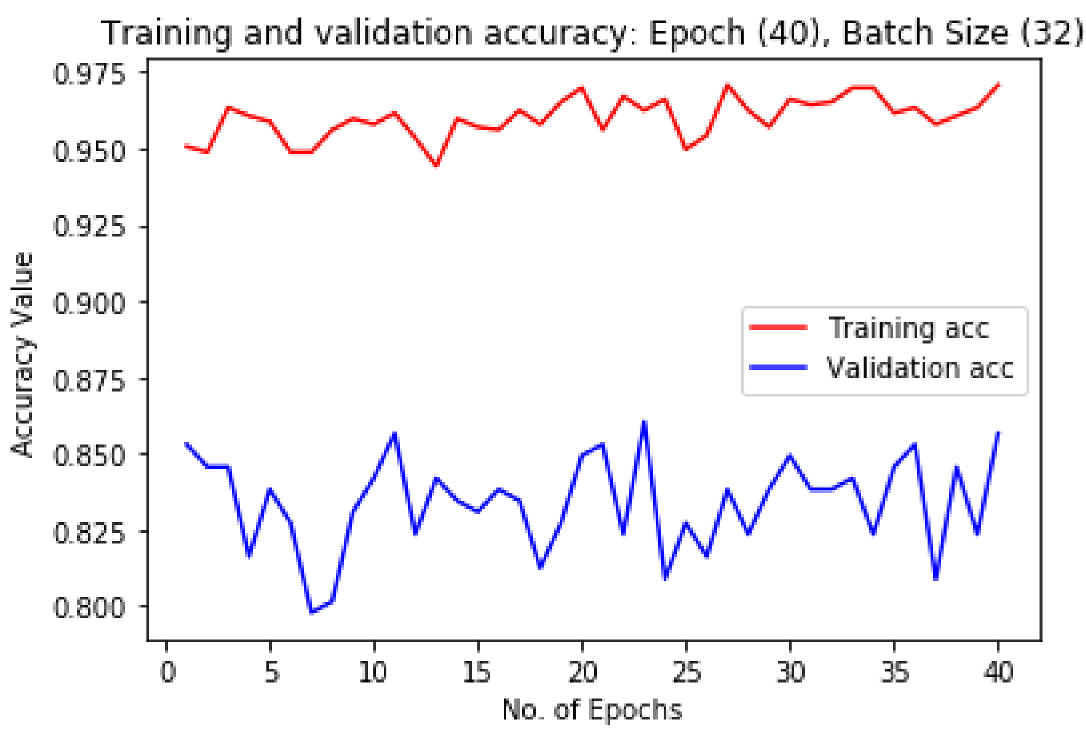


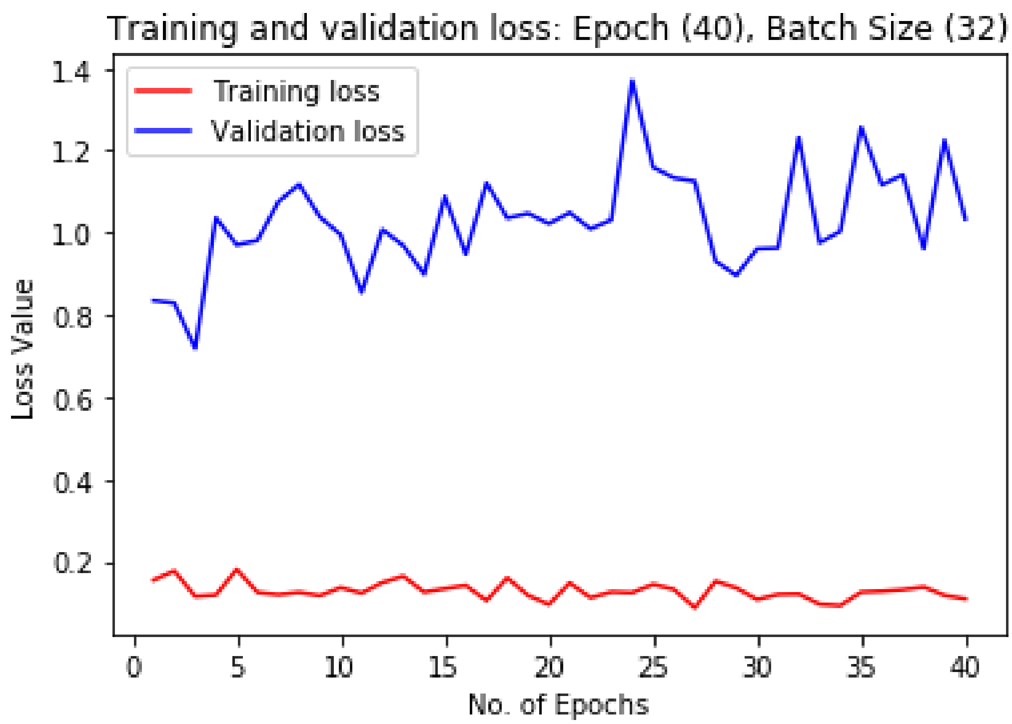
```
Test Accuracy: 83.0882%
Training Time: 454.4341125488281 sec
```






```
Test Accuracy: 85.2941%
Training Time: 1033.9515569210052 sec
```






```
Test Accuracy: 85.6618%
Training Time: 907.8175492286682 sec
```

1.4 4. Creating a CNN Model from Scratch

Create a CNN model with initial size of 224, 224, 3. CNN model would consist of following steps:

- a. Convolutional Layer with initial Input (244, 224, 3)
- b. Max Pooling to reduce special dimension
- c. Convolutional Layer
- d. Max Pooling
- e. Dropout
- f. Dense
- g. Softmax

1.4.1 Creating New model using three modularized functions

Created a New Function

1. monkey_mod => Creat new CNN model

Used Existing Functions

2. train_model => Train the created model
3. create_graph => Plot Accuracy and Loss functions

In [18]:
```
## Creating a Model from scratch after clearing session
def monkey_mod():

    ## To clear memory from the model. Remove the comment and run.
    backend.clear_session()

    ## To create a new model
    model = Sequential()

    model.add(Conv2D(filters=16, kernel_size=2, padding='same', activation='relu', inpu
    model.add(MaxPooling2D(pool_size=2))
    model.add(Conv2D(filters=32, kernel_size=2, padding='same', activation='relu'))
    model.add(MaxPooling2D(pool_size=2))
    model.add(Conv2D(filters=64, kernel_size=2, padding='same', activation='relu'))
    model.add(MaxPooling2D(pool_size=2))
    model.add(Conv2D(filters=128, kernel_size=2, padding='same', activation='relu'))
    model.add(MaxPooling2D(pool_size=2))
    model.add(Dropout(0.3))
    model.add(Dense(256, activation='relu'))
    ## Following doesn't make model efficient
    #model.add(Dropout(0.3))
    #model.add(Dense(128, activation='relu'))
    #model.add(Dropout(0.3))
    #model.add(Dense(64, activation='relu'))
    #model.add(Dropout(0.3))
    #model.add(Dense(32, activation='relu'))
    model.add(Flatten())
    model.add(Dense(11, activation='relu'))
    model.add(Dropout(0.3))
    model.add(Dense(11, activation='softmax'))

    ## Compiling the model
    model.compile(optimizer='rmsprop', loss='categorical_crossentropy', metrics=['accur
    return model
```

In [19]:
```
### Create model and train

## Submitting with below
```

```
epochs_val = [10,20,40]
batch_val = [16,32]

## Test with below
#epochs_val = [10]
#batch_val = [32]

model = monkey_mod()
(acc, val_acc, loss, val_loss, test_accu, time_taken) = train_model(model, epochs_val,
```

```
10   16
Train on 1093 samples, validate on 272 samples
Epoch 1/10
1093/1093 [==============================] - 5s 4ms/step - loss: 2.3931 - acc: 0.1144 - val_loss
Epoch 2/10
1093/1093 [==============================] - 4s 4ms/step - loss: 2.3125 - acc: 0.1629 - val_loss
Epoch 3/10
1093/1093 [==============================] - 4s 4ms/step - loss: 2.2280 - acc: 0.1958 - val_loss
Epoch 4/10
1093/1093 [==============================] - 4s 4ms/step - loss: 2.1033 - acc: 0.2461 - val_loss
Epoch 5/10
1093/1093 [==============================] - 4s 4ms/step - loss: 2.0379 - acc: 0.2617 - val_loss
Epoch 6/10
1093/1093 [==============================] - 4s 4ms/step - loss: 1.9633 - acc: 0.2983 - val_loss
Epoch 7/10
1093/1093 [==============================] - 4s 4ms/step - loss: 1.9074 - acc: 0.3239 - val_loss
Epoch 8/10
1093/1093 [==============================] - 4s 4ms/step - loss: 1.8136 - acc: 0.3513 - val_loss
Epoch 9/10
1093/1093 [==============================] - 4s 4ms/step - loss: 1.8095 - acc: 0.3468 - val_loss
Epoch 10/10
1093/1093 [==============================] - 4s 4ms/step - loss: 1.7683 - acc: 0.3513 - val_loss
Test accuracy: 41.9118%
Training Time: [44.07009315490723] sec
10   32
Train on 1093 samples, validate on 272 samples
Epoch 1/10
1093/1093 [==============================] - 4s 4ms/step - loss: 1.6784 - acc: 0.3614 - val_loss
Epoch 2/10
1093/1093 [==============================] - 4s 4ms/step - loss: 1.6769 - acc: 0.3916 - val_loss
Epoch 3/10
1093/1093 [==============================] - 4s 4ms/step - loss: 1.6243 - acc: 0.4126 - val_loss
Epoch 4/10
1093/1093 [==============================] - 4s 4ms/step - loss: 1.6150 - acc: 0.3898 - val_loss
Epoch 5/10
1093/1093 [==============================] - 4s 4ms/step - loss: 1.5945 - acc: 0.4209 - val_loss
Epoch 6/10
1093/1093 [==============================] - 4s 4ms/step - loss: 1.5555 - acc: 0.4318 - val_loss
```

```
Epoch 7/10
1093/1093 [==============================] - 4s 4ms/step - loss: 1.5374 - acc: 0.4209 - val_loss
Epoch 8/10
1093/1093 [==============================] - 4s 4ms/step - loss: 1.5690 - acc: 0.4273 - val_loss
Epoch 9/10
1093/1093 [==============================] - 4s 4ms/step - loss: 1.5202 - acc: 0.4254 - val_loss
Epoch 10/10
1093/1093 [==============================] - 4s 4ms/step - loss: 1.5162 - acc: 0.4437 - val_loss
Test accuracy: 49.2647%
Training Time: [44.07009315490723, 40.20986723899841] sec
20   16
Train on 1093 samples, validate on 272 samples
Epoch 1/20
1093/1093 [==============================] - 4s 4ms/step - loss: 1.5071 - acc: 0.4446 - val_loss
Epoch 2/20
1093/1093 [==============================] - 4s 4ms/step - loss: 1.4696 - acc: 0.4446 - val_loss
Epoch 3/20
1093/1093 [==============================] - 4s 4ms/step - loss: 1.5320 - acc: 0.4337 - val_loss
Epoch 4/20
1093/1093 [==============================] - 4s 4ms/step - loss: 1.4614 - acc: 0.4639 - val_loss
Epoch 5/20
1093/1093 [==============================] - 4s 4ms/step - loss: 1.4734 - acc: 0.4584 - val_loss
Epoch 6/20
1093/1093 [==============================] - 4s 4ms/step - loss: 1.4594 - acc: 0.4748 - val_loss
Epoch 7/20
1093/1093 [==============================] - 4s 4ms/step - loss: 1.4270 - acc: 0.4547 - val_loss
Epoch 8/20
1093/1093 [==============================] - 4s 4ms/step - loss: 1.3799 - acc: 0.4602 - val_loss
Epoch 9/20
1093/1093 [==============================] - 4s 4ms/step - loss: 1.3811 - acc: 0.4867 - val_loss
Epoch 10/20
1093/1093 [==============================] - 4s 4ms/step - loss: 1.4494 - acc: 0.4776 - val_loss
Epoch 11/20
1093/1093 [==============================] - 4s 4ms/step - loss: 1.3788 - acc: 0.4831 - val_loss
Epoch 12/20
1093/1093 [==============================] - 4s 4ms/step - loss: 1.3293 - acc: 0.4968 - val_loss
Epoch 13/20
1093/1093 [==============================] - 4s 4ms/step - loss: 1.3637 - acc: 0.4968 - val_loss
Epoch 14/20
1093/1093 [==============================] - 4s 4ms/step - loss: 1.3572 - acc: 0.5078 - val_loss
Epoch 15/20
1093/1093 [==============================] - 4s 4ms/step - loss: 1.3192 - acc: 0.5288 - val_loss
Epoch 16/20
1093/1093 [==============================] - 4s 4ms/step - loss: 1.3048 - acc: 0.5096 - val_loss
Epoch 17/20
1093/1093 [==============================] - 4s 4ms/step - loss: 1.3704 - acc: 0.4904 - val_loss
Epoch 18/20
1093/1093 [==============================] - 4s 4ms/step - loss: 1.3205 - acc: 0.5005 - val_loss
```

```
Epoch 19/20
1093/1093 [==============================] - 4s 4ms/step - loss: 1.3781 - acc: 0.4959 - val_loss
Epoch 20/20
1093/1093 [==============================] - 4s 4ms/step - loss: 1.3040 - acc: 0.5224 - val_loss
Test accuracy: 52.2059%
Training Time: [44.07009315490723, 40.20986723899841, 85.70699787139893] sec
20   32
Train on 1093 samples, validate on 272 samples
Epoch 1/20
1093/1093 [==============================] - 4s 4ms/step - loss: 1.2516 - acc: 0.5407 - val_loss
Epoch 2/20
1093/1093 [==============================] - 4s 4ms/step - loss: 1.3495 - acc: 0.5270 - val_loss
Epoch 3/20
1093/1093 [==============================] - 4s 4ms/step - loss: 1.2260 - acc: 0.5425 - val_loss
Epoch 4/20
1093/1093 [==============================] - 4s 4ms/step - loss: 1.2674 - acc: 0.5306 - val_loss
Epoch 5/20
1093/1093 [==============================] - 4s 4ms/step - loss: 1.3164 - acc: 0.5242 - val_loss
Epoch 6/20
1093/1093 [==============================] - 4s 4ms/step - loss: 1.2598 - acc: 0.5380 - val_loss
Epoch 7/20
1093/1093 [==============================] - 4s 4ms/step - loss: 1.2232 - acc: 0.5489 - val_loss
Epoch 8/20
1093/1093 [==============================] - 4s 4ms/step - loss: 1.3041 - acc: 0.5160 - val_loss
Epoch 9/20
1093/1093 [==============================] - 4s 4ms/step - loss: 1.2127 - acc: 0.5526 - val_loss
Epoch 10/20
1093/1093 [==============================] - 4s 4ms/step - loss: 1.2602 - acc: 0.5270 - val_loss
Epoch 11/20
1093/1093 [==============================] - 4s 4ms/step - loss: 1.2137 - acc: 0.5535 - val_loss
Epoch 12/20
1093/1093 [==============================] - 4s 4ms/step - loss: 1.2638 - acc: 0.5279 - val_loss
Epoch 13/20
1093/1093 [==============================] - 4s 4ms/step - loss: 1.2237 - acc: 0.5471 - val_loss
Epoch 14/20
1093/1093 [==============================] - 4s 4ms/step - loss: 1.2361 - acc: 0.5453 - val_loss
Epoch 15/20
1093/1093 [==============================] - 4s 4ms/step - loss: 1.2304 - acc: 0.5471 - val_loss
Epoch 16/20
1093/1093 [==============================] - 4s 4ms/step - loss: 1.1718 - acc: 0.5773 - val_loss
Epoch 17/20
1093/1093 [==============================] - 4s 4ms/step - loss: 1.2040 - acc: 0.5508 - val_loss
Epoch 18/20
1093/1093 [==============================] - 4s 4ms/step - loss: 1.2342 - acc: 0.5599 - val_loss
Epoch 19/20
1093/1093 [==============================] - 4s 4ms/step - loss: 1.1840 - acc: 0.5563 - val_loss
Epoch 20/20
1093/1093 [==============================] - 4s 4ms/step - loss: 1.2072 - acc: 0.5627 - val_loss
```

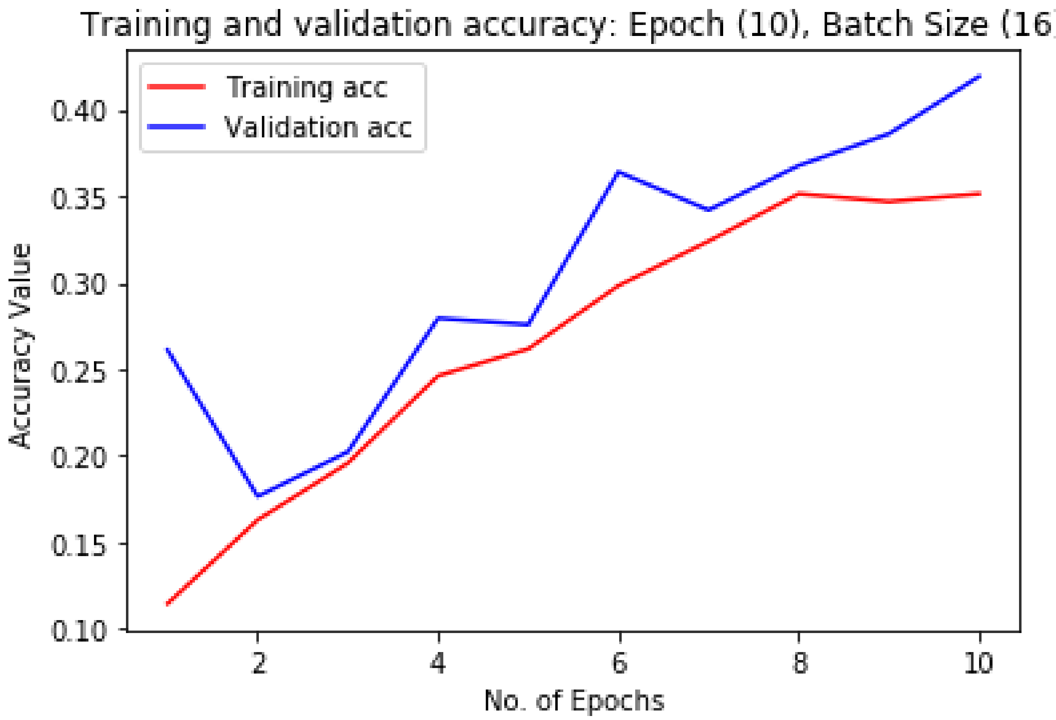
```
Test accuracy: 53.6765%
Training Time: [44.07009315490723, 40.20986723899841, 85.70699787139893, 80.04647326469421] sec
40   16
Train on 1093 samples, validate on 272 samples
Epoch 1/40
1093/1093 [==============================] - 4s 4ms/step - loss: 1.2420 - acc: 0.5489 - val_loss
Epoch 2/40
1093/1093 [==============================] - 4s 4ms/step - loss: 1.2520 - acc: 0.5645 - val_loss
Epoch 3/40
1093/1093 [==============================] - 4s 4ms/step - loss: 1.2813 - acc: 0.5480 - val_loss
Epoch 4/40
1093/1093 [==============================] - 4s 4ms/step - loss: 1.2724 - acc: 0.5389 - val_loss
Epoch 5/40
1093/1093 [==============================] - 4s 4ms/step - loss: 1.1940 - acc: 0.5755 - val_loss
Epoch 6/40
1093/1093 [==============================] - 4s 4ms/step - loss: 1.2575 - acc: 0.5508 - val_loss
Epoch 7/40
1093/1093 [==============================] - 4s 4ms/step - loss: 1.2732 - acc: 0.5672 - val_loss
Epoch 8/40
1093/1093 [==============================] - 4s 4ms/step - loss: 1.2329 - acc: 0.5471 - val_loss
Epoch 9/40
1093/1093 [==============================] - 4s 4ms/step - loss: 1.2331 - acc: 0.5746 - val_loss
Epoch 10/40
1093/1093 [==============================] - 4s 4ms/step - loss: 1.1660 - acc: 0.5746 - val_loss
Epoch 11/40
1093/1093 [==============================] - 4s 4ms/step - loss: 1.2843 - acc: 0.5444 - val_loss
Epoch 12/40
1093/1093 [==============================] - 4s 4ms/step - loss: 1.2316 - acc: 0.5691 - val_loss
Epoch 13/40
1093/1093 [==============================] - 4s 4ms/step - loss: 1.2924 - acc: 0.5453 - val_loss
Epoch 14/40
1093/1093 [==============================] - 4s 4ms/step - loss: 1.2741 - acc: 0.5599 - val_loss
Epoch 15/40
1093/1093 [==============================] - 4s 4ms/step - loss: 1.2477 - acc: 0.5929 - val_loss
Epoch 16/40
1093/1093 [==============================] - 4s 4ms/step - loss: 1.2426 - acc: 0.5627 - val_loss
Epoch 17/40
1093/1093 [==============================] - 4s 4ms/step - loss: 1.2329 - acc: 0.5654 - val_loss
Epoch 18/40
1093/1093 [==============================] - 4s 4ms/step - loss: 1.2455 - acc: 0.5636 - val_loss
Epoch 19/40
1093/1093 [==============================] - 4s 4ms/step - loss: 1.2465 - acc: 0.5663 - val_loss
Epoch 20/40
1093/1093 [==============================] - 4s 4ms/step - loss: 1.2372 - acc: 0.5590 - val_loss
Epoch 21/40
1093/1093 [==============================] - 4s 4ms/step - loss: 1.1956 - acc: 0.5929 - val_loss
Epoch 22/40
1093/1093 [==============================] - 4s 4ms/step - loss: 1.1817 - acc: 0.5947 - val_loss
```

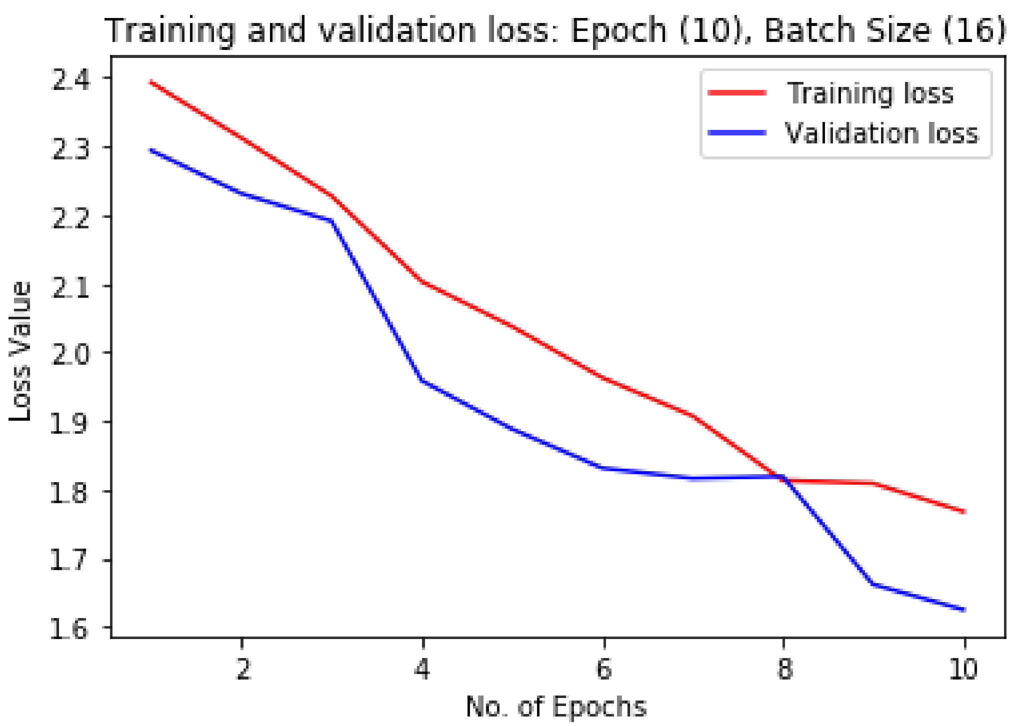
```
Epoch 23/40
1093/1093 [==============================] - 4s 4ms/step - loss: 1.2616 - acc: 0.5709 - val_loss
Epoch 24/40
1093/1093 [==============================] - 4s 4ms/step - loss: 1.1846 - acc: 0.5773 - val_loss
Epoch 25/40
1093/1093 [==============================] - 4s 4ms/step - loss: 1.2209 - acc: 0.5819 - val_loss
Epoch 26/40
1093/1093 [==============================] - 4s 4ms/step - loss: 1.2208 - acc: 0.5727 - val_loss
Epoch 27/40
1093/1093 [==============================] - 4s 4ms/step - loss: 1.2451 - acc: 0.5645 - val_loss
Epoch 28/40
1093/1093 [==============================] - 4s 4ms/step - loss: 1.3129 - acc: 0.5398 - val_loss
Epoch 29/40
1093/1093 [==============================] - 4s 4ms/step - loss: 1.2027 - acc: 0.5947 - val_loss
Epoch 30/40
1093/1093 [==============================] - 4s 4ms/step - loss: 1.2169 - acc: 0.5846 - val_loss
Epoch 31/40
1093/1093 [==============================] - 4s 4ms/step - loss: 1.2135 - acc: 0.5801 - val_loss
Epoch 32/40
1093/1093 [==============================] - 4s 4ms/step - loss: 1.2547 - acc: 0.5828 - val_loss
Epoch 33/40
1093/1093 [==============================] - 4s 4ms/step - loss: 1.2098 - acc: 0.5773 - val_loss
Epoch 34/40
1093/1093 [==============================] - 4s 4ms/step - loss: 1.2155 - acc: 0.5737 - val_loss
Epoch 35/40
1093/1093 [==============================] - 4s 4ms/step - loss: 1.2699 - acc: 0.5828 - val_loss
Epoch 36/40
1093/1093 [==============================] - 4s 4ms/step - loss: 1.1952 - acc: 0.5855 - val_loss
Epoch 37/40
1093/1093 [==============================] - 4s 4ms/step - loss: 1.1455 - acc: 0.6011 - val_loss
Epoch 38/40
1093/1093 [==============================] - 4s 4ms/step - loss: 1.2168 - acc: 0.5865 - val_loss
Epoch 39/40
1093/1093 [==============================] - 4s 4ms/step - loss: 1.1972 - acc: 0.5654 - val_loss
Epoch 40/40
1093/1093 [==============================] - 4s 4ms/step - loss: 1.2123 - acc: 0.5901 - val_loss
Test accuracy: 53.6765%
Training Time: [44.07009315490723, 40.20986723899841, 85.70699787139893, 80.04647326469421, 172.
40   32
Train on 1093 samples, validate on 272 samples
Epoch 1/40
1093/1093 [==============================] - 4s 4ms/step - loss: 1.2128 - acc: 0.5773 - val_loss
Epoch 2/40
1093/1093 [==============================] - 4s 4ms/step - loss: 1.2667 - acc: 0.5700 - val_loss
Epoch 3/40
1093/1093 [==============================] - 4s 4ms/step - loss: 1.1590 - acc: 0.5965 - val_loss
Epoch 4/40
1093/1093 [==============================] - 4s 4ms/step - loss: 1.1304 - acc: 0.6130 - val_loss
```

```
Epoch 5/40
1093/1093 [==============================] - 4s 4ms/step - loss: 1.1986 - acc: 0.5984 - val_loss
Epoch 6/40
1093/1093 [==============================] - 4s 4ms/step - loss: 1.2191 - acc: 0.5892 - val_loss
Epoch 7/40
1093/1093 [==============================] - 4s 4ms/step - loss: 1.1961 - acc: 0.5929 - val_loss
Epoch 8/40
1093/1093 [==============================] - 4s 4ms/step - loss: 1.0999 - acc: 0.6221 - val_loss
Epoch 9/40
1093/1093 [==============================] - 4s 4ms/step - loss: 1.2310 - acc: 0.5837 - val_loss
Epoch 10/40
1093/1093 [==============================] - 4s 4ms/step - loss: 1.2352 - acc: 0.5746 - val_loss
Epoch 11/40
1093/1093 [==============================] - 4s 4ms/step - loss: 1.2061 - acc: 0.5773 - val_loss
Epoch 12/40
1093/1093 [==============================] - 4s 4ms/step - loss: 1.2295 - acc: 0.5837 - val_loss
Epoch 13/40
1093/1093 [==============================] - 4s 4ms/step - loss: 1.2406 - acc: 0.5691 - val_loss
Epoch 14/40
1093/1093 [==============================] - 4s 4ms/step - loss: 1.1765 - acc: 0.6139 - val_loss
Epoch 15/40
1093/1093 [==============================] - 4s 4ms/step - loss: 1.1405 - acc: 0.6249 - val_loss
Epoch 16/40
1093/1093 [==============================] - 4s 4ms/step - loss: 1.1257 - acc: 0.6295 - val_loss
Epoch 17/40
1093/1093 [==============================] - 4s 4ms/step - loss: 1.1929 - acc: 0.6029 - val_loss
Epoch 18/40
1093/1093 [==============================] - 4s 4ms/step - loss: 1.1239 - acc: 0.6295 - val_loss
Epoch 19/40
1093/1093 [==============================] - 4s 4ms/step - loss: 1.2052 - acc: 0.5984 - val_loss
Epoch 20/40
1093/1093 [==============================] - 4s 4ms/step - loss: 1.0995 - acc: 0.6450 - val_loss
Epoch 21/40
1093/1093 [==============================] - 4s 4ms/step - loss: 1.2057 - acc: 0.5892 - val_loss
Epoch 22/40
1093/1093 [==============================] - 4s 4ms/step - loss: 1.1579 - acc: 0.6157 - val_loss
Epoch 23/40
1093/1093 [==============================] - 4s 4ms/step - loss: 1.1423 - acc: 0.6066 - val_loss
Epoch 24/40
1093/1093 [==============================] - 4s 4ms/step - loss: 1.1428 - acc: 0.5993 - val_loss
Epoch 25/40
1093/1093 [==============================] - 4s 4ms/step - loss: 1.1355 - acc: 0.6139 - val_loss
Epoch 26/40
1093/1093 [==============================] - 4s 4ms/step - loss: 1.1471 - acc: 0.5947 - val_loss
Epoch 27/40
1093/1093 [==============================] - 4s 4ms/step - loss: 1.1475 - acc: 0.5892 - val_loss
Epoch 28/40
1093/1093 [==============================] - 4s 4ms/step - loss: 1.1826 - acc: 0.5919 - val_loss
```

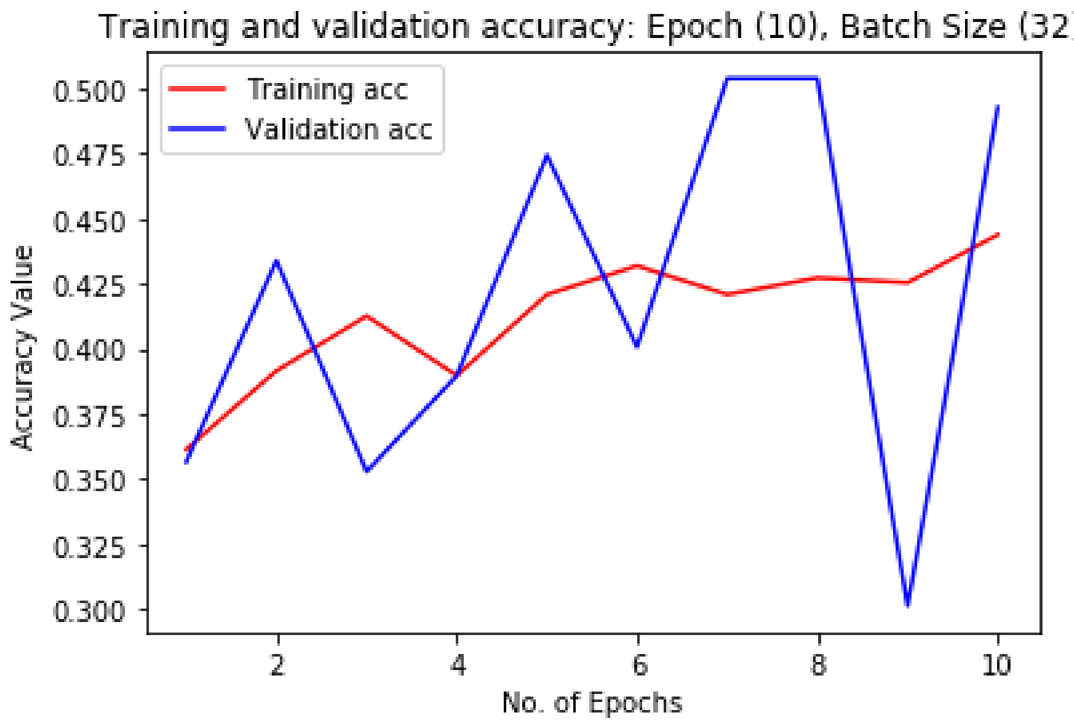
```
Epoch 29/40
1093/1093 [==============================] - 4s 4ms/step - loss: 1.2361 - acc: 0.6011 - val_loss
Epoch 30/40
1093/1093 [==============================] - 4s 4ms/step - loss: 1.1863 - acc: 0.5947 - val_loss
Epoch 31/40
1093/1093 [==============================] - 4s 4ms/step - loss: 1.1590 - acc: 0.6258 - val_loss
Epoch 32/40
1093/1093 [==============================] - 4s 4ms/step - loss: 1.1468 - acc: 0.6048 - val_loss
Epoch 33/40
1093/1093 [==============================] - 4s 4ms/step - loss: 1.2941 - acc: 0.5801 - val_loss
Epoch 34/40
1093/1093 [==============================] - 4s 4ms/step - loss: 1.1029 - acc: 0.6368 - val_loss
Epoch 35/40
1093/1093 [==============================] - 4s 4ms/step - loss: 1.1772 - acc: 0.5892 - val_loss
Epoch 36/40
1093/1093 [==============================] - 4s 4ms/step - loss: 1.1330 - acc: 0.6203 - val_loss
Epoch 37/40
1093/1093 [==============================] - 4s 4ms/step - loss: 1.1018 - acc: 0.6331 - val_loss
Epoch 38/40
1093/1093 [==============================] - 4s 4ms/step - loss: 1.2481 - acc: 0.6112 - val_loss
Epoch 39/40
1093/1093 [==============================] - 4s 4ms/step - loss: 1.1878 - acc: 0.6112 - val_loss
Epoch 40/40
1093/1093 [==============================] - 4s 4ms/step - loss: 1.1102 - acc: 0.6194 - val_loss
Test accuracy: 55.5147%
Training Time: [44.07009315490723, 40.20986723899841, 85.70699787139893, 80.04647326469421, 172.
```

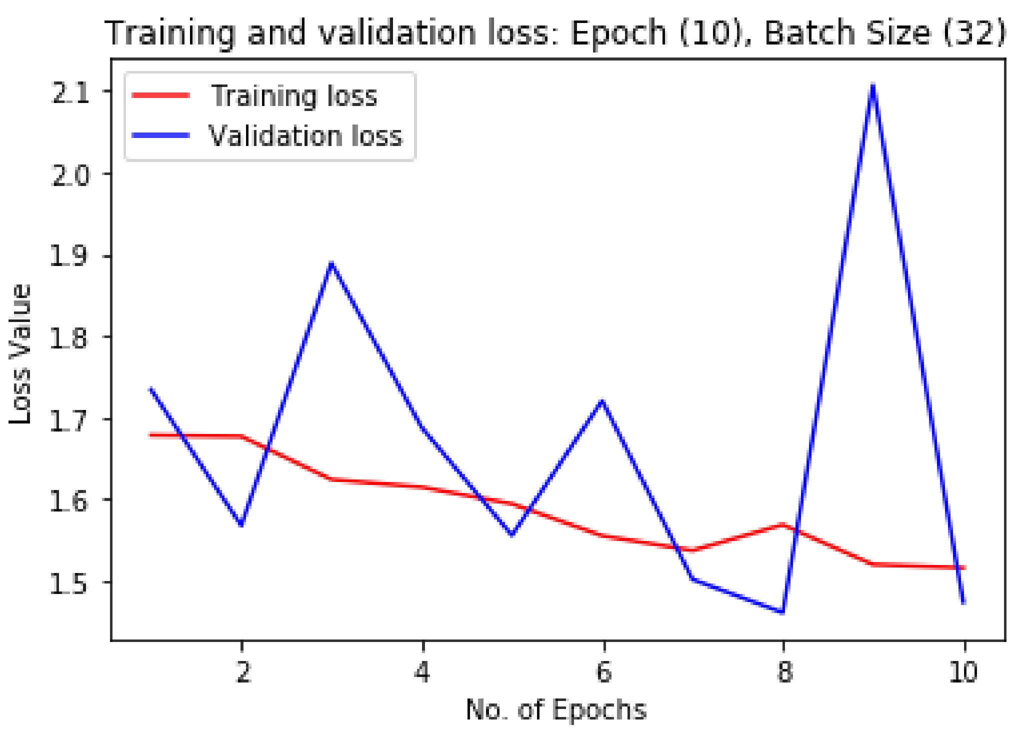
In [20]:
```
## Calling graph function to Plot the models
create_graph(epochs_val, batch_val, acc, val_acc, loss, val_loss, test_accu, time_taken
```


Training and validation accuracy: Epoch (10), Batch Size (16)
Training acc
Validation acc
Accuracy Value
No. of Epochs
0.10
0.15
0.20
0.25
0.30
0.35
0.40
2
4
6
8
10

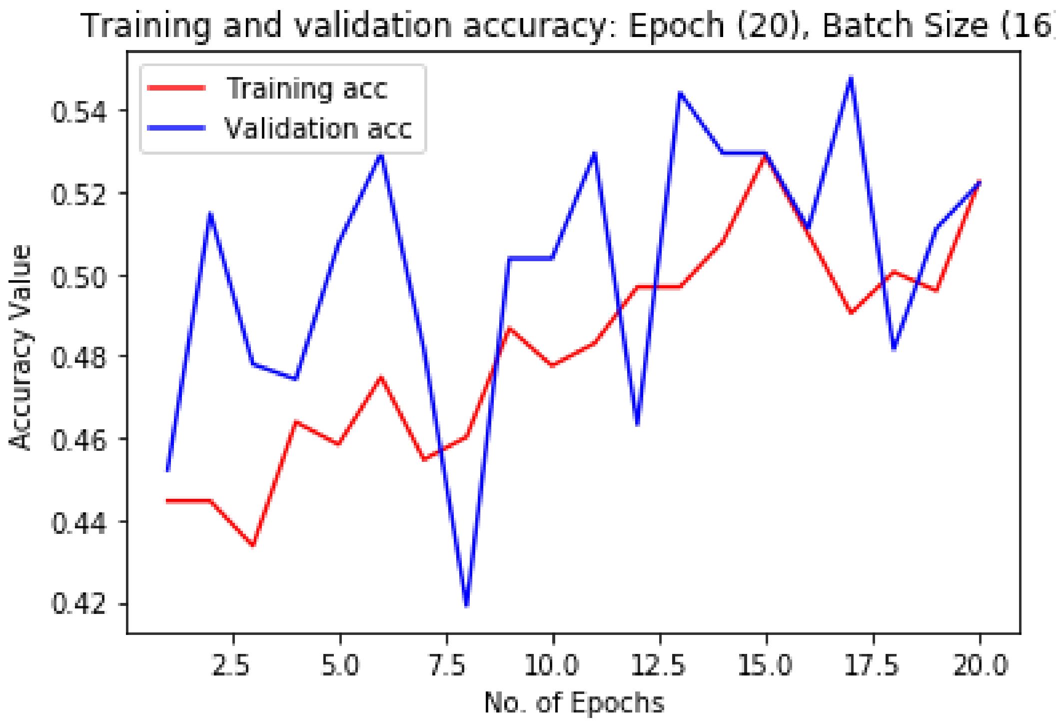

Training and validation loss: Epoch (10), Batch Size (16)
Training loss
Validation loss
Loss Value
No. of Epochs
1.6
1.7
1.8
1.9
2.0
2.1
2.2
2.3
2.4
2
4
6
8
10

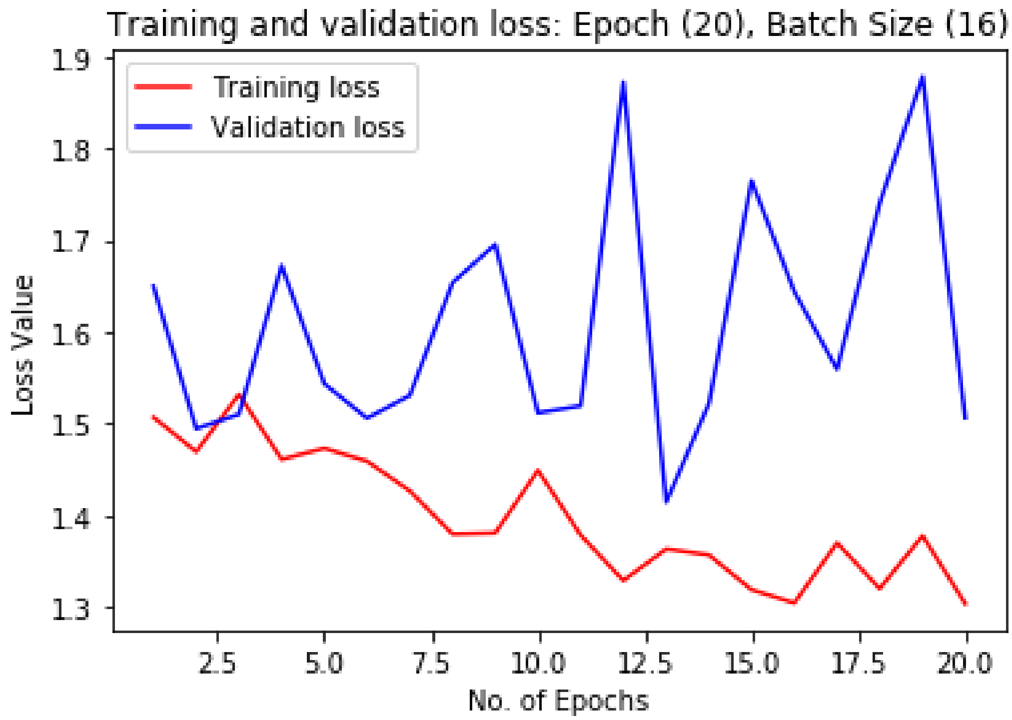
```
Test Accuracy: 41.9118%
Training Time: 44.07009315490723 sec
```



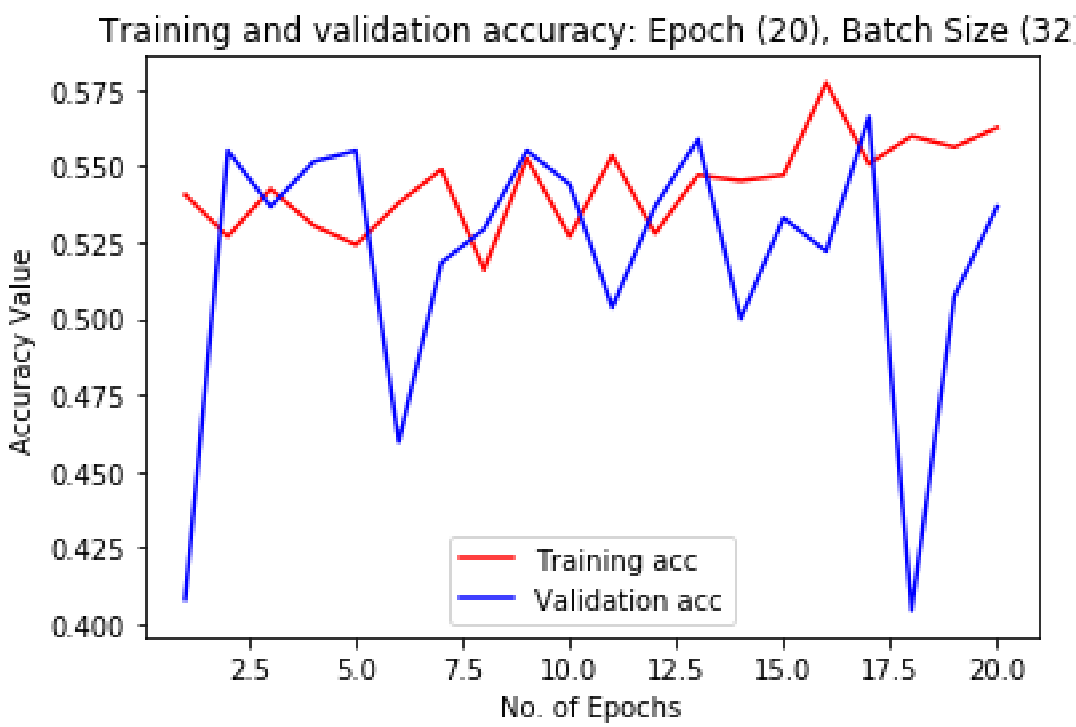


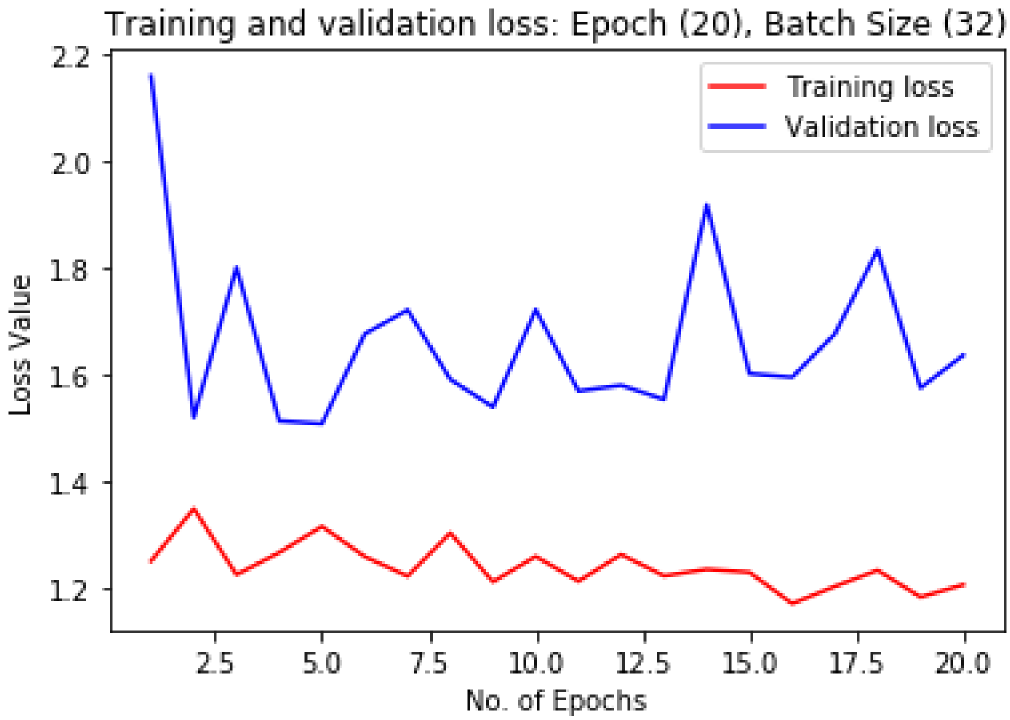
```
Test Accuracy: 49.2647%
Training Time: 40.20986723899841 sec
```



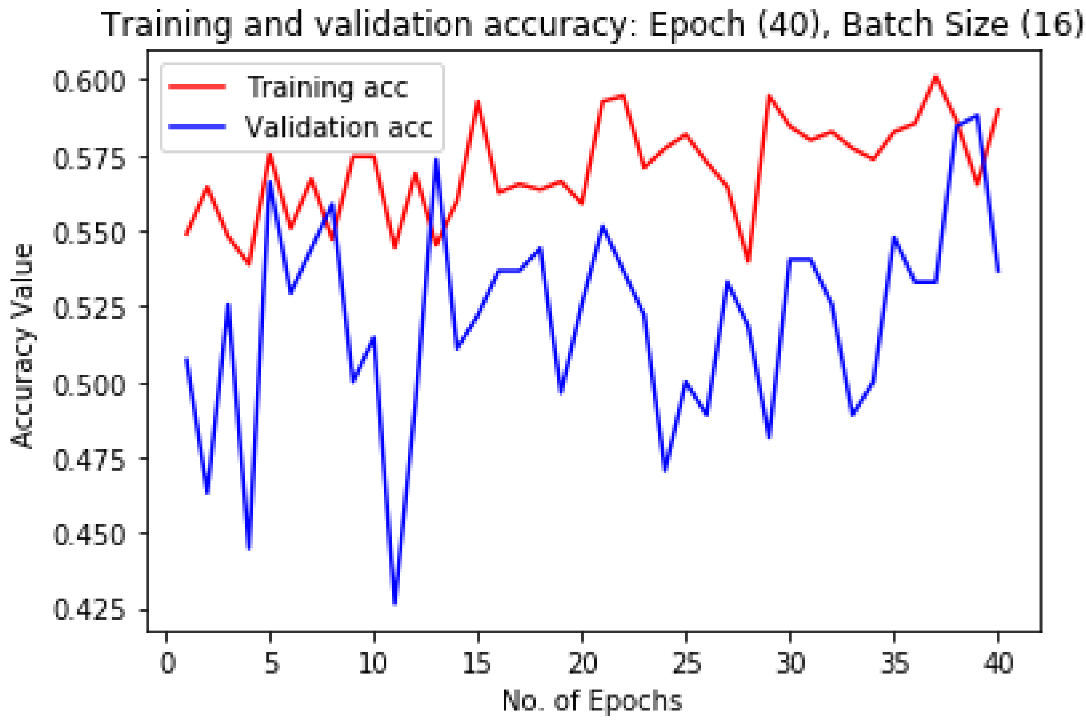


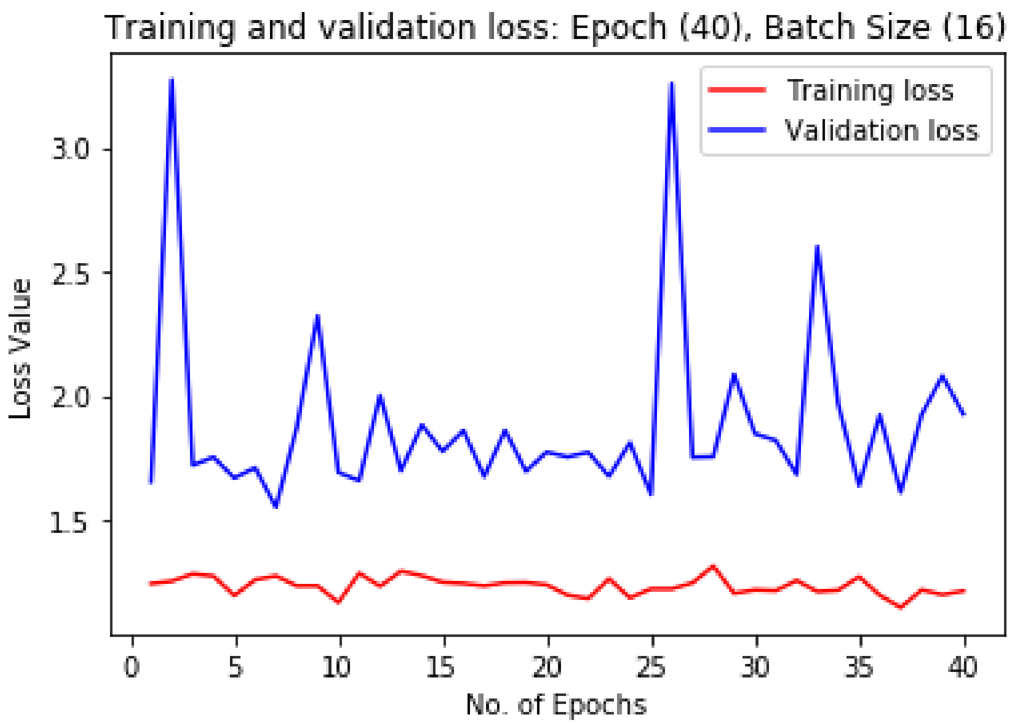

```
Test Accuracy: 52.2059%
Training Time: 85.70699787139893 sec
```



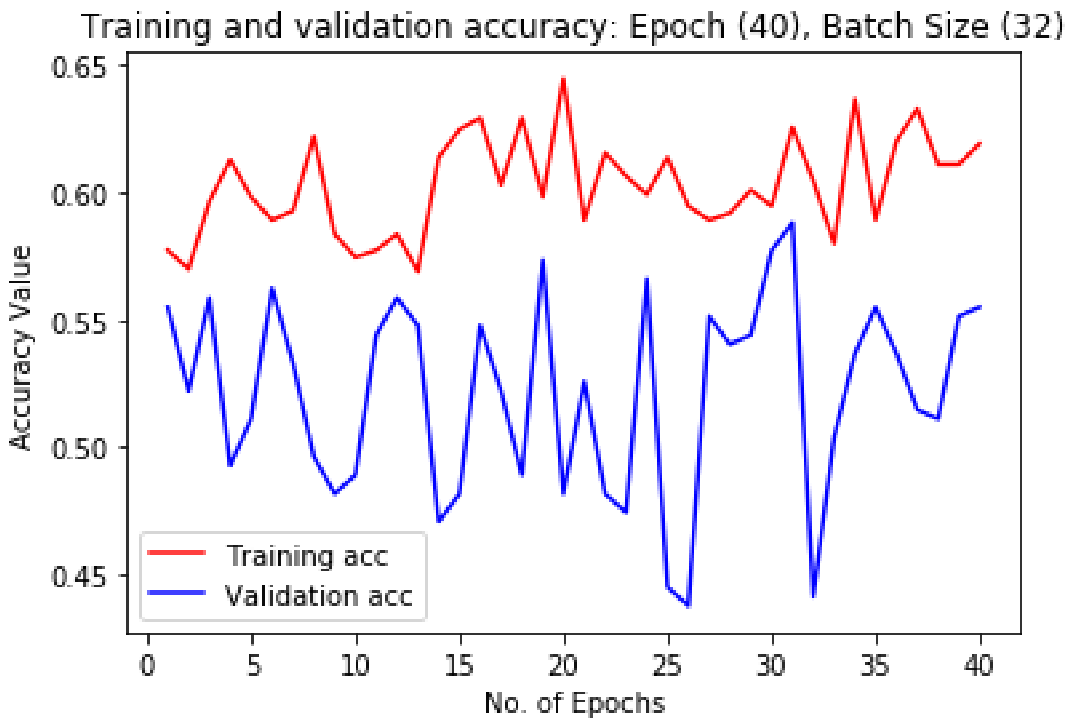


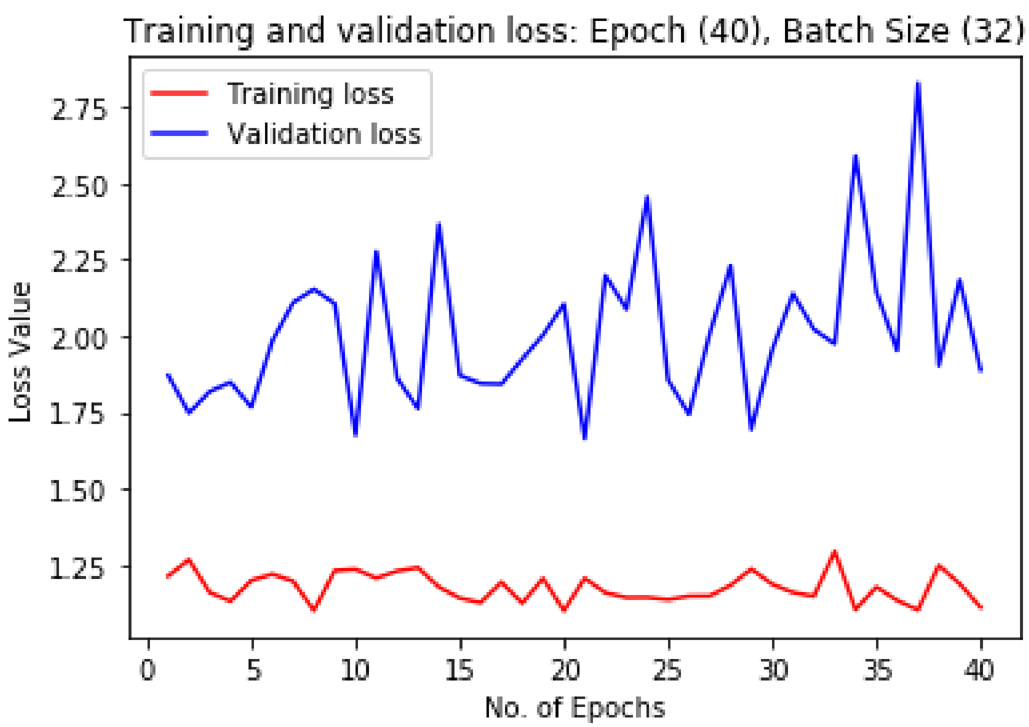

```
Test Accuracy: 53.6765%
Training Time: 80.04647326469421 sec
```


Training and validation accuracy: Epoch (40), Batch Size (16)
Training acc
Validation acc
Accuracy Value
No. of Epochs
0.600
0.575
0.550
0.525
0.500
0.475
0.450
0.425
0
5
10
15
20
25
30
35
40


Training and validation loss: Epoch (40), Batch Size (16)
Training loss
Validation loss
Loss Value
No. of Epochs
3.0
2.5
2.0
1.5
0
5
10
15
20
25
30
35
40

```
Test Accuracy: 53.6765%
Training Time: 172.35104942321777 sec
```





```
Test Accuracy: 55.5147%
Training Time: 159.63770246505737 sec
```

1.5 #### End of Project