Lab 8
Neural Networks

Neural Network Lab

The purpose of this lab is to introduce neural networks with python. We will be using the CNN code from TensorFlow. The code includes a dataset with pictures and classifications that will allow pictures to be classified through training. Neural network training works by comparing a picture with pictures already identified in the dataset. Each epoch will give a more accurate result which will lead to eventually identifying the picture with a certain accuracy. The changes made to the CNN code is increasing the number of epochs to 150. This code took a total of 10 hours to complete. Highest accuracy reached is 87%.

```
optimizer = tf.keras.optimizers.SGD(lr=0.01, momentum=0.9)
model.compile(optimizer='adam',
              loss=tf.keras.losses.SparseCategoricalCrossentropy(from_logits=True),
              metrics=['accuracy'])

history = model.fit(train_images, train_labels, epochs=150, batch_size=64,
                    validation_data=(test_images, test_labels))
```

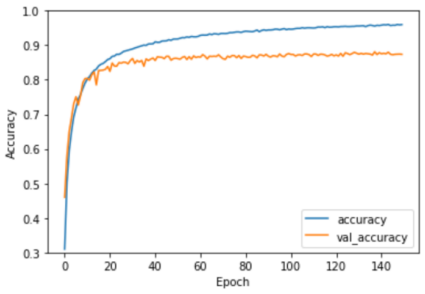
```
uracy: 0.8744
Epoch 150/150
782/782 [==============================] - 175s 224ms/step - loss: 0.1194 - accuracy: 0.9598 - val_loss: 0.5149 - val_acc
uracy: 0.8738
```

Evaluate the model

In [21]:
```
plt.plot(history.history['accuracy'], label='accuracy')
plt.plot(history.history['val_accuracy'], label = 'val_accuracy')
plt.xlabel('Epoch')
plt.ylabel('Accuracy')
plt.ylim([0.3, 1])
plt.legend(loc='lower right')

test_loss, test_acc = model.evaluate(test_images,  test_labels, verbose=2)
```

```
313/313 - 8s - loss: 0.5149 - accuracy: 0.8738 - 8s/epoch - 26ms/step
```



In [22]:
```
print(test_acc)
```

```
0.8737999796867371
```

The following test images were used to test the code. The results are displayed below.

TOI

```
Downloading data from https://sniteartmuseum.nd.edu/assets/166204/original/ferrari.jpg
98304/96948 [==============================] - 0s 1us/step
106496/96948 [==============================] - 0s 1us/step
This image most likely belongs to automobile with a 23.19 percent confidence.
tf.Tensor(
[0.08533984 0.23194769 0.0853384  0.08533934 0.08533829 0.08533818
 0.08533902 0.08533832 0.08533838 0.08534261], shape=(10,), dtype=float32)
Downloading data from https://upload.wikimedia.org/wikipedia/commons/f/f0/White_horse.jpg
221184/216027 [==============================] - 0s 0us/step
229376/216027 [==============================] - 0s 0us/step
This image most likely belongs to truck with a 23.20 percent confidence.
tf.Tensor(
[0.08533788 0.08533814 0.08533973 0.08534046 0.08533882 0.08533968
 0.08533791 0.08533809 0.08533787 0.23195143], shape=(10,), dtype=float32)
Downloading data from https://static.toiimg.com/thumb/msid-67586673,width-1070,height-580,overlay-toi_sw,pt-32,y_pad-40,resizemode-75,imgsize-3918697/67586673.jpg
65536/60395 [==================================] - 0s 0us/step
73728/60395 [==================================] - 0s 0us/step
This image most likely belongs to truck with a 23.20 percent confidence.
tf.Tensor(
[0.08533788 0.08533814 0.08533973 0.08534046 0.08533882 0.08533968
 0.08533791 0.08533809 0.08533787 0.23195143], shape=(10,), dtype=float32)
Downloading data from https://upload.wikimedia.org/wikipedia/commons/thumb/d/d3/Voyager_aircraft.jpg/1200px-Voyager_aircraft.jpg
385024/384113 [==============================] - 0s 0us/step
393216/384113 [==============================] - 0s 0us/step
This image most likely belongs to automobile with a 23.19 percent confidence.
tf.Tensor(
[0.08533984 0.23194769 0.0853384  0.08533934 0.08533829 0.08533818
 0.08533902 0.08533832 0.08533838 0.08534261], shape=(10,), dtype=float32)
Downloading data from https://images.all-free-download.com/images/graphiclarge/flying_bird_201952.jpg
16384/14677 [==============================] - 0s 0us/step
24576/14677 [==================================================] - 0s 0us/step
This image most likely belongs to automobile with a 23.19 percent confidence.
tf.Tensor(
[0.08533984 0.23194769 0.0853384  0.08533934 0.08533829 0.08533818
 0.08533902 0.08533832 0.08533838 0.08534261], shape=(10,), dtype=float32)
```

Balloon Game

The game works by trying to navigate a balloon above trees and under birds. Click to make the balloon move up. The balloon will fall at a constant speed. Added levels, change speed, changed gravity, changed spacing