**Step 1:** Update Your Raspberry Pi and its packages.

*sudo apt-get update*

*sudo apt-get upgrade*

**Step 2:** Test that you have latest python version, using this command.

*python3 –-version*

It is recommended to have at least Python 3.4.

**Step 3:** We need to install libatlas library (ATLAS - Automatically Tuned Linear Algebra Software). Because TensorFlow uses numpy. So, install it using the following command

*sudo apt install libatlas-base-dev*

**Step 4:** Install TensorFlow using Pip3 install command.

*pip3 install tensorflow*

**Step 5: Predicting an Image Using Imagenet Model Example:**

TensorFlow has published a model to predict images. You need to download the model first then run it.

**Step 1:**Run the following command to download the models. You might need to have git installed.

*git clone https://github.com/tensorflow/models.git*

**Step 2:** Navigate to imagenet example.

*cd models/tutorials/image/imagenet*

**Pro Tip:** On the new Raspbian Stretch, you can find the ‘classify\_image.py’ file manually and then ‘Right Click’ on it. Choose ‘Copy Path(s)’. Then paste it in terminal after the ‘cd’ and press enter. By this way you can navigate faster without any errors

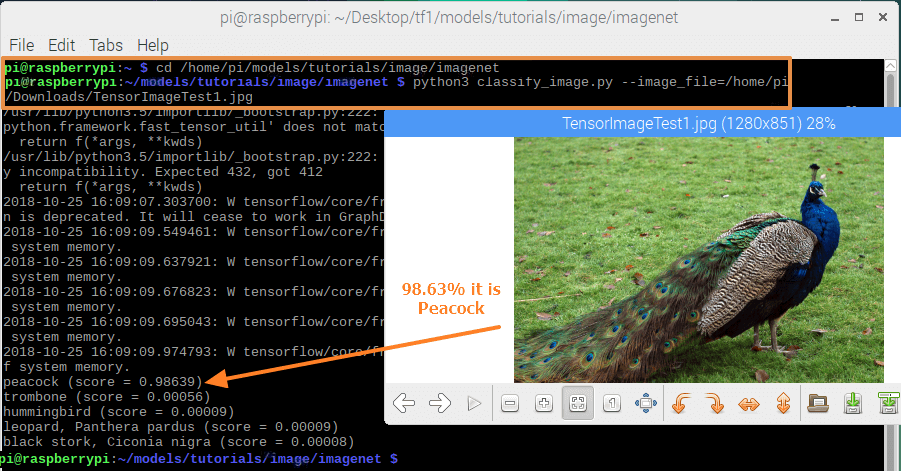
I used ‘Copy Path(s)’ method so it will include the exact path on the image (/home/pi).

**Step 3:** Run the example using this command. It will take about 30 seconds to show the predicted result.

*python3 classify\_image.py -- image\_file=/home/pi/Images/peacock.jpg*

*python3 classify\_image.py --image\_file=/home/pi/Downloads/Test1.jpg*

**Step 6: Custom Image Prediction**

[](https://content.instructables.com/ORIG/F3Q/BTQH/JNOVHY46/F3QBTQHJNOVHY46.png?auto=webp&frame=1&fit=bounds&md=3aacf01759dc157ae20d4965c8304cd6)

For better results use less memory images.

You can try out other examples too. But you need to install necessary packages before execution. We will cover some interesting TensorFlow topics in the upcoming tutorials.

Check out our [FactoryForward blog](https://www.factoryforward.com/blog/) for more tutorials on Arduino, Raspberry Pi, microBit,etc.

Tambien de esta forma se puede tener base de datos de emociones, capturas la parte de la cara (boca, ojos) que expresa emociones y esa es comparadara para ver el estado de animo.

De ahi que una empresa puede vender la base de datos de emociones, animales, objetos de transito, tipos de autos.

Arduino usa sensores basados en voltaje, TensorFLow es como sensor que simula el oido y el ojo.

En que ayuda? Ayuda en una tarea dificil, detectar en los Rayos X alguna anomalia, tienes una base de datos de diferentes tipos de tumores en sus etapas iniciales, un pequeno tumor es dificil de detectar, es como en el cuadro de busque las 7 diferencias, ese scaneo, pero si lo metes con esto da al menos la referencia de donde puede haber algo malo.

**Step 1: Machine Learning**

Machine Learning and Deep Learning will come under Artificial Intelligence (AI). A Machine Learning will observe and analyze the available data and improves it results over time.

**Example:** YouTube Recommended videos feature. It shows related videos that you viewed before. The prediction is limited to text-based results only. But deep learning can go deeper than this.

**Step 2: Deep Learning**

The deep learning is almost similar to that, but it makes more accurate decision on its own by collecting various information of an object. It has many layers of analysis and takes a decision according to it. To fasten the process, it uses Neural Network and provides us more exact result that we needed (means better prediction than ML). Something like how a human brain thinks and makes decisions.

**Example:** Object detection. It detects what is available in an image. Something similar that you can differentiate an Arduino and Raspberry Pi by its appearance, size and Colors.