# Install the the following package

pip install Flask

pip install pillow

pip install torchvision

pip install torch

pip install transformers

pip install numpy

pip install scipy

pip install urllib.request

Then, you need to make sure your device is connected to the internet to use this application.

# How to use

Step 1: Start a command prompt at the root of the project

Step 2: Run this command FLASK\_DEBUG=1 FLASK\_APP=app.py flask run

Step 3: Go to <http://localhost:5000/predict>

Step 4: Select which model you want to use for the image classification (fist line) and the sentiment classification (second line). By default, it is set to twitter-roberta and densenet121.

Step 5: Select an image of an animal or an object you would like to classify (.jpg formant only) and write your sentiment in English language.

Step 6: Upload and wait a little while (it can take several seconds, but the first time try might take longer).

# Image classification models

For this classifier, we want to be able to classify images of a large number of objects / animals. Therefore, we turned to two Densenet models (Dense Convolutional Network). According to the following article: <https://arxiv.org/abs/1608.06993>, These models have been trained and tested with different datasets (CIFAR-10, CIFAR-100, SVHN, and ImageNet).

The Input is a .jpg image of 3 x 224 x 224 and they are 1000 outputs.

We can therefore have confidence in these two models.  
According to Pytorch's website, Densenet121 would be the least performing of the Densenet models while Densenet161 would be the best.

# Sentiment classification models

For this task, we used two different models.

The first is a Roberta model. It takes a sentence in input and has 3 outputs (positive, neutral and negative). He was trained with a dataset composed of 58 million of English tweets so we can consider it as reliable but only for English sentences and not two long sentences (a tweet is a maximum of 140 characters long).

The second is Bertweet-base-sentiment-analysis. It has the same input and output as the first one, so it was simpler to implement it. It based on BERTweet model (trained with 850M English tweets), a Roberta model and retrained with SemEval 2017 corpus (around ~40k tweets). We can also consider it as reliable with the same condition of the first one.