This tutorial consists of 4 parts:

* Dependencies
* Data processing.
* Training.
* Prediction.

1. **Dependencies**

* Python 3.5.2
* Keras 2.2.0
* Tensorflow-gpu 1.4.1
* Opencv 3.4.1
* Numpy 1.14.5

All the packages are built on Linux.

1. **Data processing**
2. **Overview**

* Clean the dataset (remove junk file, error file, etc).
* Splits the training dataset into 2 sets:

+ Training set (90% of the dataset).

+ Validation set (10% of the dataset).

1. **Coding implementation**

* To clean the dataset, we run:

**python3 prepare\_image.py**

+ Set up the data link in line 9.

* To splits the dataset, we run:

**python3 splitdata\_python.py**

1. **Training**
2. **Overview**

* This model is a cascade model of 3 networks: InceptionResnetV2, Xception, and Resnet101 network.
* In all the networks, we used pretrained model based on Imagenet classification.
* Some data augmentation techniques have been using:

+ rescale=1/255

+ rotation\_range=20

+ width\_shift\_range=0.2

+ height\_shift\_range=0.2

+ shear\_range=0.2

+ zoom\_range=0.2

+ horizontal\_flip=True

+ fill\_mode='nearest'

* It took 36 hours for each sub-model (~20 epochs) training on one NVIDIA GeForce GTX 1080 Ti.

1. **Coding implementation**

* To train the InceptionResnetV2/Xception/Resnet101 network, we run:

**python3** **train\_inceptionresnet.py/train\_xception.py/train\_resnet101.py**

with some modifications:

+ Load pretrained model:

* **train\_inceptionresnet.py:** Line 85.
* **train\_xception.py:**
* **train\_resnet101.py:** Line 168.

+ Load dataset:

* **train\_inceptionresnet.py:** Line 18, 19.
* **train\_xception.py:** Line 18, 19.
* **train\_resnet101.py:** Line 126, 127.

1. **Prediction**
2. **Overview**

* At first, we run the prediction separately on 3 networks, which creates 3 numpy prediction matrix.
* Then, we take the sum and calculate the final result (top-3 predictions) using softmax.

1. **Coding implementation**

* To get the prediction matrix of 3 networks, we run:
* **python3 prediction\_inceptionresnet.py**
* **python3 prediction\_xception.py**
* **python3 prediction\_resnet101.py**

with some modifications:

+ Load trained model:

* **prediction\_inceptionresnet.py:** Line 127.
* **prediction\_xception.py**
* **prediction\_resnet101.py:** Line 127.
* To get the final result csv file in the result section, we run: **python3 predict\_cascade.py**