ImageNet

조 영 혁

노다시스템

Classifying Images with ImageNet

Imagenet => Image recognition

- 어떠한 사진을 보여주었을 때 이 사진이 무엇인지 맞출 수 있는 컴퓨터를 만드는 프로젝트로
- 2015년 오차율이 3%이내
- Imagenet 데이터는 1000개의 카테고리 수백만개이 이미지 데이터로 구성

유명한 CNN

- ResNet
- VGG16
- VGG19
- Inception_V3
- Xception
- TensorFlow

ImageNet

1. console 프로그램을 이용한 jetson 사용하기
- classifcation networks that have been trained on large datasets to identify scenes and object

< **Using the Console Program on Jetson >**
1. Open Terminal
2. $ cd jetson-inference/build/aarch64/bin
3. $ ./imagenet-console.py –-network=googlenet images/orange_0.jpg output_0.jpg
4. $ ./imagenet-console.py images/granny_smith_1.jpg output_1.jpg

ImageNet

Network	CLI argument	NetworkType enum
AlexNet	`alexnet`	`ALEXNET`
GoogleNet	`googlenet`	`GOOGLENET`
GoogleNet-12	`googlenet-12`	`GOOGLENET_12`
ResNet-18	`resnet-18`	`RESNET_18`
ResNet-50	`resnet-50`	`RESNET_50`
ResNet-101	`resnet-101`	`RESNET_101`
ResNet-152	`resnet-152`	`RESNET_152`
VGG-16	`vgg-16`	`VGG-16`
VGG-19	`vgg-19`	`VGG-19`
Inception-v4	`inception-v4`	`INCEPTION_V4`

note: to download additional networks, run the Model Downloader tool

```
$ cd jetson-inference/tools
$ ./download-models.sh
```

ImageNet

< Imagenet-console을 이용한 network 이해 >
=> 100%에 가장 근접한 개, 고양이 등등 찾기

이미지	Network	Network	Network	Network	Network
	확률	확률	확률	확률	확률
개					
	(%)				
고양이					
사과					
배					
오렌지					
사람					

ImageNet

2. Coding your own image recognition program

1. 파일 만들기, data 가져오기

```
$ cd ~/
$ mkdir my-recognition-python
$ cd my-recognition-python
$ touch my-recognition.py
$ chmod +x my-recognition.py
$ wget https://github.com/dusty-nv/jetson-inference/raw/master/data/images/black_bear.jpg
$ wget https://github.com/dusty-nv/jetson-inference/raw/master/data/images/brown_bear.jpg
$ wget https://github.com/dusty-nv/jetson-inference/raw/master/data/images/polar_bear.jpg
```

: 빈 파일 만들기

ImageNet

2. source code

$ gedit my-recognition.py : edit 실행

```
#!/usr/bin/python

import jetson.inference
import jetson.utils

import argparse

# parse the command line
parser = argparse.ArgumentParser()
parser.add_argument("filename", type=str, help="filename of the image to process")
parser.add_argument("--network", type=str, default="googlenet", help="model to use, can be: googlenet, resnet-18, ect. (see --help for others)")
opt = parser.parse_args()

# load an image (into shared CPU/GPU memory)
img, width, height = jetson.utils.loadImageRGBA(opt.filename)

# load the recognition network
net = jetson.inference.imageNet(opt.network)

# classify the image
class_idx, confidence = net.Classify(img, width, height)

# find the object description
class_desc = net.GetClassDesc(class_idx)

# print out the result
print("image is recognized as '{:s}' (class #{:d}) with {:f}% confidence".format(class_desc, class_idx, confidence * 100))
```

; Shebang sequence 추가 for usging python interpreter

; Importing Modules

$ ./my-recognition.py my_image.jpg
$ ./my-recognition.py --network=resnet-18 my_image.jpg

; JPG, PNG, TGA, and BMP formats are supported.

; performs the inferencing with TensorRT.
=> 0 ~ 999 이미지 모델 리턴

; return the string containing the text decription

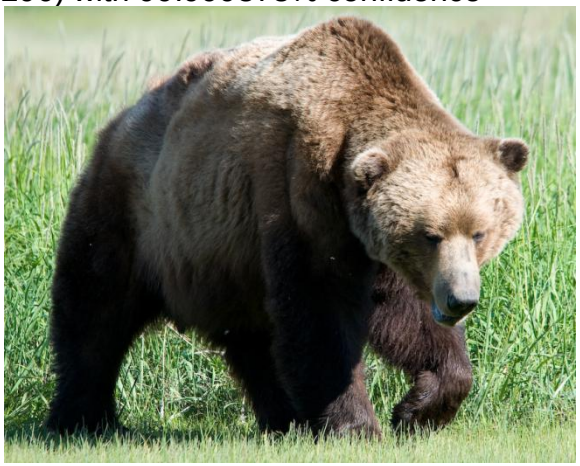
ImageNet

https://github.com/dusty-nv/jetson-inference/blob/master/data/networks/ilsvrc12_synset_words.txt

```
n02672831 accordion, piano accordion, squeeze box
n02676566 acoustic guitar
n02687172 aircraft carrier, carrier, flattop, attack aircraft carrier
n02690373 airliner
n02692877 airship, dirigible
n02699494 altar
n02701002 ambulance
n02704792 amphibian, amphibious vehicle
n02708093 analog clock
n02727426 apiary, bee house
n02730930 apron
n02747177 ashcan, trash can, garbage can, wastebin, ash bin, ash-bin, ashbin, dustbin, trash barrel, trash bin
n02749479 assault rifle, assault gun
n02769748 backpack, back pack, knapsack, packsack, rucksack, haversack
```

ImageNet

Example1)
$ ./my-recognition.py polar_bear.jpg
image is recognized as 'ice bear, polar bear, Ursus Maritimus, Thalarctos maritimus' (class #296) with 99.999878% confidence



Example2)
$ ./my-recognition brown_bear.jpg
image is recognized as 'brown bear, bruin, Ursus arctos' (class #294) with 99.928925% confidence

Example3)
$ ./my-recognition.py --network=resnet-18 polar_bear.jpg
image is recognized as 'ice bear, polar bear, Ursus Maritimus, Thalarctos maritimus' (class #296) with 99.743396% confidence

ImageNet

3. Running the Live Camera Recognition Demo

Realtime image recognition camera demo
=> imagenet-camera.py

```
$ ./imagenet-camera.py                          # using GoogleNet, default MIPI CSI camera (1280x720)
$ ./imagenet-camera.py --network=resnet-18      # using ResNet-18, default MIPI CSI camera (1280x720)
$ ./imagenet-camera.py --camera=/dev/video0     # using GoogleNet, V4L2 camera /dev/video0 (1280x720)
$ ./imagenet-camera.py --width=640 --height=480 # using GoogleNet, default MIPI CSI camera (640x480)
```

- `--network` flag setting the classification model (default is GoogleNet)
  - See Downloading Other Classification Models for the networks available to use.
- `--camera` flag setting the camera device to use
  - MIPI CSI cameras are used by specifying the sensor index (`0` or `1` , ect.)
  - V4L2 USB cameras are used by specifying their `/dev/video` node (`/dev/video0` , `/dev/video1` , ect.)
  - The default is to use MIPI CSI sensor 0 (`--camera=0`)
- `--width` and `--height` flags setting the camera resolution (default is `1280x720`)
  - The resolution should be set to a format that the camera supports.
  - Query the available formats with the following commands:

```
$ sudo apt-get install v4l-utils
$ v4l2-ctl --list-formats-ext
```

ILSVRC ImageNet dataset => data/networks/ilsvrc12_synset_words.txt

ImageNet

< Imagenet-camera를 이용한 network 특성 이해 >
=> 100%에 가장 근접한 개, 고양이 등등 찾기

이미지	Network	Network	Network	Network	Network
	확률	확률	확률	확률	확률
개					
	(%)				
고양이					
사과					
배					
오렌지					
사람					

THANK YOU

Suggestions Questions