Inference with pre-trained models

Intel Korea

Haeyoung Lee

순서

1. Intel Model Zoo 에서 사용하고자 하는 pre-trained model 을 고릅니다.

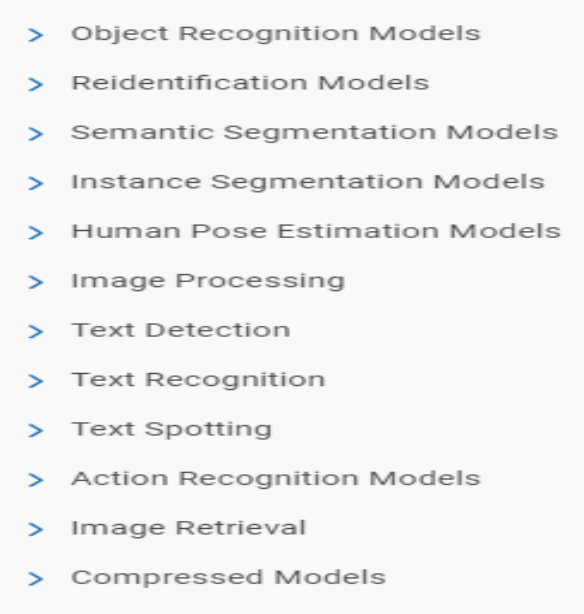
2. Model Downloader 을 통해서 pre-trained model 을 알맞은 형태로 다운로드 해 줍니다.

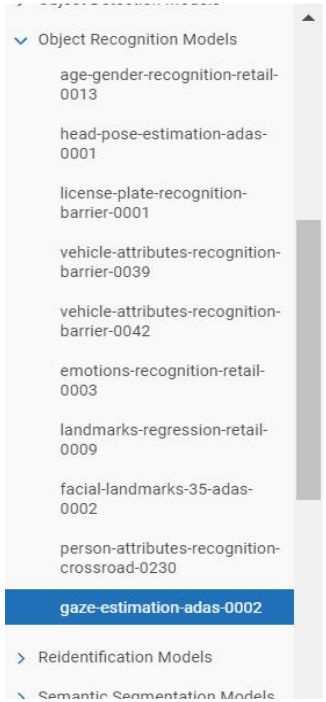
3. 샘플 추론 코드를 실행하여 다운받은 모델로 원하는 사진 또는 영상을 입력하여 추론해봅니다.

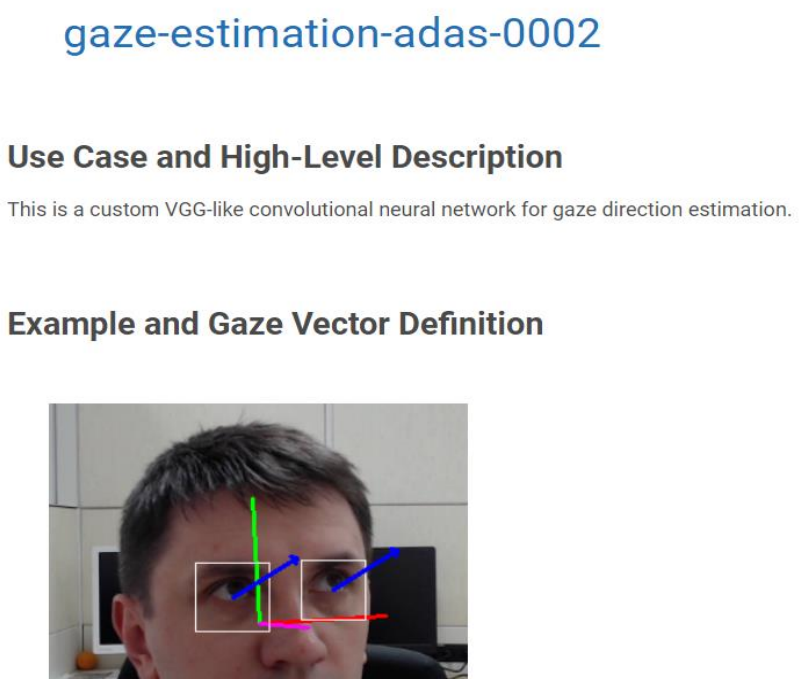
1. Pre-trained model 고르기

1. 먼저 사용하고자 하는 Pre-trained 모델을 골라야 합니다. 아래 링크에 들어가시면 다양한 Pretrained Models 의 상세한 설명을 볼 수 있으니 적절한 모델을 골라줍니다.

이번 자료에서는 gaze-estimation-adas-0002 를 골라 데모를 보여드리겠습니다.
https://docs.openvinotoolkit.org/latest/omz_models_intel_index.html





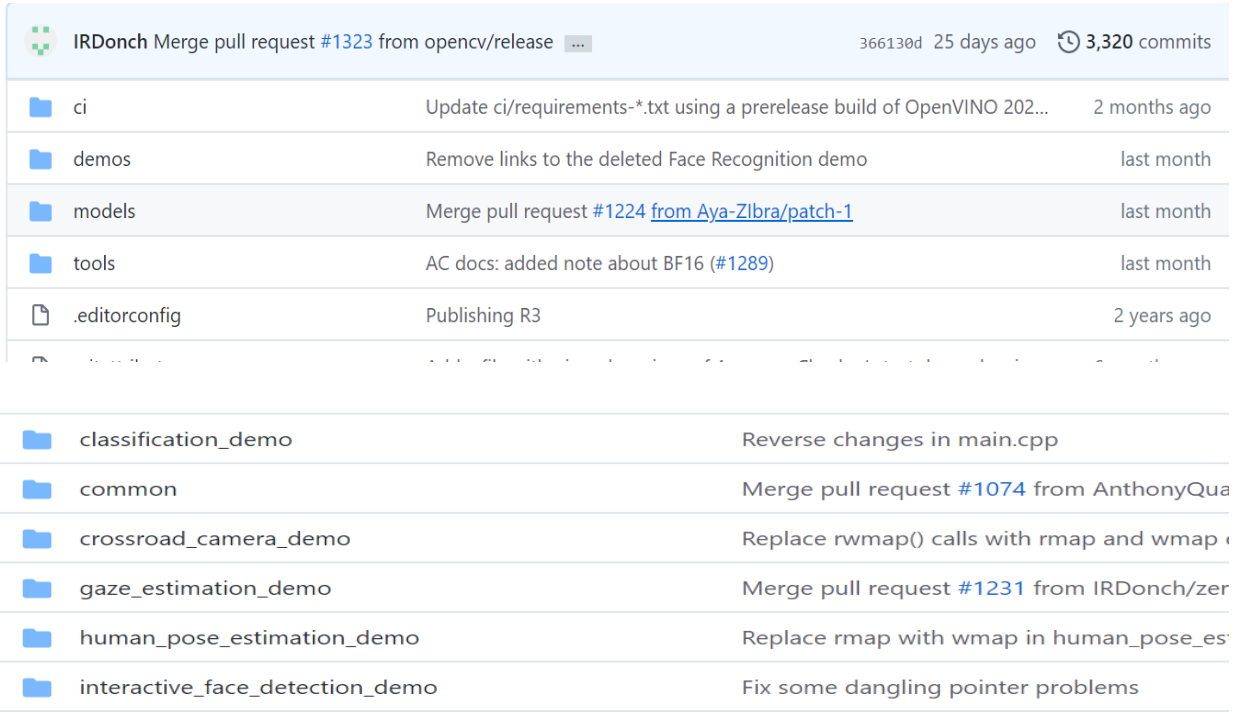


1. Pre-trained model 고르기

2. Intel 에서 제공하는 Open_model_zoo github 를 사용하셔도 좋습니다. 아래 링크에 들어가시면 다양한 데모를 read me 를 통해서 살펴볼 수 있습니다.

https://github.com/openvinotoolkit/open_model_zoo

위 링크의 다양한 데모 중 예시로 **Gaze estimation 데모**를 실행해보도록 하겠습니다.



위 링크에 접속하셔서 **demos** 를 클릭해주고,

gaze_estimation_demo 로 들어가줍니다.

1. Pre-trained model 고르기

README.md

Gaze Estimation Demo

This demo showcases the work of gaze estimation model. The corresponding pre-trained model `gaze-estimation-adas-0002` is delivered with the product.

The demo also relies on the following auxiliary networks:

- `face-detection-retail-0004` or `face-detection-adas-0001` detection networks for finding faces
- `head-pose-estimation-adas-0001`, which estimates head pose in Tait-Bryan angles, serving as an input for gaze estimation model
- `facial-landmarks-35-adas-0002`, which estimates coordinates of facial landmarks for detected faces. The keypoints at the corners of eyes are used to locate eyes regions required for the gaze estimation model
- `open-closed-eye-0001`, which estimates eyes state of detected faces.

For more information about the pre-trained models, refer to the model documentation.

Other demo objectives are:

- Video/Camera as inputs, via OpenCV*
- Visualization of gaze estimation results, and, optionally, results of inference on auxiliary models

3. 스크롤을 내리시면 README.md 의 파일을 보여주는데, 여기에 Gaze Estimation Demo 에 대한 상세한 설명이 있습니다.
이번 데모를 실행하기 위해서는 총 5가지 모델이 필요합니다.

face-detection-retail-0004

facial-landmarks-35-adas-0002

gaze-estimation-adas-0002

head-pose-estimation-adas-0001

open-closed-eye-0001

Model Downloader을 통해 위 5개의 모델 xml, bin 파일을 다운로드 해줍니다. (다음 슬라이드 참고)

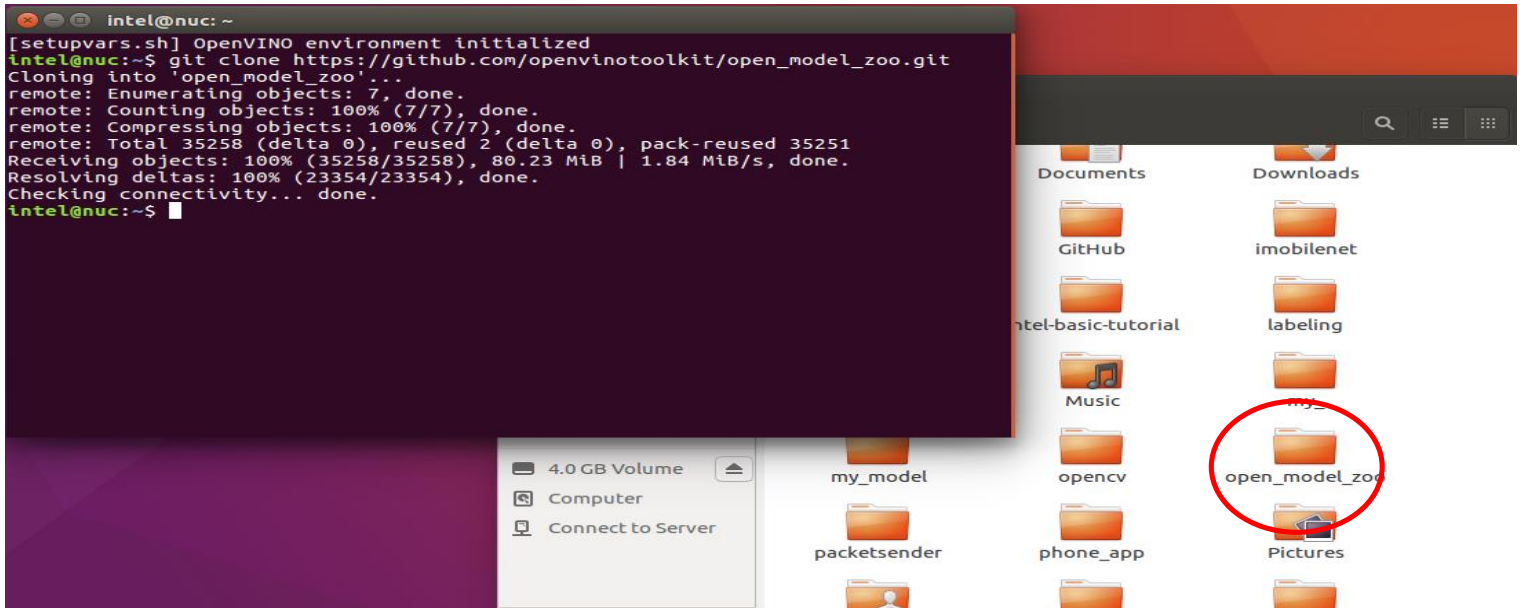
2. Model Downloader로 모델 다운로드하기

$ git clone https://github.com/openvinotoolkit/open_model_zoo.git

1. 위의 명령어를 입력해 필요한 코드들이 담겨있는 open_model_zoo 디렉토리를 다운로드 해줍니다.

   이 때 인터넷이 연결되어 있어야 합니다. (원래 OpenVINO 를 설치하면 OpenVINO 경로에 open_model_zoo 가 있지만 경로가 복잡하므로 git clone 을 통해 home 밑에 다운로드해줍니다.)

   다운로드가 완료되면 아래와 같이 home 디렉토리 아래 **open_model_zoo** 폴더를 확인할 수 있습니다.



2. Model Downloader로 모델 다운로드하기

2. 오른쪽과 같은 모델들을 다운로드하기 위해서는 OpenVINO 의 툴인 Model Downloader 를 사용해야 합니다.

- Model Downloader는 파이썬 파일로 존재하며, 위치는 아래 사진과 같이 **/home/intel/open_model_zoo/tools/downloader** 에 있습니다.

face-detection-retail-0004

facial-landmarks-35-adas-0002

gaze-estimation-adas-0002

head-pose-estimation-adas-0001

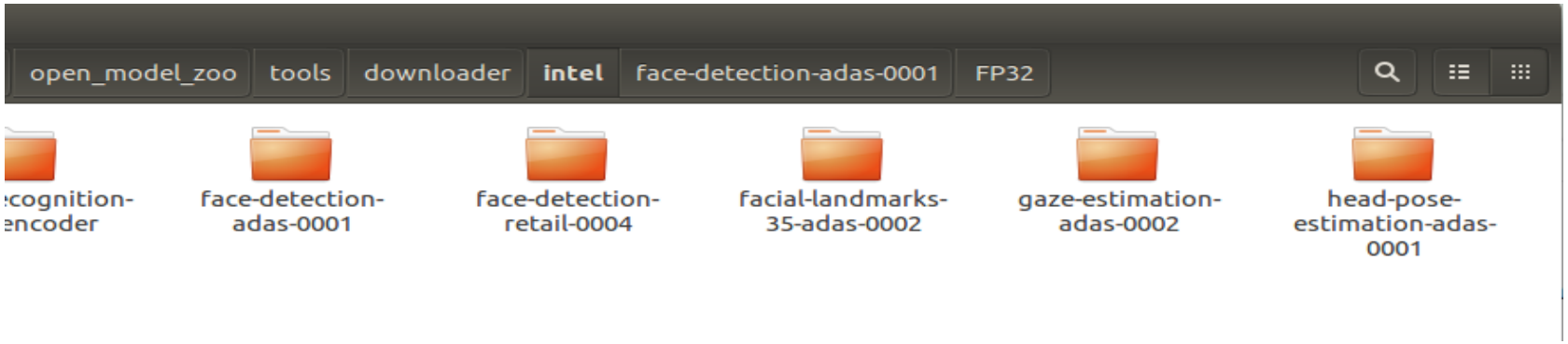
open-closed-eye-0001

3. 위의 경로로 들어가 downloader.py 를 실행하고, 명령을 입력할 때 '**--name**' 와 함께 원하는 모델을 넣어줍니다.

2. Model Downloader로 모델 다운로드하기

```
$ cd /home/intel/open_model_zoo/tools/downloader
$ sudo python3 downloader.py --name gaze-estimation-adas-0002    ( --name 뒤의 모델 이름 바꿔서 5개 모두 진행 )
```

4. 위와 같은 명령어를 입력해 앞의 5개의 모델을 전부 다운로드해줍니다.
   다운로드를 완료하면 아래의 그림과 같이 downloader 아래 intel 폴더 안에 모델들이 다운로드된 것을 확인할 수 있습니다.

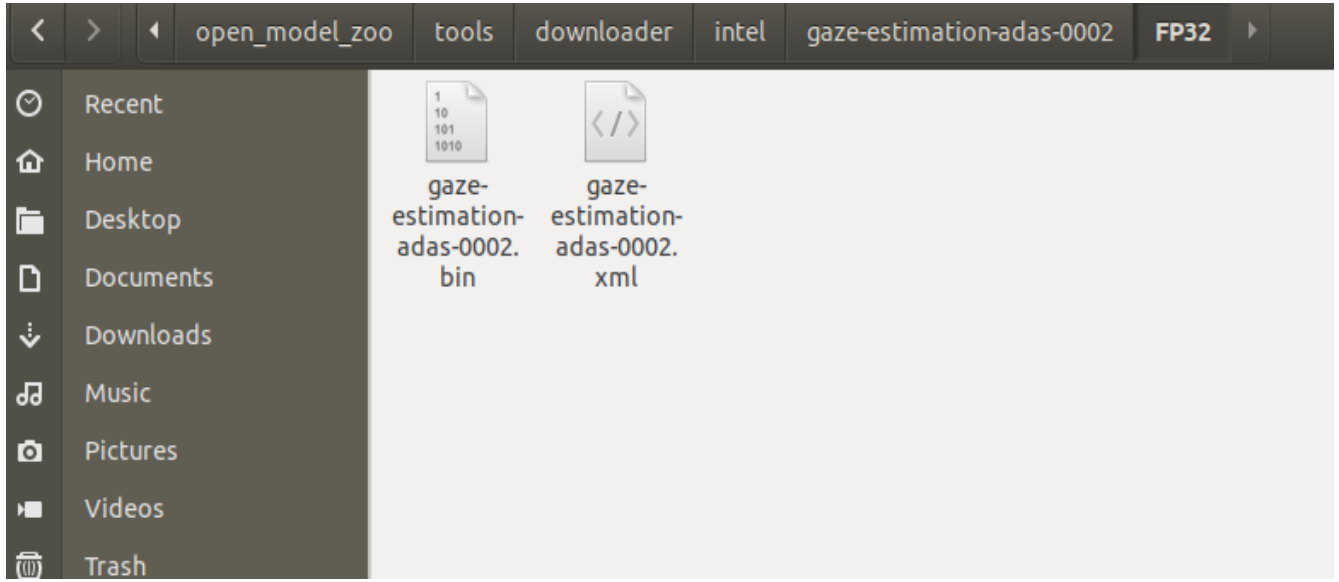


2. Model Downloader로 모델 다운로드하기

5. 다운로드가 완료된 모델을 클릭해보면 선택한 모델이 FP16, FP16-INT8, FP32 의 다양한 포맷으로 다운받아진 것을 확인할 수 있습니다.

6. 여기서 FP32 에 들어가 선택한 모델이 xml, bin 파일로 잘 변환된 것을 확인해줍니다.

(오픈비노를 이용해 추론하기 위해서는 Inference Engine이 인식할 수 있는 IR 파일로 변환이 되어야 합니다.)

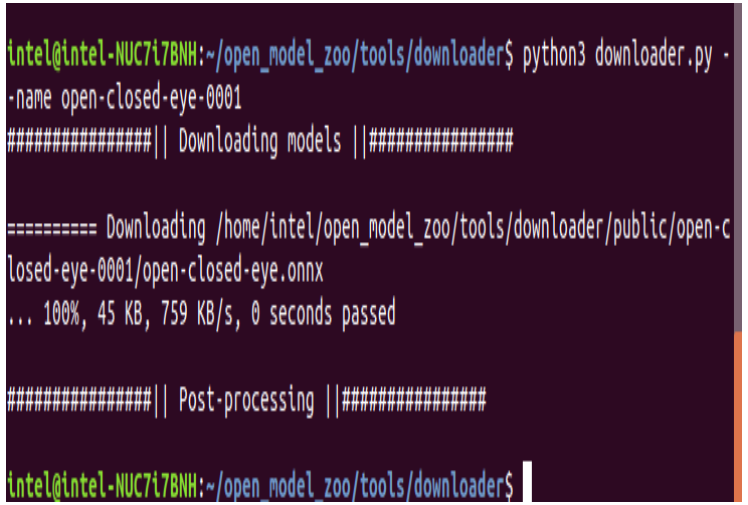


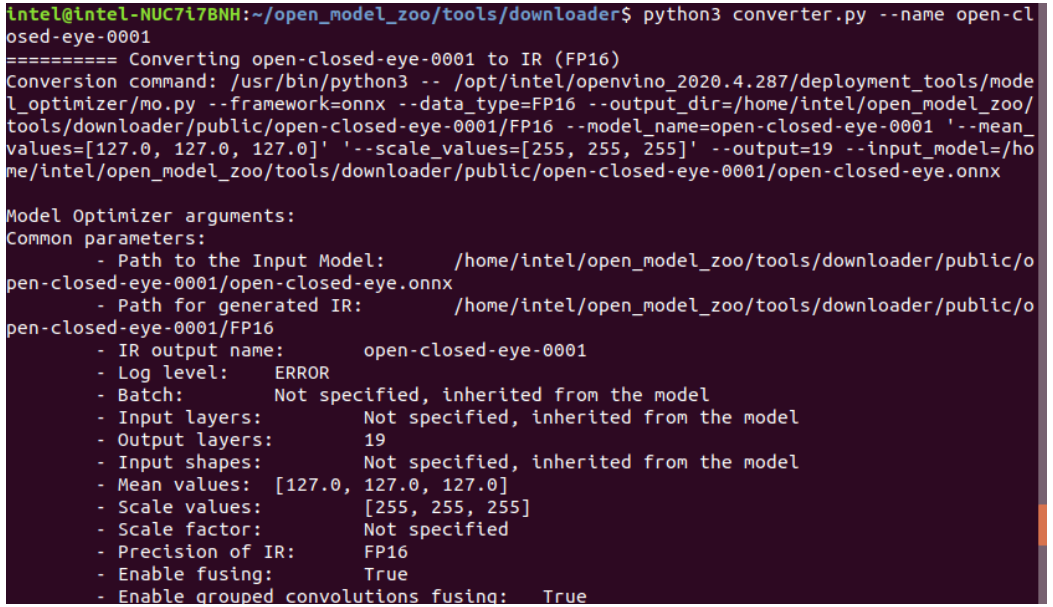
7. 다운받은 모델들의 경로를 간단하게 하기 위해 FP32 아래의 xml과 bin파일을 **home/intel/model** 로 **복사**해줍니다.

2. Model Downloader로 모델 다운로드하기

```
$ sudo python3 converter.py --name open-closed-eye-0001 --mo /opt/intel/openvino/deployment_tools/model_optimizer/mo.py
```

8. 5개의 모델 중 'open-closed-eye-0001' 모델은 다른 모델과 다르게 xml, bin 파일로 다운로드되지 않고 onnx 파일로 다운로드됩니다. 따라서 OpenVINO 추론을 위해서는 onnx 파일을 다시 xml, bin파일로 변환해주어야 합니다. 이 onnx 파일은 아래 다운 완료된 코드에서 확인할 수 있듯이 downloader 밑에 intel 이 아닌 public 에 저장되어 있습니다.

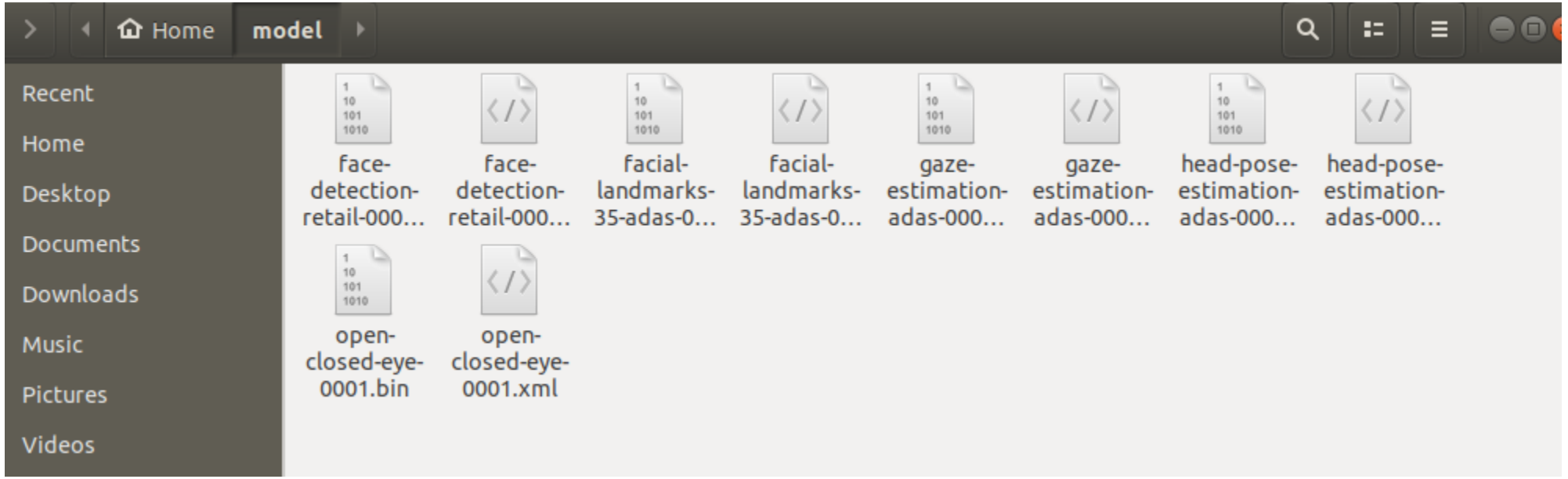




9. 위의 코드를 통해 onnx 파일을 xml, bin 파일로 변환해주면 public/open-closed-eye-0001 밑에 FP32 가 생성되고, FP32 밑에 open-closed-eye.xml 과 bin 파일을 /home/intel/model 로 복사해 줍니다.

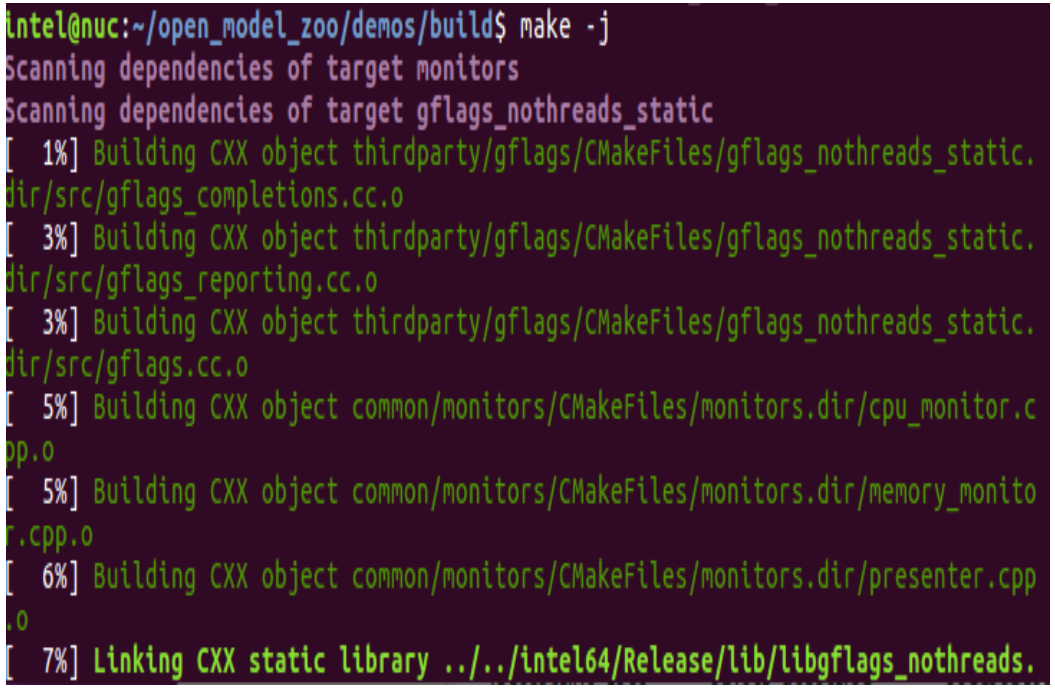
2. Model Downloader로 모델 다운로드하기

10. 아래와 같이 /home/intel/model 아래에 5개의 모델이 다 준비가 되었다면,
    모델 다운로드 단계는 완료되었습니다.
    이제 5개의 모델을 이용한 gaze estimation demo 를 실행하기 위해 준비해야합니다.



3. Demo 실행하기

```
$ cd open_model_zoo/demos
$ mkdir build
$ cd build
$ cmake ../
$ make -j   (에러 발생시 이 명령어 반복 실행)
```



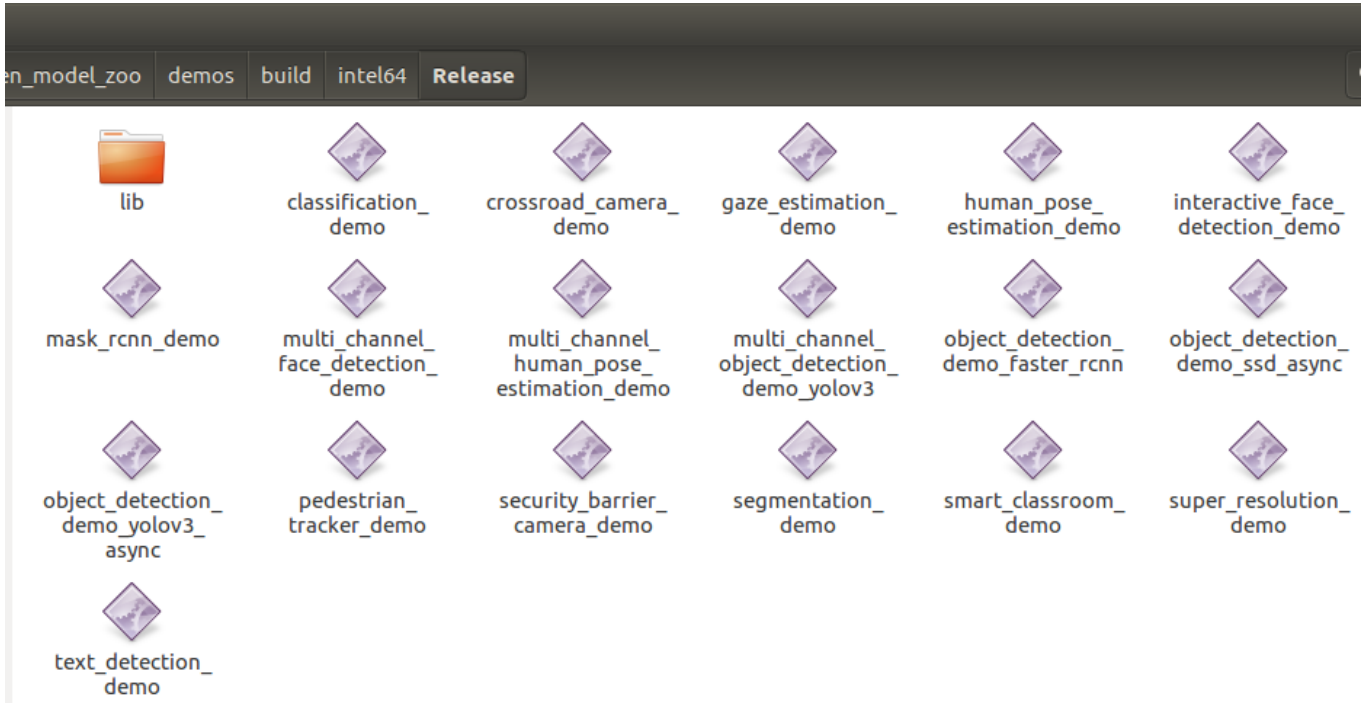
1. open_model_zoo 아래 demos 폴더에는 다양한 데모들이 준비되어 있습니다. 이러한 데모들 중 c 로 개발된 데모를 실행하기 위해서는 먼저 데모파일들을 빌드해서 실행가능하도록 해주어야 합니다.

- 위의 명령어들을 하나씩 입력해서 데모를 빌드해줍니다.
"**make –j**" 를 통해 데모를 하나씩 빌드해줄 수 있는데, 여기서 빌드가 완료된 데모 실행 파일은 **/home/intel/open_model_zoo/demos/build/intel64/Release** 에 저장됩니다.
- 이 때 make –j 의 명령에서 에러가 발생할 수 있는데, 에러가 발생하면 다시 한번 "**make –j**" 명령어를 입력해주고, **100% 다운로드가 완료될 때까지 반복해서 입력**해줍니다.

3. Demo 실행하기

2. 빌드가 100% 완료되면 아래와 같이 **/home/intel/open_model_zoo/demos/build/intel64/Release** 아래 다양한 실행파일들이 빌드된 것을 확인할 수 있습니다.

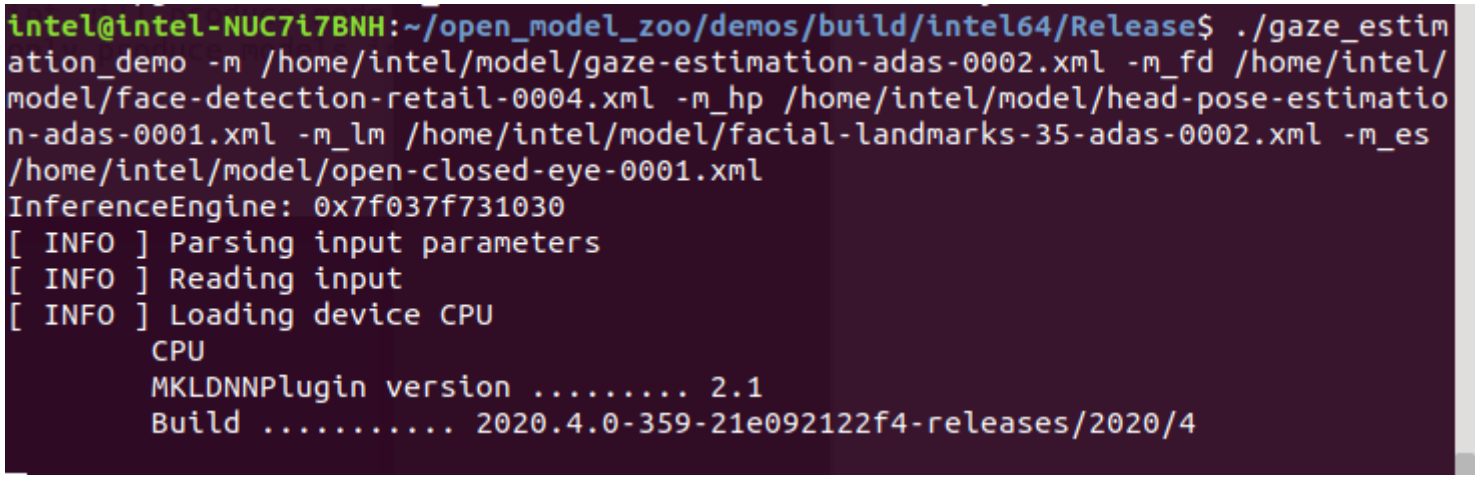
```
[ 85%] Linking CXX executable ../intel64/Release/mask_rcnn_demo
[ 85%] Built target mask_rcnn_demo
[ 86%] Linking CXX executable ../intel64/Release/object_detection_demo_faster_rc
nn
[ 86%] Built target object_detection_demo_faster_rcnn
[ 87%] Linking CXX executable ../intel64/Release/pedestrian_tracker_demo
[ 94%] Built target pedestrian_tracker_demo
[ 95%] Linking CXX executable ../intel64/Release/object_detection_demo_yolov3_as
ync
[ 95%] Built target object_detection_demo_yolov3_async
[ 96%] Linking CXX executable ../../intel64/Release/multi_channel_face_detection
_demo
[ 97%] Linking CXX executable ../../intel64/Release/multi_channel_human_pose_est
imation_demo
[ 97%] Built target multi_channel_face_detection_demo
[ 97%] Built target multi_channel_human_pose_estimation_demo
[ 98%] Linking CXX executable ../intel64/Release/security_barrier_camera_demo
[ 98%] Built target security_barrier_camera_demo
[100%] Linking CXX executable ../../intel64/Release/multi_channel_object_detecti
on_demo_yolov3
[100%] Built target multi_channel_object_detection_demo_yolov3
Scanning dependencies of target ie_samples
[100%] Built target ie_samples
intel@nuc:~/open_model_zoo/demos/build$
```



3. Demo 실행하기

```
$ cd intel64/Release
$ ./gaze_estimation_demo -m /home/intel/model/gaze-estimation-adas-0002.xml -m_fd /home/intel/model/face-detection-retail-0004.xml -m_hp /home/intel/model/head-pose-estimation-adas-0001.xml -m_lm /home/intel/model/facial-landmarks-35-adas-0002.xml -m_es /home/intel/model/open-closed-eye-0001.xml
```

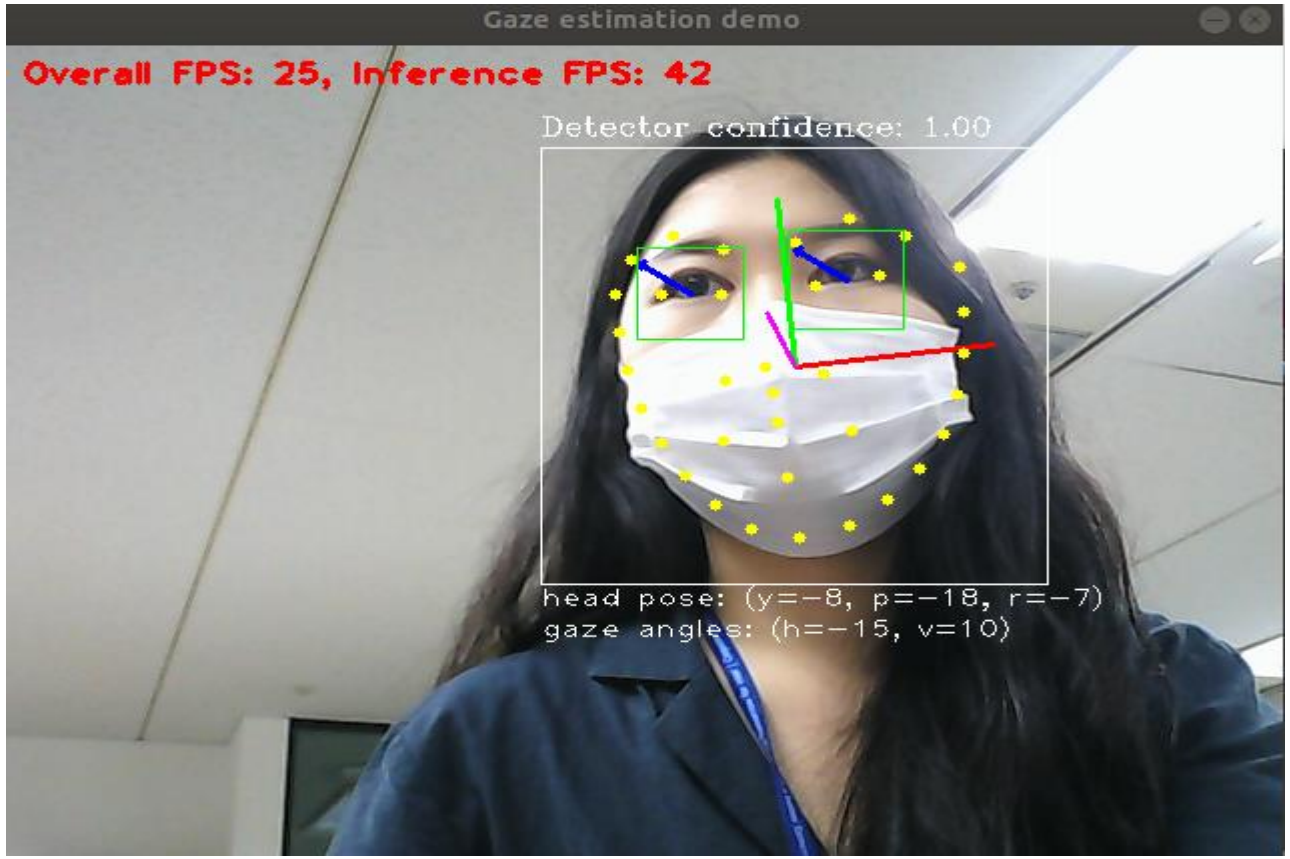
3. 이제 Gaze estimation DEMO 실행을 위해 위의 명령어를 입력해줍니다.



3. Demo 실행하기

4. 아래와 같이 gaze estimation DEMO 가 잘 실행됩니다.

이 때 오른쪽 아래 사진과 같이 해당 키를 누르면 다양한 옵션이 on / off 됩니다. (A 누르면 전체 on, N 누르면 전체 off 등)




```
80 The demo allows you to control what information is displaye
81 The following keys are supported:
82 * G - to toggle displaying gaze vector
83 * B - to toggle displaying face detector bounding boxes
84 * O - to toggle displaying head pose information
85 * L - to toggle displaying facial landmarks
86 * E - to toggle displaying eyes state
87 * A - to switch on displaying all inference results
88 * N - to switch off displaying all inference results
89 * F - to flip frames horizontally
90 * Esc - to quit the demo
91
```

* BERT Demo 실행

```
$ cd /home/intel/open_model_zoo/tools/downloader
$ python3 downloader.py --name bert-small-uncased-whole-word-masking-squad-0001
```

- BERT demo 는 question-answering demo 로, 질문을 입력하면 대답을 해주는 데모입니다.
- 먼저 input 으로 질문을 할 내용이 담긴 웹사이트 주소와 모델을 주고, 질문을 입력하면 대답을 해줍니다.

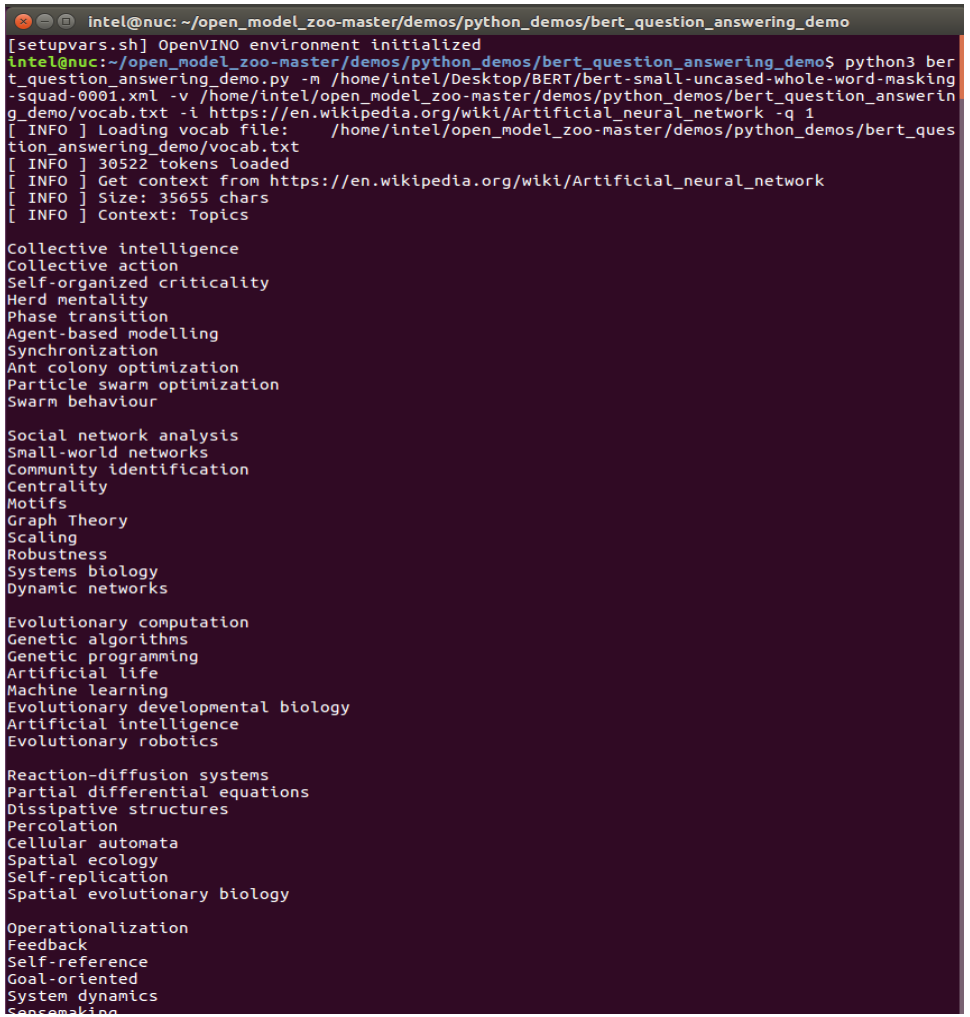
1. 앞의 방법과 동일하게 이번 'Bert question-answering demo' 를 위해서 'bert-small-uncased-whole-word-masking-squad-0001' 모델을 다운로드해주고, 다운로드 완료된 FP32 밑의 xml과 bin 파일을 /home/intel/model/ 의 경로에 복사해줍니다.

* BERT Demo 실행

```
$ cd /home/intel/open_model_zoo/demos/python_demos/bert_question_answering_demo
$ python3 bert_question_answering_demo.py -m /home/intel/model/bert-small-uncased-whole-word-masking-squad-0001.xml -v /home/intel/open_model_zoo/models/intel/bert-small-uncased-whole-word-masking-squad-0001/vocab.txt -i https://en.Wikipedia.org/wiki/Artificial_neural_network -q 1
```

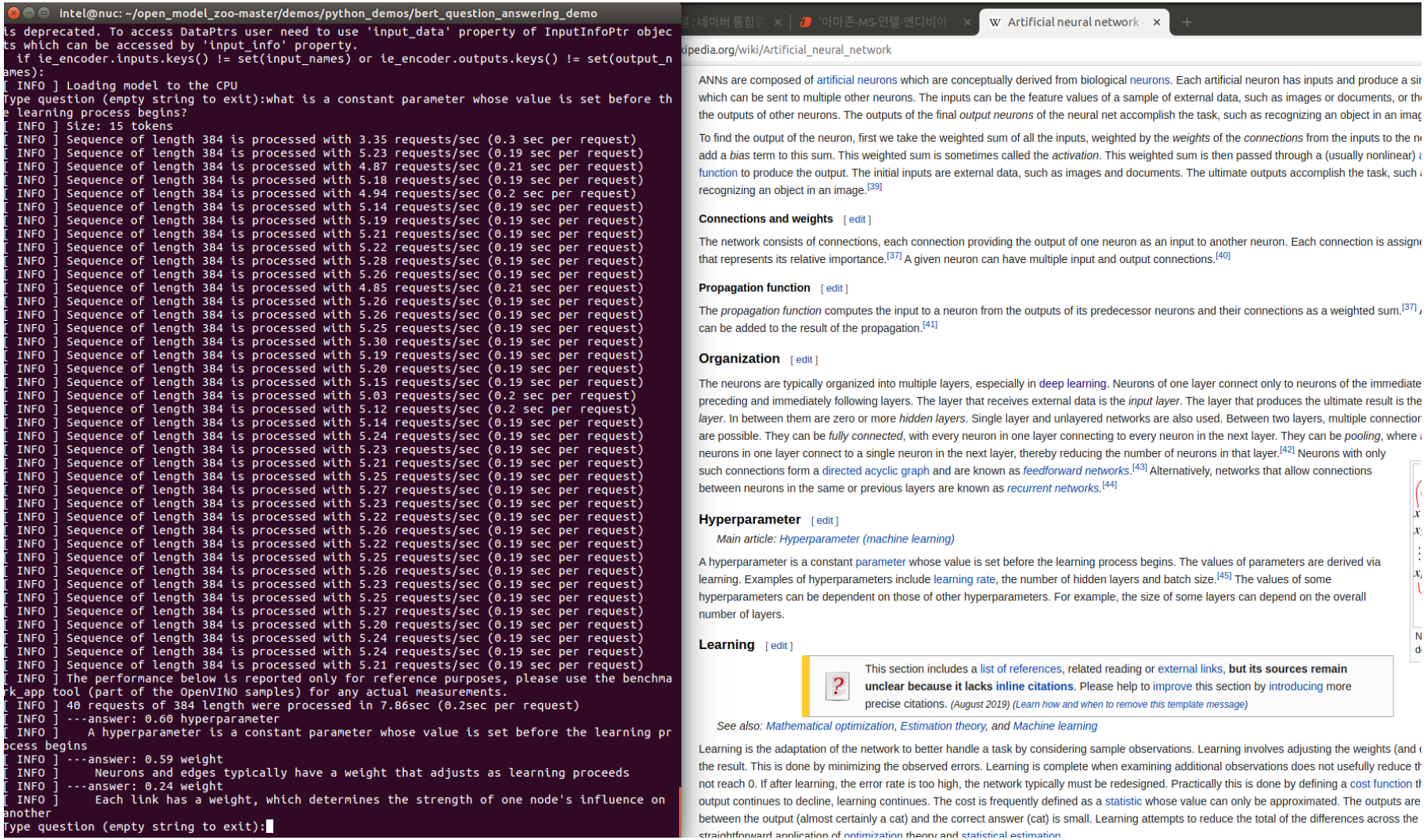
- 이 데모는 앞의 gaze_estimation_demo 와 달리 python 으로 개발된 데모입니다. 따라서 python_demo 에 가서 demo를 찾아 실행해줍니다.
- 또 input 으로 –v 가 필요한데, 이는 BERT 모델을 train 할 때 이용했던 단어 목록으로, /home/intel/open_model_zoo/models/intel/bert-small-uncased-whole-word-masking-squad-0001/vocab.txt 에서 확인할 수 있고, 인풋으로 입력해야합니다.

* BERT Demo 실행



- 데모를 실행하면 왼쪽과 같이 웹사이트 내용을 읽어오고, 질문을 입력하는 창이 뜹니다.
- 해당 내용에 맞는 질문을 입력하시면, 정답 확률과 함께 정답을 찾아주게 됩니다.

* BERT Demo 실행



- Hyperparameter 의 정의를 주고 무엇인지 물었을 때 답이 hyperparameter 라고 하는 예시입니다.

Thank you