## Set up the development environment

#### 1.1 install ANACONDA

Visit URL https://www.anaconda.com/distribution/#download-section

Select "Linux"-->"64-Bit (x86) Installer", download the installation script: Anaconda3-2020.11-Linux-x86\_64.sh(Just nee to download the latest version, no specific version requirements)

Run the script to install:

$ bash Anaconda3-2020.11-Linux-x86\_64.sh

When the installation page prompts "**Do you wish the installer to initialize Anaconda3 by running conda init?**", it is recommended, select "yes".

When the installation page displays "**Thank you for installing Anaconda3!**", the installation is successful.

Reopen the terminal or execute the following command to make the installation take effect immediately:

$source ~/.bashrc

#### 1.2 Qualcomm Neural processing SDK Setup

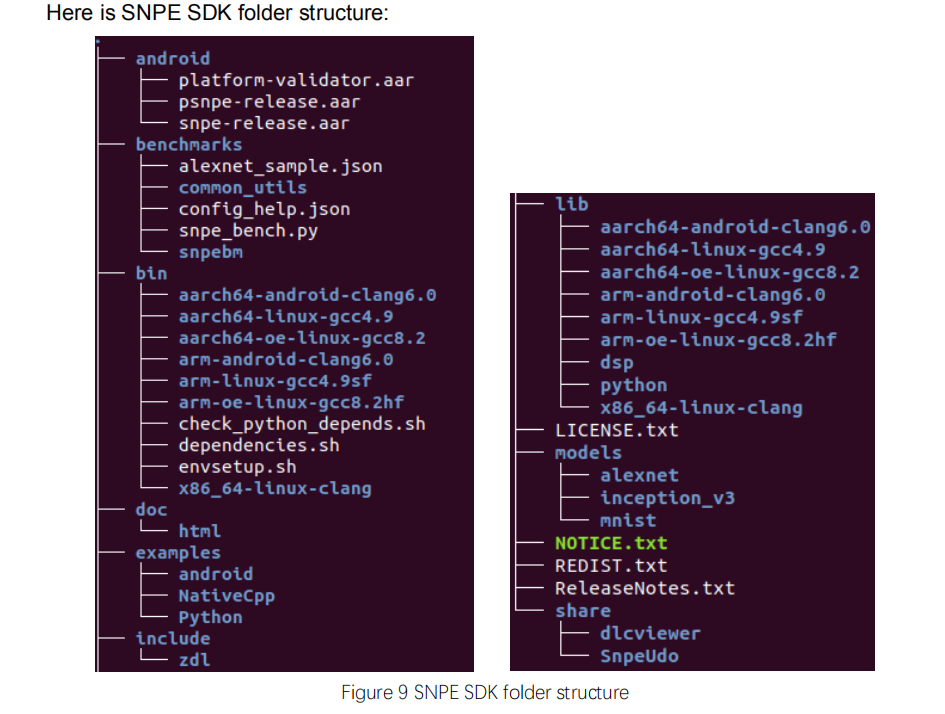
Visit Qualcomm website: (register, log in to Qualcomm account)

https://developer.qualcomm.com/software/qualcomm-neural-processing-sdk

In the ‘Neural Processing SDK for AI -> Tools & Resources’ page,

Select the Qualcomm Neural Processing SDK for AI v1.47. version to download to the local, and unzip the file

Qualcomm Neural processing SDK File structure:



**Configuration environment and dependencies:**

1. Run the dependencies script to check the system for Ubuntu package dependencies. It will ask to install ones that are missing. Install the missing packages.

$source snpe-X.Y.Z/bin/dependencies.sh

2. Run the python dependency checker to check the system for python package dependencies.

Install the missing packages.

$source snpe-X.Y.Z/bin/check\_python\_depends.sh

python package versions tested with the Qualcomm Neural processing SDK are:

numpy v1.16.5

sphinx v2.2.1

scipy v1.3.1

matplotlib v3.0.3

skimage v0.15.0

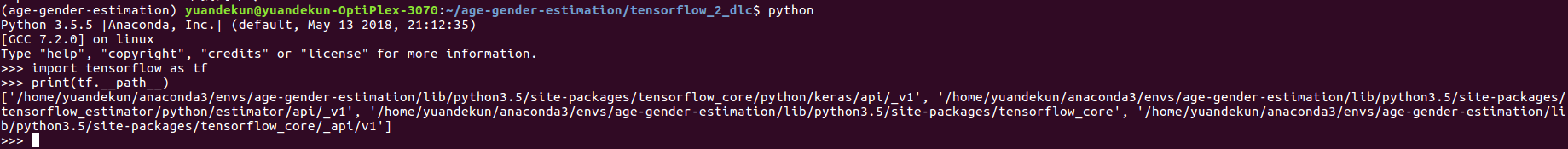
protobuf v3.6.0

pyyaml v5.1

3. Define $SNPE\_ROOT in .bashrc file.

export SNPE\_ROOT=/home/thundersoft/SNPE-envri/snpe-1.47.0

1. Configure the TensorFlow environment:
2. Set the $TENSORFLOW\_DIR



$TENSORFLOW\_DIR=/home/yuandekun/anaconda3/envs/age-gender-estimation/lib/python3.5/site-packages/tensorflow\_core

#### 2)Run script:

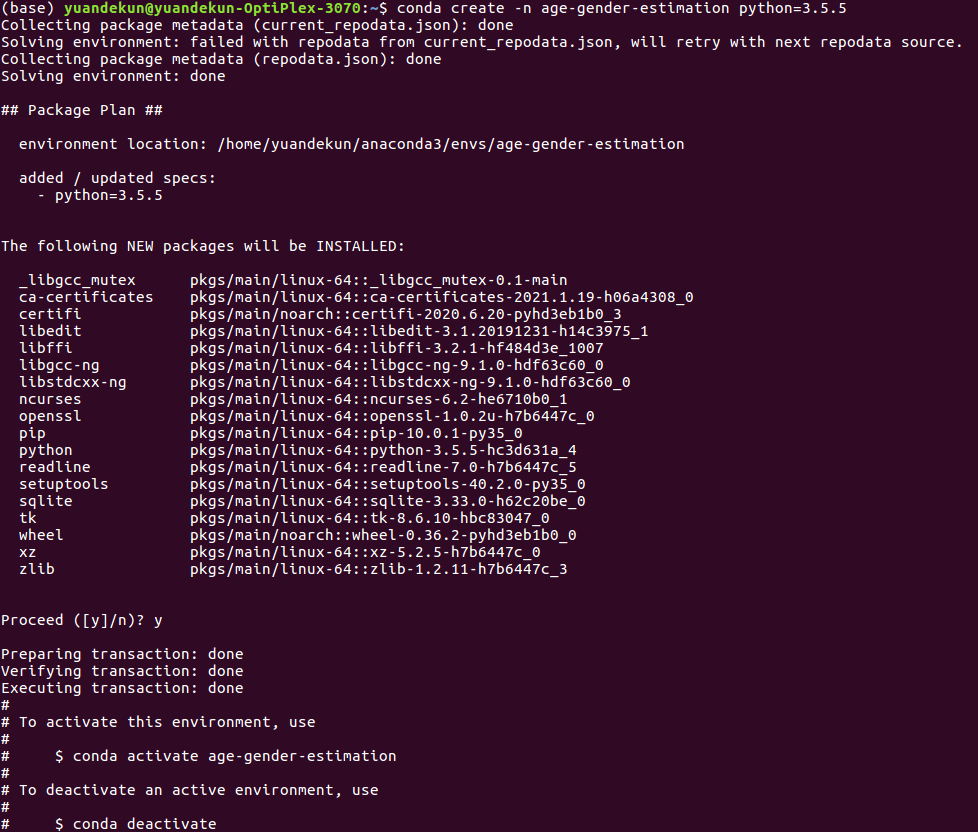
$ cd $SNPE\_ROOT

$ source bin/envsetup.sh -t $TENSORFLOW\_DIR

#### 1.3 Set up age-gender-estimation Development environment

1.Create a 3.5 virtual environment

$conda create -n age-gender-estimation python=3.5.5



1. Activate the virtual environment

conda activate age-gender-estimation

1. Download **age-gender-estimation** demo and h5 file from:

<https://github.com/yu4u/age-gender-estimation/releases/tag/v0.5>

1. Install some dependent librarys and tools

pip install --upgrade pip

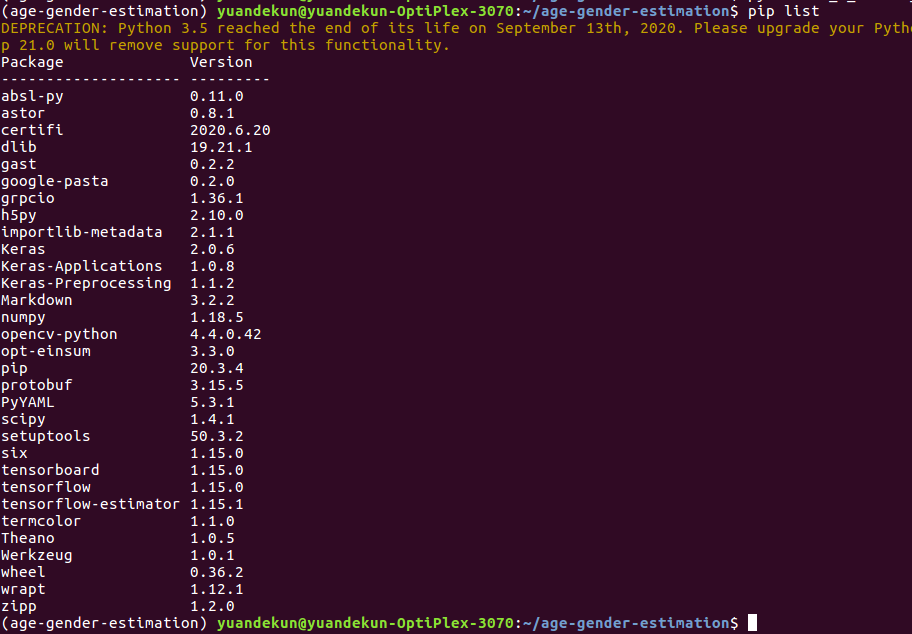
pip install opencv-python

pip install dlib

pip install Keras==2.0.6

pip install tensorflow==1.15

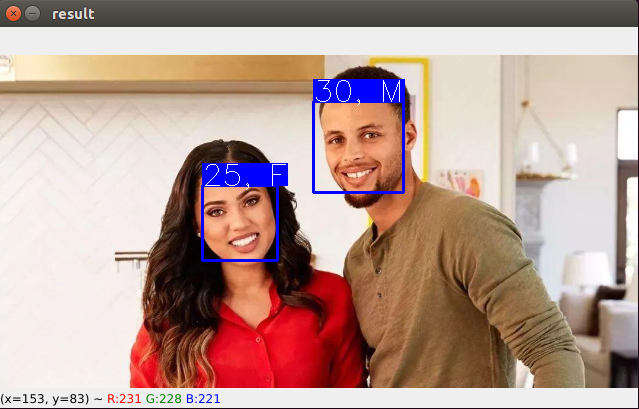
$pip list



## Show age-gender-estimation

$cd <project dir>  
$python demo\_show.py --weight\_file ./pretrained\_models/weights.28-3.73.hdf5

Result:

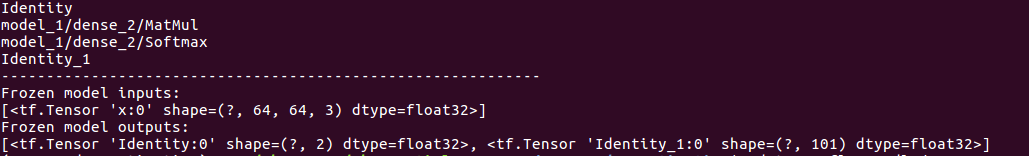


## Convert to DLC file

#### Convert hdfs to pb file

$python h5\_2\_frozen\_pb.py ./pretrained\_models/weights.28-3.73.hdf5

Getting Frozen model layers, input and output(see <project>/tensorflow\_2\_dlc/Frozen\_model\_layers)



#### Convert pb to DLC

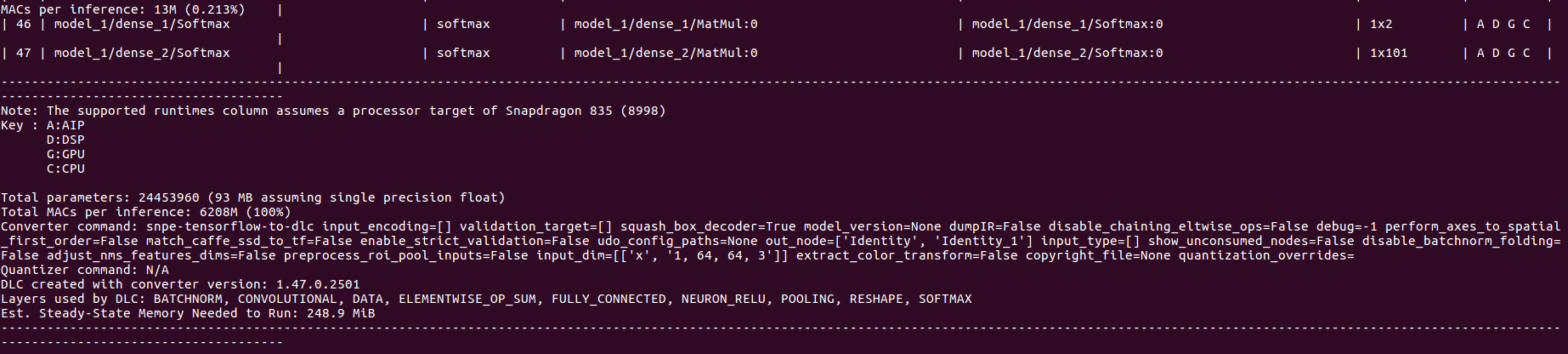
$snpe-tensorflow-to-dlc --input\_network weights.28-3.73.hdf5.pb --input\_dim x '1, 64, 64, 3' --out\_node Identity --out\_node Identity\_1 --output\_path age\_gender\_estimation.dlc

#### Check DLC

1. Run snpe-dlc-info to check input, output layer

$snpe-dlc-info -i age\_gender\_estimation.dlc

See <project dir>/tensorflow\_2\_dlc/age\_gender\_estimation.dlc.info for detail



1. Run snpe-dlc-viewer to check input, output layer

snpe-dlc-viewer -i age\_gender\_estimation.dlc

