ML Demos User Manual
iMX8XML Reference Design

Version	0.2
Status	Draft
Date	14-Oct-2019

Contents

Figures

Tables

DOCUMENT DETAILS

➢ Document History

Version	Author		Reviewer		Approver	
	Name	Date (DD-MM-YYYY)	Name	Date (DD-MM-YYYY)	Name	Date (DD-MM-YYYY)
0.1	*Anil Patel*	*19-Apr-2019*	Prajose John	*19-Apr-2019*	Bhavin Patel	*19-Apr-2019*
0.2	*Anil Patel*	*15-Oct-2019*	Prajose John	*15-Oct-2019*	Bhavin Patel	*15-Oct-2019*

Version	Description Of Changes
0.1	*initial draft*
0.2	*Added EIQ support and demos*

Table 1: Documents History

➢ Definition, Acronyms and Abbreviations

Definition/Acronym/Abbreviation	Description
cd	Change directory
scp	Secure copy over the network
dfl	Default
Wi-Fi	Wireless fidelity
LTE	Long-Term Evolution
ML	Machine Learning
SVM	Support Vector Machine
CNN	Convolutional Neural Network
tf	Tensorflow

Table 2: Definition, Acronyms and Abbreviations

➢ References

No.	Document	Version	Remarks
1	Refer the User guide V2.2	2.2	Prod Release.

Table 3: References

Introduction

➢ Purpose of the document

- Purpose of this document is to use / understand / demonstrate Machine Learning Demos to run on iMX8ML_RD AIML firmware.

➢ About the System

- This system contains iMX8X reference design with multiple interfaces. This is used for Machine learning experience.

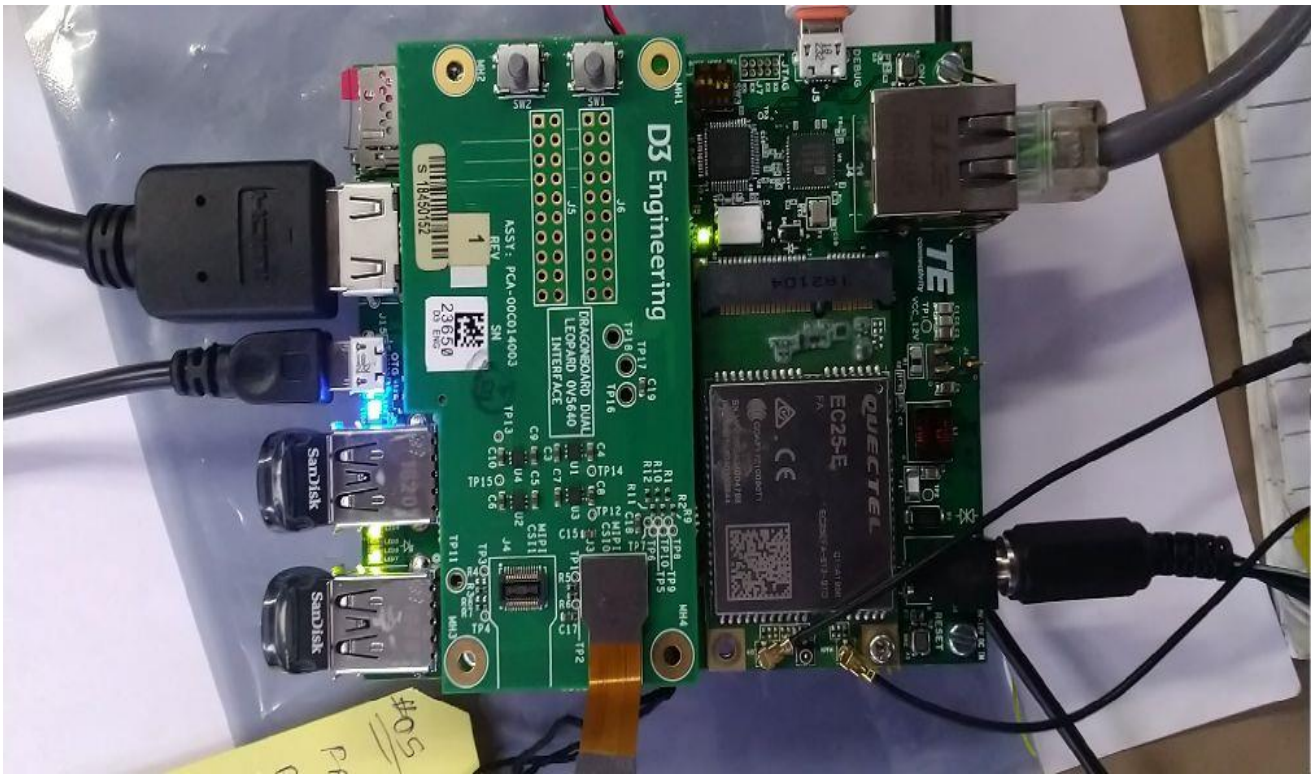

Figure 1: iMX8XML RD

➢ Pre-requisite

- x86 host system having Linux Ubuntu 16.04 LTS installed
- Basic understanding of Linux commands
- Flash the AIML firmware image to SD Card with all required python packages (Refer User Guide)
- Webcam or Mezzanine D3 Camera
- USB HUB / mouse / Keyboard
- HDMI Display with HDMI connector
- Ethernet or WiFi with Internet Connectivity (for Audio google API Demo Only)
- Open board's terminal- console (**minicom**) on x86 Host PC (Refer User Guide)

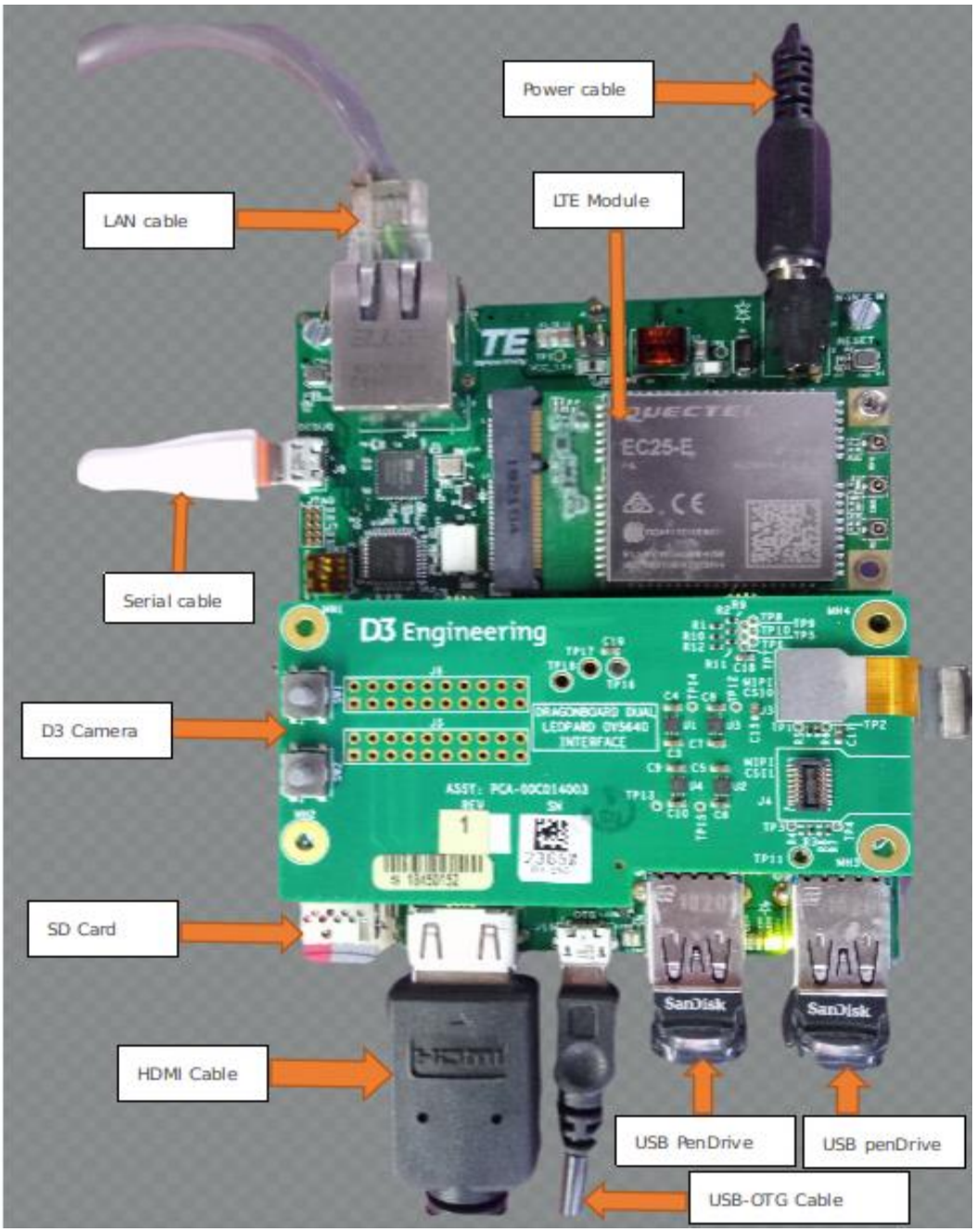

Figure 2: Hardware Setup

ML DEMOS BACKGROUND

To demonstrate board's capabilities for Machine Learning demos, we implemented few Audio and video related ML demos. These demos mainly depend on OpenCV, Tensorflow, Caffe, ARM NN and some python packages. All video ML demos required video source (webcam or D3 Mezzanine based OV5640 camera) to capture live stream and perform some action on it. Moreover, Audio demos capture audio from DMIC or any other USB mic and perform speech recognition on it.

All Demos are located on home folder of board under "ARROW_DEMOS" name.

➢ Copy Demos to SD Card

(If we have constraint of size of board and want to copy demos to USB or another partition then only follow this steps otherwise no need to do these steps.)

Our original firmware image took around 7GB space inside SD card. We can use rest of the space of sdcard as storage device for ML demos. For that, we need to create FAT or EXT4 partition. We recommended ext4 partition as it is default Linux file system. To do so follow below procedure.

- Flash SD card with required AIML firmware release. (Firmware release version must be BETA release 0.3 or above). (Refer User Guide for this.)
- In Linux HOST OS, open "**disk**" utility and see sdcard partitions in it. You can see image like below.

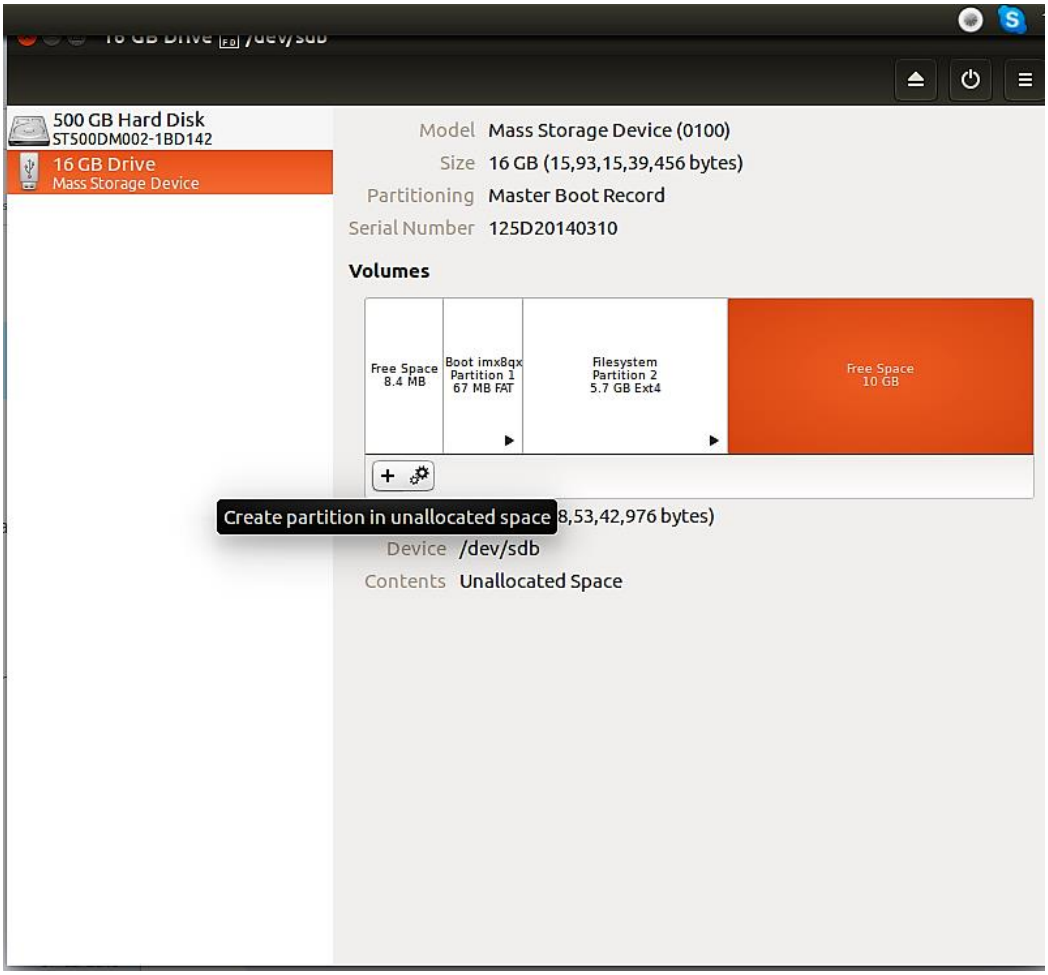

Figure 3: SD card partitions overview after flashing firmware

- As shown in above figure, in SD card partitions, we can see unused partition (10 GB) at last. We can utilized it.
- Now Click on “+” sign to create new partition.
- Please select file system ext4 and name it as shown below.

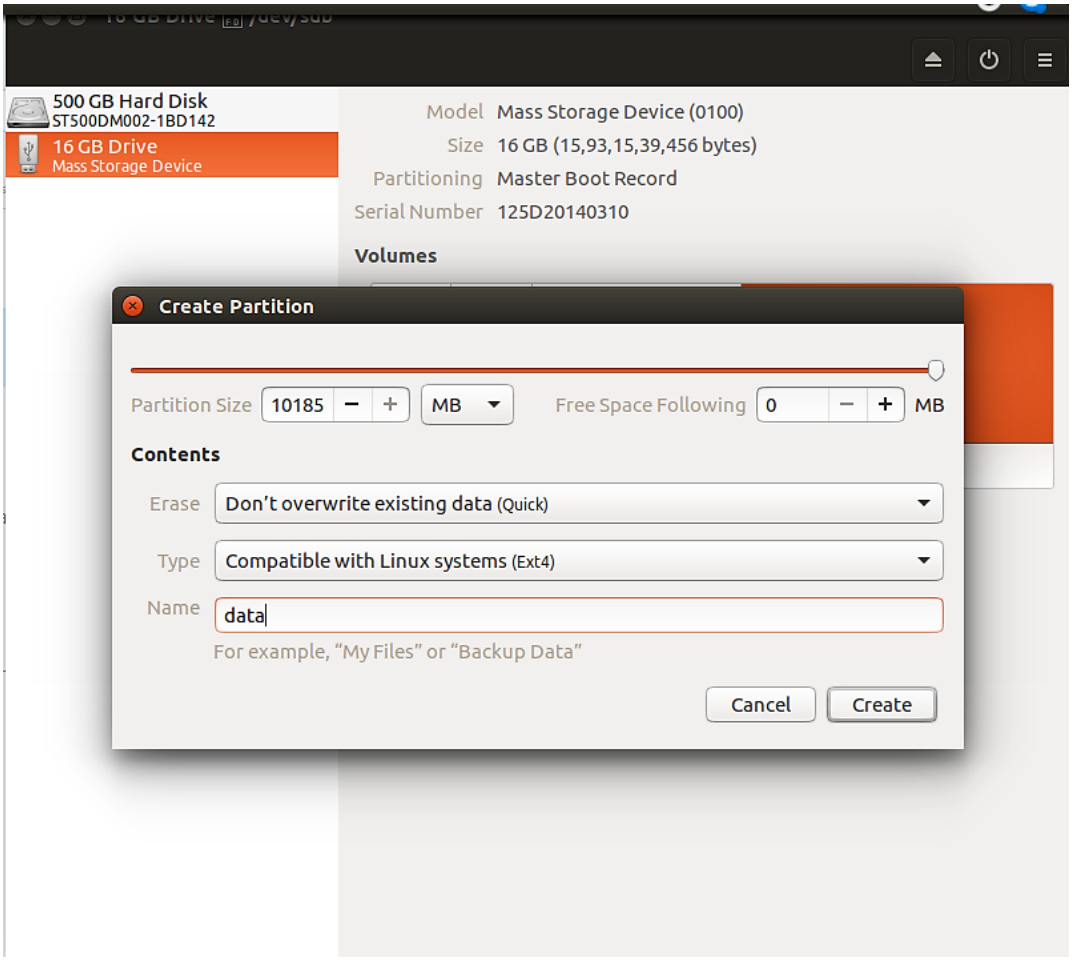

Figure 4: Create New EXT4 partition

- It will take few times and create partition and we can able to mount that partition. (See beow Figure for reference.)

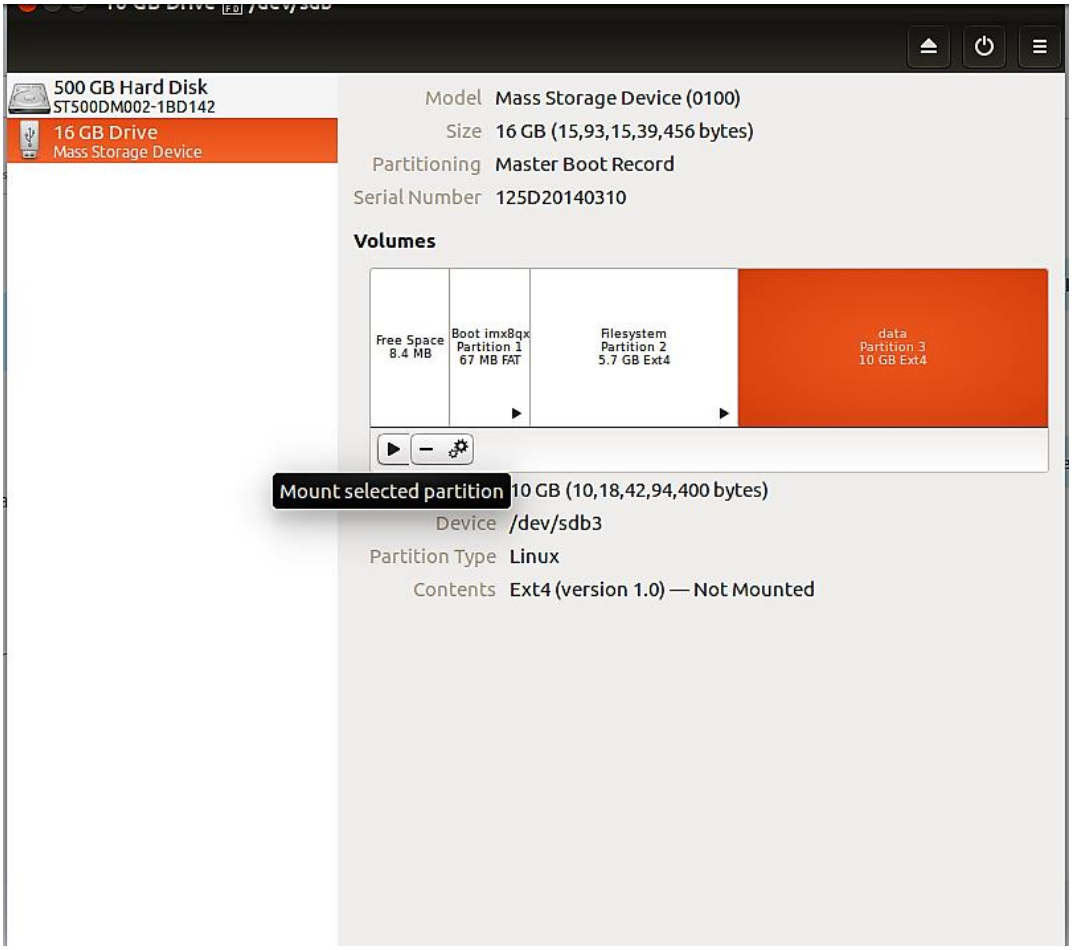

Figure 5: SD card partition after creating new one

- Copy ARROW DEMOS on this partition. If you are unable to do that then kindly unmount and mount again.
- After successful copy, boot our AIML board with this sdcard.
- After boot up we can see Demos at below location:

 # ls -la /run/media/mmcblk1p3/ARROW_DEMOS/

total 40

drwxrwxrwx 7 1000 tracing 4096 Apr 9 14:22 .

drwx------ 5 1000 tracing 4096 Apr 9 14:22 ..

drwxrwxrwx 4 1000 tracing 4096 Apr 1 13:07 ai-crowd_count

drwxrwxrwx 6 1000 tracing 4096 Apr 9 14:40 face_recognition

drwxrwxrwx 4 1000 tracing 4096 Apr 9 12:24 real-time-object-detection

-rwxrwxrwx 1 1000 tracing 9322 Apr 9 14:18 run_ml_demos.sh

drwxrwxrwx 2 1000 tracing 4096 Apr 12 11:12 speech_recognition_tensorflow

➢ Run Setup

All required Python packages for ML demos are already installed in AIML firmware image. Our demos run using Python3 so we have added package for Python3 and not for python. We have also provide support for Python and Python3 PIP Package. Through which we can add or remove any python package and remove dependencies for rebuilding firmware image each times.

To install any python3 or python package use command:

<pip3 or pip> install <PACKAGE NAME OR PACKAGE WHEEL NAME>

To remove any python3 or python package use command:

<pip3 or pip> uninstall <PACKAGE NAME>

Some python packages, i.e. tensorflow has no standard python wheel package for ARM AARCH64 platform so we need to cross compile from source and need to create one (wheel package) for board. We already did that and provide wheel packages at home folder to setup python module on our board. For that user need to run **setup_ml_demo.sh** script using below commands:

 # sh ~/setup_ml_demo.sh

This script takes approximately 15-20 minutes and will install all required python3 packages for ML demos so **it has to be run once before running all the ML demos**. We already provide required wheel package at home folder so this script do not required any internet connectivity for installing packages. However, apart from script if you want to install any package as describe above then you required clientless (No firewall) internet connectivity.

ML Demos not comes with default firmware images. This is because we do not want to increase size of original firmware image as it take much times to flash SD card. In addition, by separate release of ML demos we can remove dependencies of firmware image release. This lead us to improve our demos without affecting firmware packages if we don't have any dependencies of software or packages.

RUNNING ML DEMOS

To run ML Demos we have created run_ml_demo.sh shell script. This script ask for user preferences like demo type, camera types, camera node entry, desired MIC etc and based on that run ML demos.

In this section, we discuss how to run each demo. Details description of Demo is in next section.

1. Crowd Counting Demo

This is a demo application using Python, QT, and Tensorflow to be run on embedded devices for Crowd counting. In this demo, we count the heads/persons in the crowd. Therefore, it is useful in human flow monitoring or traffic control.

This demo run on either pre-captured Image mode or in Live Camera mode. In pre-captured image mode, we took few sample images and find head counts in those images. In live camera mode we capture live frame through webcam or D3 mezzanine camera and try to find head count from it. User can select any mode by clicking on GUI.

Pre-requisite:

- Webcam or D3 Mezzanine camera
- USB mouse
- HDMI Display having minimum 1080p resolution

Steps to run Demo:

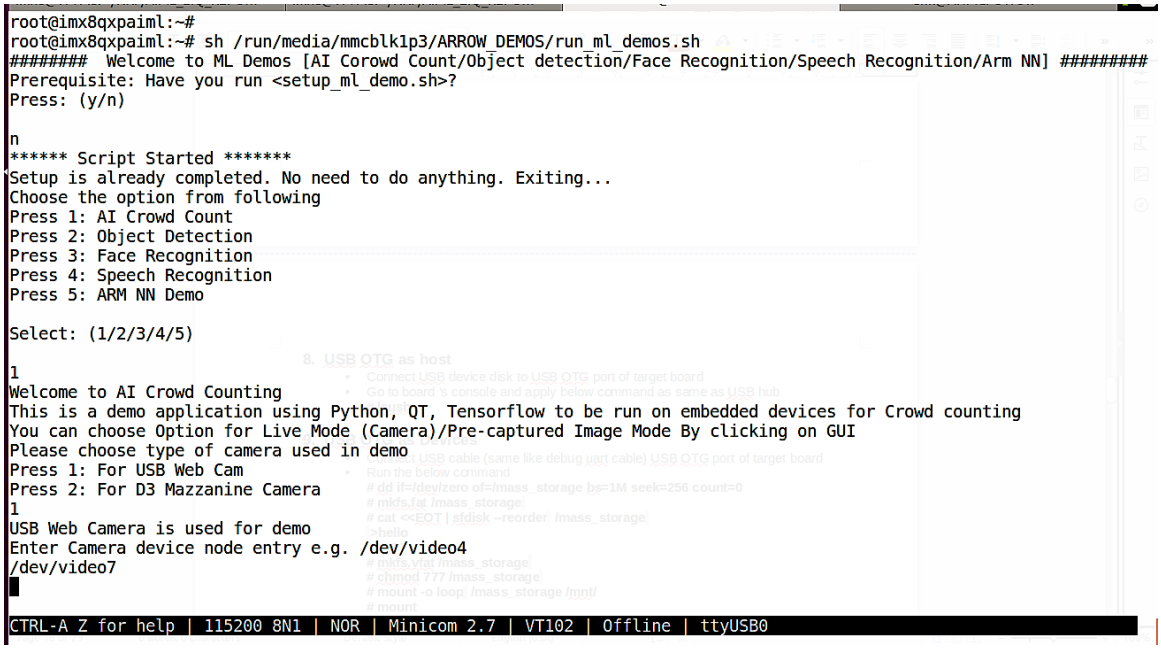

Figure 6: Run Crowd Count Demo

Run **/run/media/mmcblk1p3/ARROW_DEMOS/run_ml_demos.sh script** and select **option 1.**

See below full log to run demo, where user input is in **BOLD RED** fonts.

 # sh /run/media/mmcblk1p3/ARROW_DEMOS/run_ml_demos.sh

######### Welcome to ML Demos [AI Corowd Count/Object detection/Face Recognition/Speech Recognition/Arm NN] ##########

Prerequisite: Have you run <setup_ml_demo.sh>?

Press: (y/n)

n

****** Script Started *******

Setup is already completed. No need to do anything. Exiting...

Choose the option from following

Press 1: AI Crowd Count

Press 2: Object Detection

Press 3: Face Recognition

.

.

Select: (1/2/3/4/5)

1

Welcome to AI Crowd Counting

This is a demo application using Python, QT, Tensorflow to be run on embedded devices for Crowd counting

You can choose Option for Live Mode (Camera)/Pre-captured Image Mode By clicking on GUI

Please choose type of camera used in demo

Press 1: For USB Web Cam

Press 2: For D3 Mazzanine Camera

2

D3 Mazzanine Camera is used for demo

[130.962684] random: crng init done

[130.966100] random: 7 urandom warning(s) missed due to ratelimiting

ImportError: No module named 'numpy.core._multiarray_umath'

ImportError: No module named 'numpy.core._multiarray_umath'

NN> using tensorflow version: 1.3999999999999999

NN> using CPU-only (NO CUDA) with tensorflow

LOADING TENSORFLOW GRAPH

160x120 used for live mode as these typically are close

640x480 used for pre-captured large images of crowd as these are far

TF> using 640x480 resolution

If you run demo using **USB webcam** then last entry would be like:

Please choose type of camera used in demo

Press 1: For USB Web Cam

Press 2: For D3 Mazzanine Camera

1

USB Web Camera is used for demo

Enter Camera device node entry e.g. /dev/video4

/dev/video7

Here **"/dev/video7**" is webcam camera node by which we capture frame. User can check his/her node entry by plugging/unplugging webcam and see **which /dev/ node entry is removed/showed**.

By default demo run in pre-capture mode, and see output of headcount with inference time and date. Inference time is time taken to process one frame and finding headcount from it. Inference time is in Milli Seconds (ms). In pre-capture mode, Inference time is around 5000 to 6000 ms (5 to 6 sec) while in live mode, inference time is 450 to 550 ms. That's because we use smaller input image in live mode.

As shown in below image, on right side we show "**Density Maps**" for left side input image.

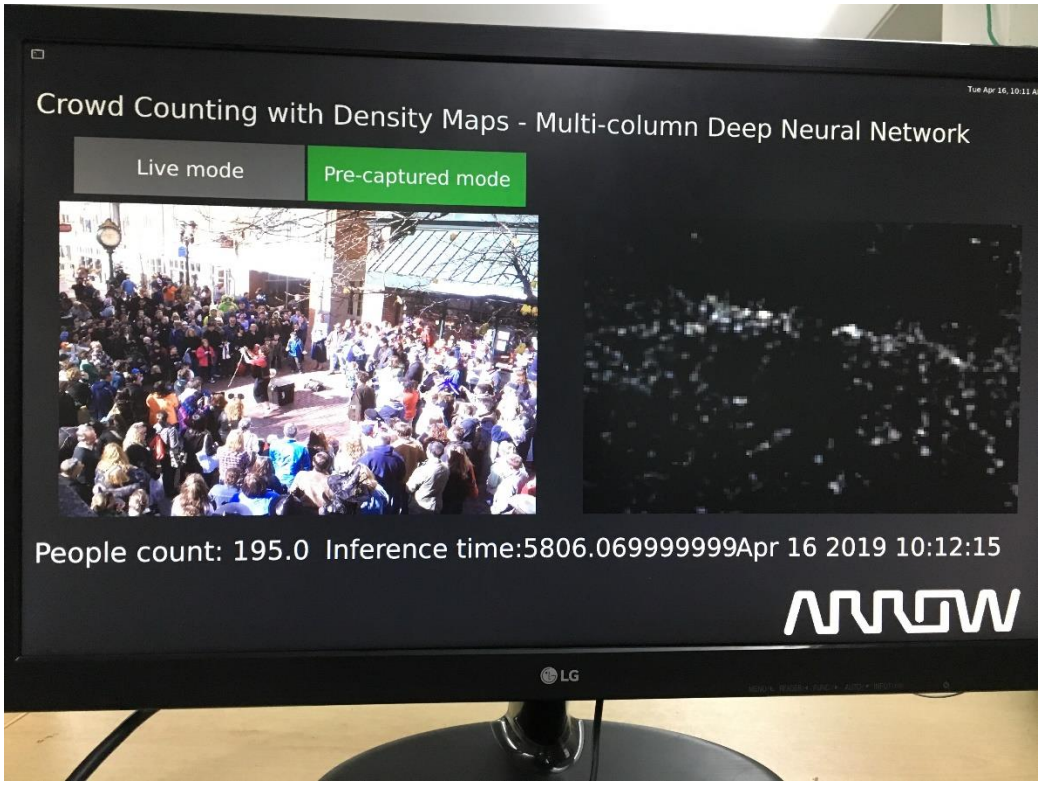

Figure 7: Crowd Count Pre-Captured Mode

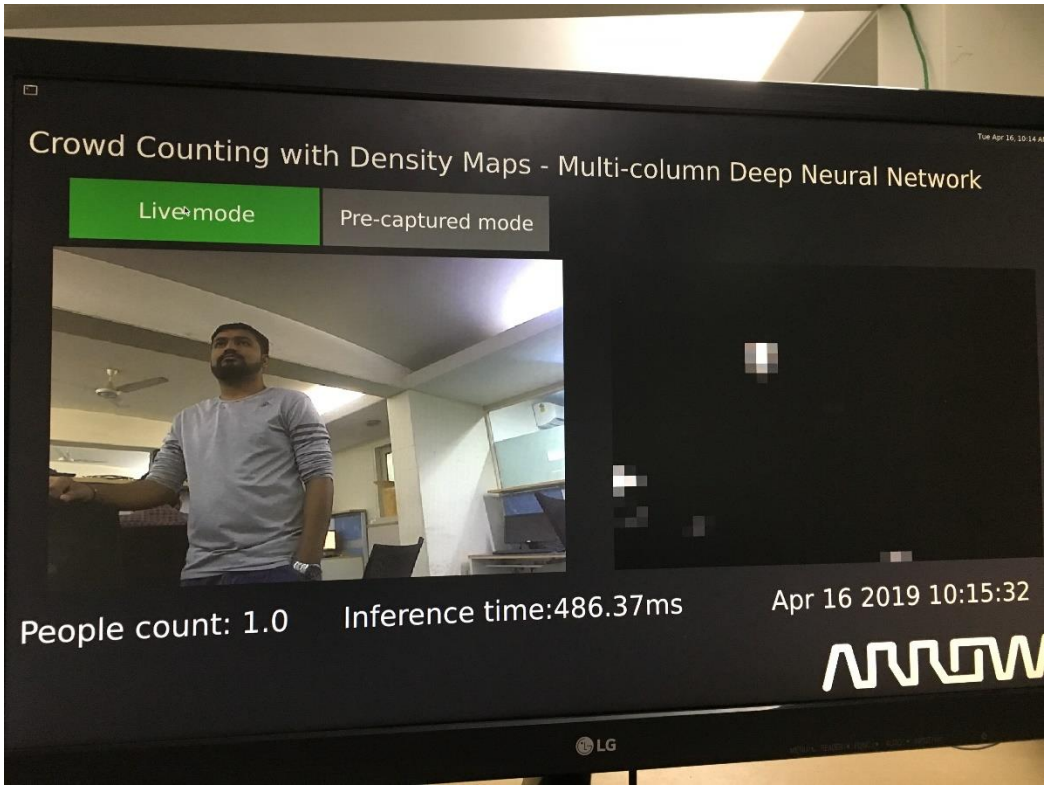

Figure 8: Crowd Count Live Mode

2. Object Detection Demo

In this Demo, we detect few objects like aeroplane, bicycle, bus, car, cat, cow, dog, horse, motorbike, person, sheep, train (objects necessary for self-driving cars.)

Here we have two version of Object detection. Both demo use same caffe based object detection model so accuracy remain same for both the demos. The only difference is in video output.

In Fast Object detection, we have smooth video. Here we create two python process. In one python process, we sample one frame at a time and done object detection on that frame. Another python process will use this object detection credentials and apply it on the entire frame it read from Camera. Thus, Camera output is smooth but object detection take 2 to 3 secs to give actual real-time output.

In Slow Object detection, we have single python process, which read camera frame first and do object detection on it. Therefore, here we do object detection on each frame and video output is choppy. However, here we got real-time object detection output. No delay. Due to that, this demo is perfect to identify board's capabilities.

Pre-requisite:

- Webcam or D3 Mezzanine camera
- USB mouse
- HDMI Display having minimum 1080p resolution
- Objects which we want to detect
- Object Images and another PC or laptop (in case of no real objects)

Steps to run Demo:

Run **/run/media/mmcblk1p3/ARROW_DEMOS/run_ml_demos.sh script** and select **option 2.**

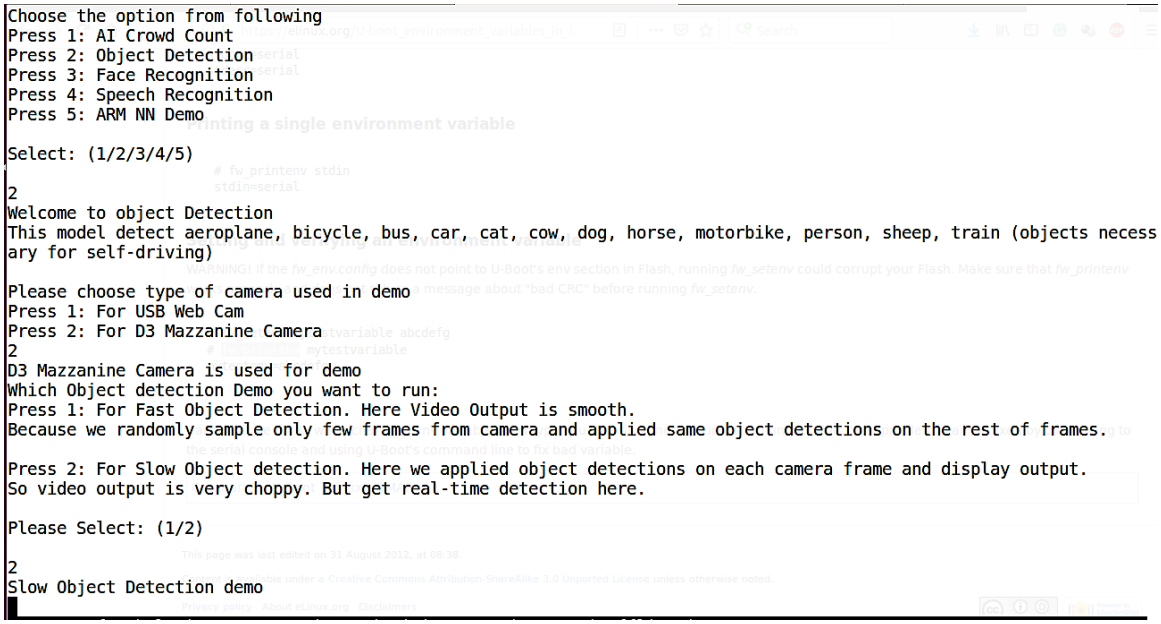

Figure 9: Run Object Detection Demo

See below full log to run demo, where user input is in **BOLD RED** fonts.

 # sh /run/media/mmcblk1p3/ARROW_DEMOS/run_ml_demos.sh

######### Welcome to ML Demos [AI Corowd Count/Object detection/Face Recognition/Speech Recognition/Arm NN] ##########

#

Prerequisite: Have you run <setup_ml_demo.sh>?

Press: (y/n)

y

Choose the option from following

Press 1: AI Crowd Count

Press 2: Object Detection

Press 3: Face Recognition

.

.

Select: (1/2/3/4/5)

2

Welcome to object Detection

This model detect aeroplane, bicycle, bus, car, cat, cow, dog, horse, motorbike, person, sheep, train (objects necess

ary for self-driving)

Please choose type of camera used in demo

Press 1: For USB Web Cam

Press 2: For D3 Mazzanine Camera

1

USB Web Camera is used for demo

Enter Camera device node entry e.g. /dev/video4

/dev/video7

Which Object detection Demo you want to run:

Press 1: For Fast Object Detection. Here Video Output is smooth.

Because we randomly sample only few frames from camera and applied same object detections on the rest of frames.

Press 2: For Slow Object detection. Here we applied object detections on each camera frame and display output.

So video output is very choppy. But get real-time detection here.

Please Select: (1/2)

2

Slow Object Detection demo

Loading model...

Starting video stream...

*(python3:3911): GStreamer-CRITICAL **: gst_element_get_state: assertion 'GST_IS_ELEMENT (element)' failed*

Using Wayland-EGL

Using the 'xdg-shell-v6' shell integration

Total Elapsed time: 15.52

Approx. FPS: 0.90

Exiting Demo...

Now in this demo, we need to provide object in front of camera to detect it. Person is the best real-time object for detection. Most of the object is easily available outside environment. However, to test model we don’t need actual object. We can simply provide good image of object instead of real object to verify our model. Input image must be provided with correct angle and exposure of light to detect objects. For that put input image inside PC or Laptop and set camera in front of it. For reference, we have attached below figure of setup. Also provided sample input image and its outputs.

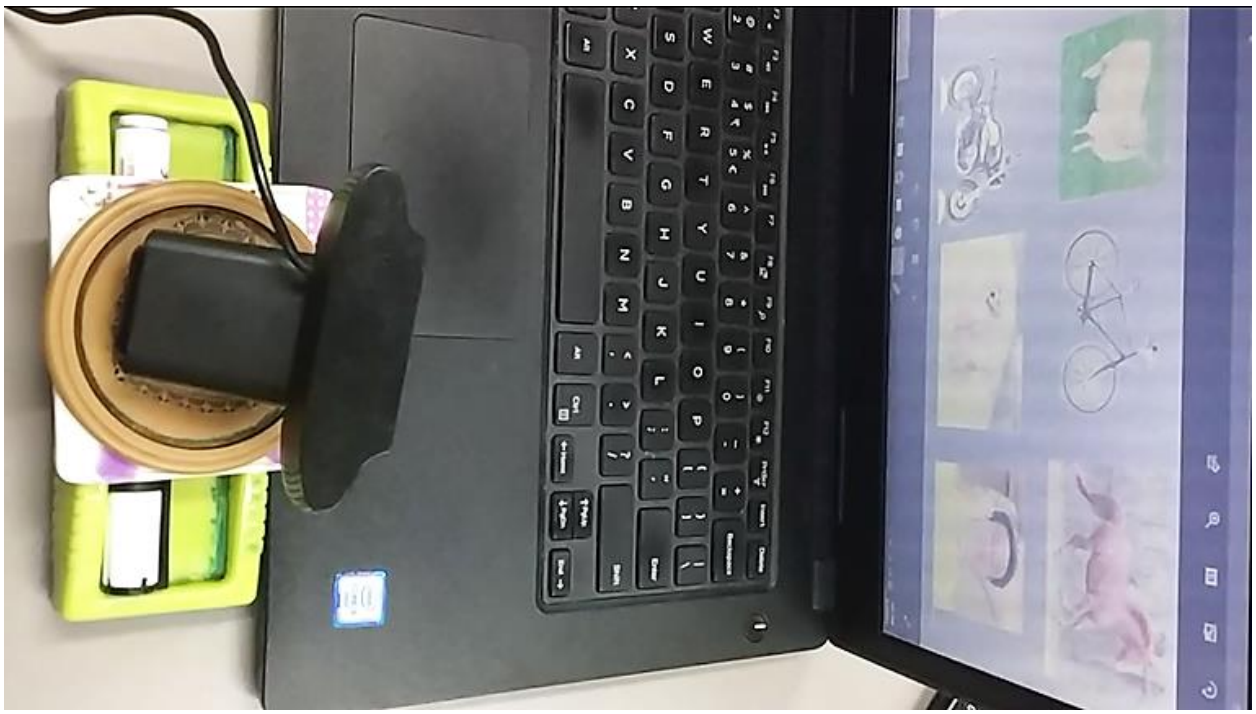

Figure 10: Setup for object detection

Figure 11: Sample Input Image for Object Detection

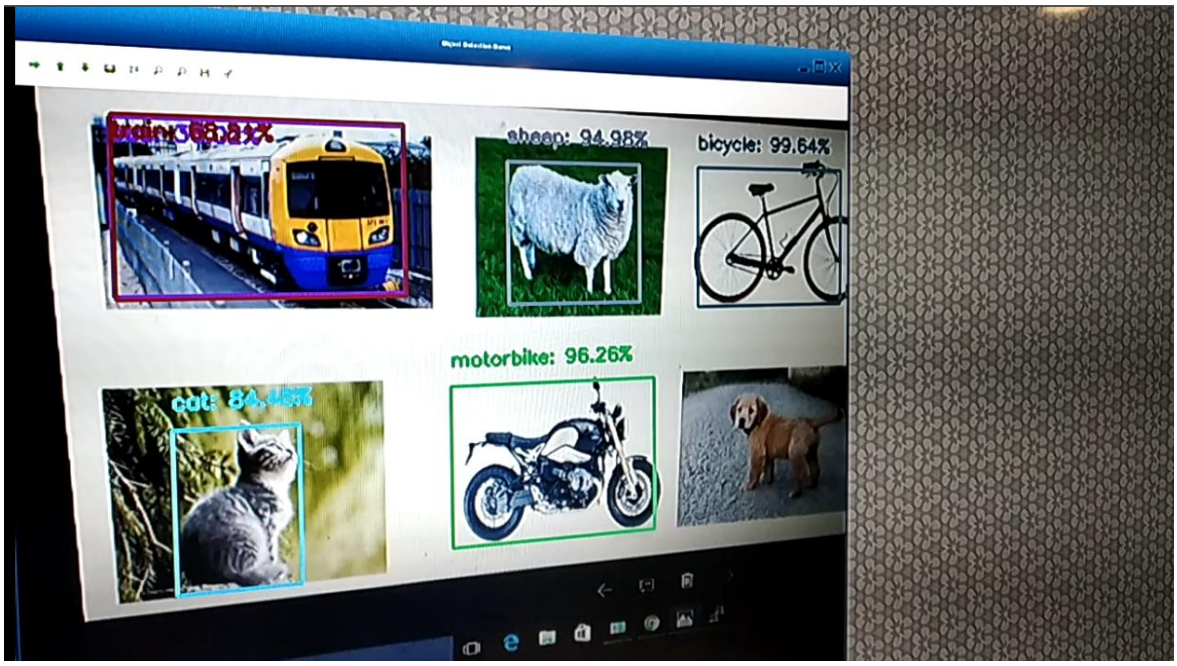

Figure 12: Sample Object detection output

As shown in above figure, in output image, dog is not detected, as it is not in good angle and exposure. Due to that model detect that with very low percentage and we ignore that due to low confidence. Also model confused train image with train and bus. If we retrain our model to train more images like that then we may get better performance.

3. Face Recognition Demo

In face recognition demo, we first detect face from given image or video frame. After detecting face, we apply our pre-trained model on it and try to recognize face.

In this demo, we have used few ML techniques and OpenCV face recognition models. We also used “face recognition” python module. Based on speed and accuracy we finally conclude that:

- **FACE DETECTION:** OpenCV is very basic model. It is slow in face detection as compare to ML module but fast compare to Python module.
- **FACE RECOGNITION:** Here OpenCV is winner. ML secure second position while python module is last on recognition time.
- **ACCURACY:** Here Python module is winner. ML have more accuracy then OpenCV model.

Based on above conclusion, we have made two face recognition demos.

In **Fast face recognition demo**, we use python face recognition module, which is more accurate than other two, but very slow. To make it fast we create two python processes. In one python process, we sample one frame at a time and done face recognition on it. Another python process will use this face recognition data and apply it on the entire frame it read from Camera. Thus, Camera output is smooth but real face detection take 3 to 4 secs to give actual real-time output.

In **real-time face recognition demo**, we have single python process, which read camera frame first and do face recognition on it. Therefore, here we do face recognition on each frame and video output is choppy. However, here we got real-time output and No delay. Due to that, this demo is perfect to identify board’s capabilities.

Both the demo have **capability to retrain model** run-time on board. For that, we need to provide new label (Person Name). After that, we capture new training data (photos), in which have label person’s face. If camera frame have more than one face, then we consider it malfunction or wrong frame and won’t consider that frame valid for training data. So simply ignore it. It is user’s responsibility to provide correct face data. If user provide different face images for training under same name then our model will be confuse as it found separate face vector data for different faces but found same name (label). In addition, if user create multiple label with same faces data then also model will easily get confused and will give output with mix of those labels.

Therefore, ideally we should provide **correct data and label** for training data. This training data is same for both the demos. Therefore, no need to create separate training data for both demos for example “anil_fast” and “anil_slow”. Also user need to provide data with different poses like smiling face, sad face, neutral face, angry face, face with eyeglasses etc. In addition, create training data with small (far) and big (near) faces.

We took approximately 50 photos of user. User can interrupt this capturing process by **pressing “q”** button from keyboard attached to board. Same interrupt process is working for testing as well. Means user can cancel testing any time by pressing “q” button.

If user want to re-train model again without capturing new training data, then we must not provide label and simply run training process.

Pre-requisite:

- Webcam or D3 Mezzanine camera
- USB Mouse
- USB Keyboard
- Use USB Hub (if have USB webcam) because we have only two USB ports
- HDMI Display having minimum 1080p resolution

Steps to run Demo:

Run **/run/media/mmcblk1p3/ARROW_DEMOS/run_ml_demos.sh script** and select **option 3.**

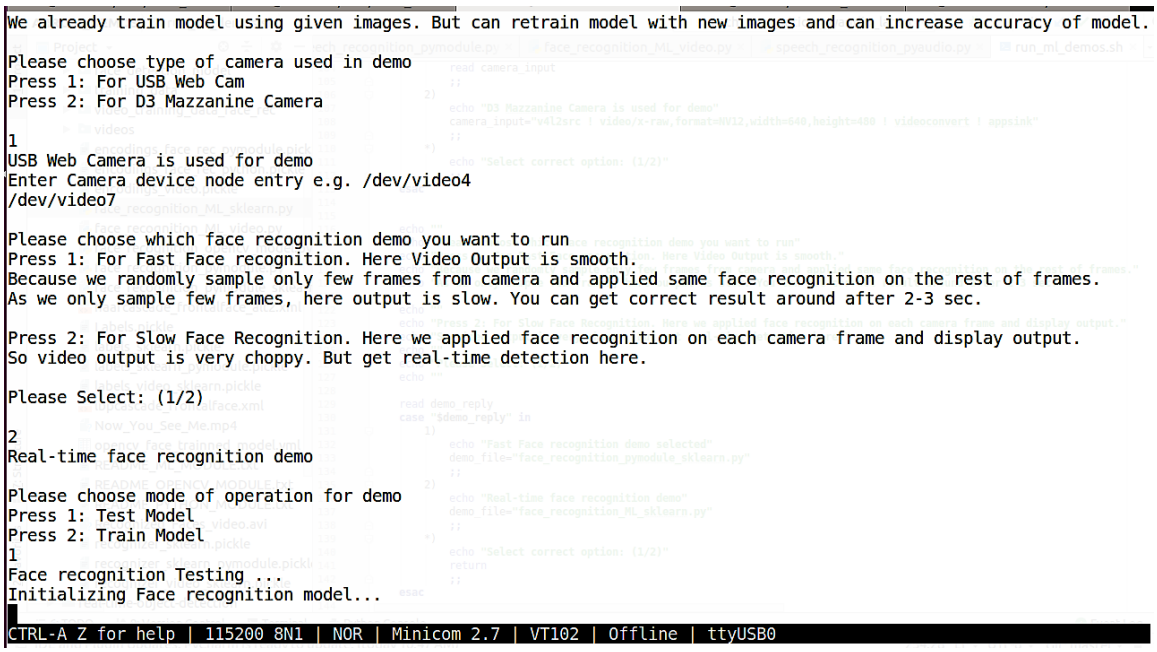

Figure 13: Face Recognition Demo Testing

See below full log to run demo, where user input is in **BOLD RED** fonts.

 # sh /run/media/mmcblk1p3/ARROW_DEMOS/run_ml_demos.sh

######### Welcome to ML Demos [AI Corowd Count/Object detection/Face Recognition/Speech Recognition/Arm NN] ###########

Prerequisite: Have you run <setup_ml_demo.sh>?

Press: (y/n)

n

****** Script Started *******

Setup is already completed. No need to do anything. Exiting...

Choose the option from following

Press 1: AI Crowd Count

Press 2: Object Detection

Press 3: Face Recognition

.

.

Select: (1/2/3/4/5)

3

Welcome to Face Recognition Demo

This is a demo application using Python modules to be run on embedded devices for recognition of faces.

We already train model using given images. But can retrain model with new images and can increase accuracy of model.

Please choose type of camera used in demo

Press 1: For USB Web Cam

Press 2: For D3 Mazzanine Camera

1

USB Web Camera is used for demo

Enter Camera device node entry e.g. /dev/video4

/dev/video7

Please choose which face recognition demo you want to run

Press 1: For Fast Face recognition. Here Video Output is smooth.

Because we randomly sample only few frames from camera and applied same face recognition on the rest of frames.

As we only sample few frames, here output is slow. You can get correct result around after 2-3 sec.

Press 2: For Slow Face Recognition. Here we applied face recognition on each camera frame and display output.

So video output is very choppy. But get real-time detection here.

Please Select: (1/2)

2

Real-time face recognition demo

Please choose mode of operation for demo

Press 1: Test Model

Press 2: Train Model

1

Face recognition Testing ...

Initializing Face recognition model...

Starting video stream...

*(python3:3946): GStreamer-CRITICAL **: gst_element_get_state: assertion 'GST_IS_ELEMENT (element)' failed*

Using Wayland-EGL

Using the 'xdg-shell-v6' shell integration

`q` pressed, Exiting...

Total Elapsed time: 12.07 sec

Approx. FPS: 0.99

Exiting Demo...

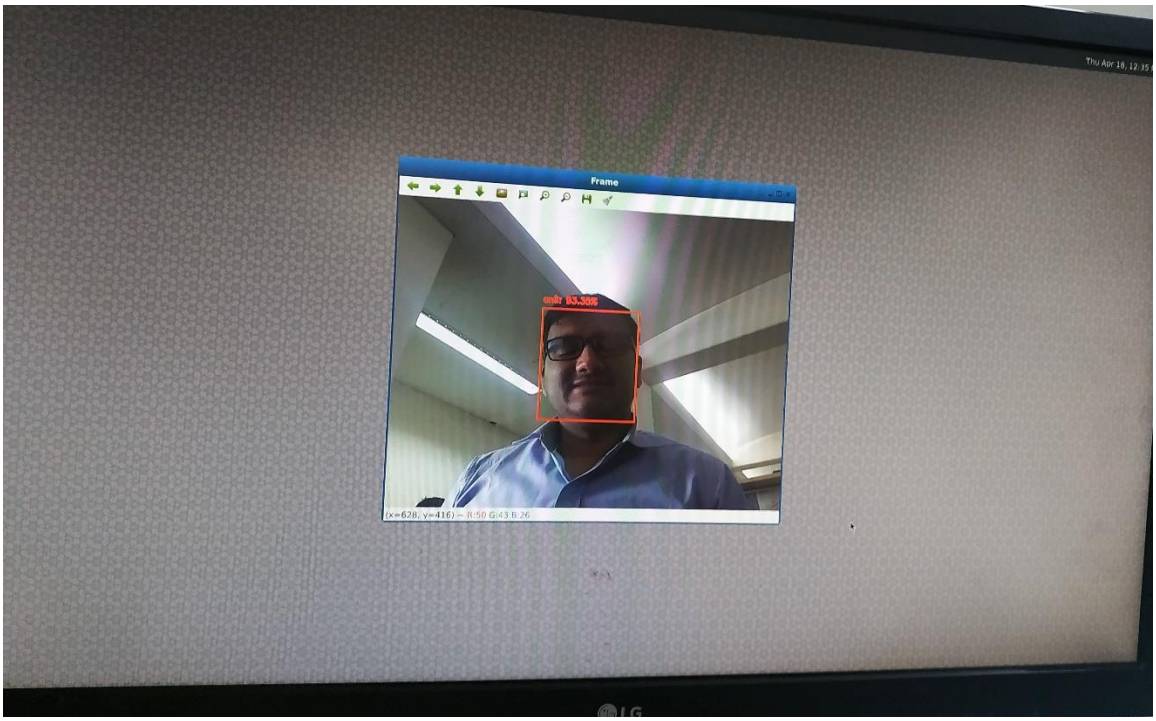

Figure 14: Face Recognition Output

If user want to re-train model with new person/label then he/she need to select train model with label as shown in below logs:

Please choose mode of operation for demo

Press 1: Test Model

Press 2: Train Model

2

Face recognition Training ...

Please provide new label... e.g. Joshua

With this label, we will create new training dataset and will our retrain model.

If you don't want to create new training dataset but simply retrain model with existing images, then press enter with

out giving any label name

anil

Directory anil created

Starting video stream to capture new training dataset...

*(python3:3993): GStreamer-CRITICAL **: gst_element_get_state: assertion 'GST_IS_ELEMENT (element)' failed*

Using Wayland-EGL

Using the 'xdg-shell-v6' shell integration

[]

detected faces - 0

[]

detected faces - 0

[]

detected faces - 0

[(125, 440, 254, 311)]

detected faces - 1

new training image anil_0_20190409-143240.png saved!

[]

detected faces - 0

[]

detected faces - 0

[(36, 366, 222, 180)]

detected faces - 1

new training image anil_1_20190409-143240.png saved!

[]

detected faces - 0

[(23, 438, 290, 171)]

detected faces - 1

new training image anil_2_20190409-143240.png saved!

[]

detected faces - 0

[]

detected faces - 0

[]

detected faces - 0

`q` pressed, Exiting... <---User Press "q" here from keyboard attached to board

Total Elapsed time: 31.82 sec

Directory anil removed

Quantifying faces from training dataset...

Processing Image : training_data/unknown/00000022.JPG

Processing Image : training_data/unknown/00000034.JPG

Processing Image : training_data/unknown/ellie_sattler.jpg

Processing Image : training_data/unknown/00000009.jpg

Processing Image : training_data/unknown/00000004.jpg

Encoding done for image = training_data/unknown/00000022.JPG

Encoding done for image = training_data/unknown/00000034.JPG

Processing Image : training_data/unknown/ds3.jpg

Processing Image : training_data/unknown/00000016.jpg

This will re-train model with new label **"anil"** training images along with other training images which are already provided inside **training_data** folder.

Here we create new training directory anil or copy new images inside existing directory if already have directory with same name. Therefore, if we have low confidence for any face we can re-train our model with that face and can increase confidence for that.

Training takes few minutes (around 10-15 minutes) depending on number of training images. In training, we utilize four cores of CPU and do training on images using four concurrent python process, which speed up our executions.

If User simply want to re-train model without providing any new input images or labels then we can do that by simply press "enter" and **leave blank when script ask for new label**. This scenario is only useful if your existing train model is corrupted or deleted by mistake. We can do training on our HOST Linux machine and can use trained model for testing but python module version must be same or compatible with version of board's package.

4. Speech Recognition Demo

This audio demo is use case of on-board DMIC. Here we have two demos for testing MIC.

In First Demo, we use our custom trained model for few selected keywords like **"yes no up down left right on off stop go".** In this demo we got around 80-85% accuracy as it is bit hard for audio to identify correct keyword compare to image where we easily get accuracy more than 90% by CNN (convolutional neural network).

Steps to run Demo:

Run **/run/media/mmcblk1p3/ARROW_DEMOS/run_ml_demos.sh script** and select **option 4.**

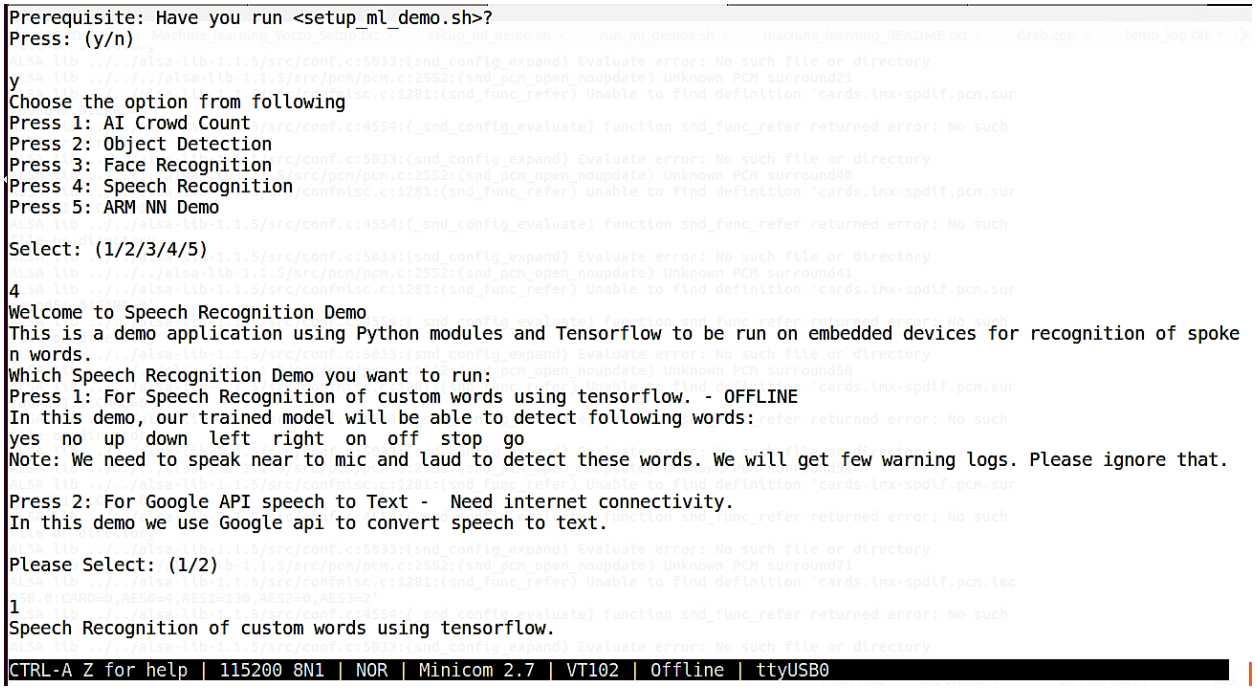

Figure 15: Run Speech Recognition Demo

 # sh /run/media/mmcblk1p3/ARROW_DEMOS/run_ml_demos.sh

######## Welcome to ML Demos [AI Corowd Count/Object detection/Face Recognition/Speech Recognition/Arm NN] ##########

Prerequisite: Have you run <setup_ml_demo.sh>?

Press: (y/n)

y

Choose the option from following

Press 1: AI Crowd Count

Press 2: Object Detection

Press 3: Face Recognition

Press 4: Speech Recognition

Press 5: ARM NN Demo

Select: (1/2/3/4/5)

4

Welcome to Speech Recognition Demo

This is a demo application using Python modules and Tensorflow to be run on embedded devices for recognition of spoken words.

Which Speech Recognition Demo you want to run:

Press 1: For Speech Recognition of custom words using tensorflow. - OFFLINE

In this demo, our trained model will be able to detect following words:

yes no up down left right on off stop go

Note: We need to speak near to mic and laud to detect these words. We will get few warning logs. Please ignore that.

Press 2: For Google API speech to Text - Need internet connectivity.

In this demo we use Google api to convert speech to text.

Please Select: (1/2)

1

Speech Recognition of custom words using tensorflow.

ImportError: No module named 'numpy.core._multiarray_umath'

ImportError: No module named 'numpy.core._multiarray_umath'

Expression 'alsa_snd_pcm_hw_params_set_buffer_size_near(pcm, hwParams, &alsaBufferFrames)' failed in '../portaudio/src/hostapi/alsa/pa_linux_alsa.c', line: 922

Expression 'alsa_snd_pcm_hw_params_set_buffer_size_near(pcm, hwParams, &alsaBufferFrames)' failed in '../portaudio/src/hostapi/alsa/pa_linux_alsa.c', line: 922

[16847.014464] usb 1-1.3: reset high-speed USB device number 3 using cdns-usb3

ALSA lib ../../alsa-lib-1.1.5/src/confmisc.c:1281:(snd_func_refer) Unable to find definition 'cards.imx-spdif.pcm.front.0:CARD=0'

ALSA lib ../../alsa-lib-1.1.5/src/conf.c:4554:(_snd_config_evaluate) function snd_func_refer returned error: No such file or directory

ALSA lib ../../alsa-lib-1.1.5/src/conf.c:5033:(snd_config_expand) Evaluate error: No such file or directory

ALSA lib ../../../alsa-lib-1.1.5/src/pcm/pcm.c:2552:(snd_pcm_open_noupdate) Unknown PCM front

ALSA lib ../../../alsa-lib-1.1.5/src/pcm/pcm.c:2552:(snd_pcm_open_noupdate) Unknown PCM cards.pcm.rear

ALSA lib ../../../alsa-lib-1.1.5/src/pcm/pcm.c:2552:(snd_pcm_open_noupdate) Unknown PCM cards.pcm.center_lfe

ALSA lib ../../../alsa-lib-1.1.5/src/pcm/pcm.c:2552:(snd_pcm_open_noupdate) Unknown PCM cards.pcm.side

ALSA lib ../../alsa-lib-1.1.5/src/confmisc.c:1281:(snd_func_refer) Unable to find definition 'cards.imx-spdif.pcm.surround51.0:CARD=0'

ALSA lib ../../alsa-lib-1.1.5/src/conf.c:4554:(_snd_config_evaluate) function snd_func_refer returned error: No such file or directory

ALSA lib ../../alsa-lib-1.1.5/src/conf.c:5033:(snd_config_expand) Evaluate error: No such file or directory

ALSA lib ../../../alsa-lib-1.1.5/src/pcm/pcm.c:2552:(snd_pcm_open_noupdate) Unknown PCM surround21

ALSA lib ../../alsa-lib-1.1.5/src/confmisc.c:1281:(snd_func_refer) Unable to find definition 'cards.imx-spdif.pcm.surround51.0:CARD=0'

ALSA lib ../../alsa-lib-1.1.5/src/conf.c:4554:(_snd_config_evaluate) function snd_func_refer returned error: No such file or directory

ALSA lib ../../alsa-lib-1.1.5/src/conf.c:5033:(snd_config_expand) Evaluate error: No such file or directory

ALSA lib ../../../alsa-lib-1.1.5/src/pcm/pcm.c:2552:(snd_pcm_open_noupdate) Unknown PCM surround21

ALSA lib ../../alsa-lib-1.1.5/src/confmisc.c:1281:(snd_func_refer) Unable to find definition 'cards.imx-spdif.pcm.surround40.0:CARD=0'

ALSA lib ../../alsa-lib-1.1.5/src/conf.c:4554:(_snd_config_evaluate) function snd_func_refer returned error: No such file or directory

ALSA lib ../../alsa-lib-1.1.5/src/conf.c:5033:(snd_config_expand) Evaluate error: No such file or directory

ALSA lib ../../../alsa-lib-1.1.5/src/pcm/pcm.c:2552:(snd_pcm_open_noupdate) Unknown PCM surround40

ALSA lib ../../alsa-lib-1.1.5/src/confmisc.c:1281:(snd_func_refer) Unable to find definition 'cards.imx-spdif.pcm.surround51.0:CARD=0'

ALSA lib ../../alsa-lib-1.1.5/src/conf.c:4554:(_snd_config_evaluate) function snd_func_refer returned error: No such file or directory

ALSA lib ../../alsa-lib-1.1.5/src/conf.c:5033:(snd_config_expand) Evaluate error: No such file or directory

ALSA lib ../../../alsa-lib-1.1.5/src/pcm/pcm.c:2552:(snd_pcm_open_noupdate) Unknown PCM surround41

ALSA lib ../../alsa-lib-1.1.5/src/confmisc.c:1281:(snd_func_refer) Unable to find definition 'cards.imx-spdif.pcm.surround51.0:CARD=0'

ALSA lib ../../alsa-lib-1.1.5/src/conf.c:4554:(_snd_config_evaluate) function snd_func_refer returned error: No such file or directory

ALSA lib ../../alsa-lib-1.1.5/src/conf.c:5033:(snd_config_expand) Evaluate error: No such file or directory

ALSA lib ../../../alsa-lib-1.1.5/src/pcm/pcm.c:2552:(snd_pcm_open_noupdate) Unknown PCM surround50

ALSA lib ../../alsa-lib-1.1.5/src/confmisc.c:1281:(snd_func_refer) Unable to find definition 'cards.imx-spdif.pcm.surround51.0:CARD=0'

ALSA lib ../../alsa-lib-1.1.5/src/conf.c:4554:(_snd_config_evaluate) function snd_func_refer returned error: No such file or directory

ALSA lib ../../alsa-lib-1.1.5/src/conf.c:5033:(snd_config_expand) Evaluate error: No such file or directory

ALSA lib ../../../alsa-lib-1.1.5/src/pcm/pcm.c:2552:(snd_pcm_open_noupdate) Unknown PCM surround51

ALSA lib ../../alsa-lib-1.1.5/src/confmisc.c:1281:(snd_func_refer) Unable to find definition 'cards.imx-spdif.pcm.surround71.0:CARD=0'

ALSA lib ../../alsa-lib-1.1.5/src/conf.c:4554:(_snd_config_evaluate) function snd_func_refer returned error: No such file or directory

ALSA lib ../../alsa-lib-1.1.5/src/conf.c:5033:(snd_config_expand) Evaluate error: No such file or directory

ALSA lib ../../../alsa-lib-1.1.5/src/pcm/pcm.c:2552:(snd_pcm_open_noupdate) Unknown PCM surround71

ALSA lib ../../alsa-lib-1.1.5/src/confmisc.c:1281:(snd_func_refer) Unable to find definition 'cards.imx-spdif.pcm.iec958.0:CARD=0,AES0=4,AES1=130,AES2=0,AES3=2'

ALSA lib ../../alsa-lib-1.1.5/src/conf.c:4554:(_snd_config_evaluate) function snd_func_refer returned error: No such file or directory

ALSA lib ../../alsa-lib-1.1.5/src/conf.c:5033:(snd_config_expand) Evaluate error: No such file or directory

ALSA lib ../../../alsa-lib-1.1.5/src/pcm/pcm.c:2552:(snd_pcm_open_noupdate) Unknown PCM iec958

ALSA lib ../../alsa-lib-1.1.5/src/confmisc.c:1281:(snd_func_refer) Unable to find definition 'cards.imx-spdif.pcm.iec958.0:CARD=0,AES0=4,AES1=130,AES2=0,AES3=2'

ALSA lib ../../alsa-lib-1.1.5/src/conf.c:4554:(_snd_config_evaluate) function snd_func_refer returned error: No such file or directory

ALSA lib ../../alsa-lib-1.1.5/src/conf.c:5033:(snd_config_expand) Evaluate error: No such file or directory

ALSA lib ../../../alsa-lib-1.1.5/src/pcm/pcm.c:2552:(snd_pcm_open_noupdate) Unknown PCM spdif

ALSA lib ../../alsa-lib-1.1.5/src/confmisc.c:1281:(snd_func_refer) Unable to find definition 'cards.imx-spdif.pcm.iec958.0:CARD=0,AES0=4,AES1=130,AES2=0,AES3=2'

ALSA lib ../../alsa-lib-1.1.5/src/conf.c:4554:(_snd_config_evaluate) function snd_func_refer returned error: No such file or directory

ALSA lib ../../alsa-lib-1.1.5/src/conf.c:5033:(snd_config_expand) Evaluate error: No such file or directory

ALSA lib ../../../alsa-lib-1.1.5/src/pcm/pcm.c:2552:(snd_pcm_open_noupdate) Unknown PCM spdif

ALSA lib ../../../alsa-lib-1.1.5/src/pcm/pcm.c:2552:(snd_pcm_open_noupdate) Unknown PCM cards.pcm.hdmi

ALSA lib ../../../alsa-lib-1.1.5/src/pcm/pcm.c:2552:(snd_pcm_open_noupdate) Unknown PCM cards.pcm.hdmi

ALSA lib ../../../alsa-lib-1.1.5/src/pcm/pcm.c:2552:(snd_pcm_open_noupdate) Unknown PCM cards.pcm.modem

ALSA lib ../../../alsa-lib-1.1.5/src/pcm/pcm.c:2552:(snd_pcm_open_noupdate) Unknown PCM cards.pcm.modem

1LSA lib ../../../alsa-lib-1.1.5/src/pcm/pcm.c:2552:(snd_pcm_open_noupdate) Unknown PCM cards.pcm.pho[n1e6l8i4n8e. 5A0L6S0A8]l ifbs l.-./es.a.i/-.d.a/ia l5s9a0-1l0i0b0-01..e1s.a5i/:s rfc/apilcemd/ ptcom .dce:r2i5v5e2: r(esqnudi_rpdcdm _SoCpKeRn _nraotuepate) Unknown PCM cards.pcm.phoneline

ALSA lib ../../alsa-lib-1.1.5/src/confmisc.c:1281:(snd_func_refer) Unable to find definition 'defaults.bluealsa.device'

ALSA lib ../../alsa-lib-1.1.5/src/conf.c:4554:(_snd_config_evaluate) function snd_func_refer returned error: No such file or directory

ALSA lib ../../alsa-lib-1.1.5/src/conf.c:5022:(snd_config_expand) Args evaluate error: No such file or directory

ALSA lib ../../../alsa-lib-1.1.5/src/pcm/pcm.c:2552:(snd_pcm_open_noupdate) Unknown PCM bluealsa

ALSA lib ../../alsa-lib-1.1.5/src/confmisc.c:1281:(snd_func_refer) Unable to find definition 'defaults.bluealsa.device'

ALSA lib ../../alsa-lib-1.1.5/src/conf.c:4554:(_snd_config_evaluate) function snd_func_refer returned error: No such file or directory

ALSA lib ../../alsa-lib-1.1.5/src/conf.c:5022:(snd_config_expand) Args evaluate error: No such file or directory

ALSA lib ../../../alsa-lib-1.1.5/src/pcm/pcm.c:2552:(snd_pcm_open_noupdate) Unknown PCM bluealsa

ALSA lib ../../alsa-plugins-1.1.5/pulse/pulse.c:243:(pulse_connect) PulseAudio: Unable to connect: Connection refused

ALSA lib ../../alsa-plugins-1.1.5/pulse/pulse.c:243:(pulse_connect) PulseAudio: Unable to connect: Connection refused

ALSA lib ../../../alsa-lib-1.1.5/src/pcm/pcm_dmix.c:1035:(snd_pcm_dmix_open) The dmix plugin supports only playback stream

ALSA lib ../../../alsa-lib-1.1.5/src/pcm/pcm_dmix.c:1035:(snd_pcm_dmix_open) The dmix plugin supports only playback stream

ALSA lib ../../../alsa-lib-1.1.5/src/pcm/pcm_dmix.c:1035:(snd_pcm_dmix_open) The dmix plugin supports only playback stream

AL[S1A6 8l4i8b. 3.0.2/9.3.6/]. .f/sall-seas-alii-bd-a1i. 15.950/1s0r0c0/0p.cems/apic:m _AdSmoiCx:. cc:a1n0'3t5 :s(estg d5_9p0c1m0_0d0m0ix.e_soapie nh)w Tphaer admmsi:x -p2l2u

in supports only playback stream

ALSA lib ../../../alsa-lib-1.1.5/src/pcm/pcm_dmix.c:1035:(snd_pcm_dmix_open) The dmix plugin supports only playback stream

ALSA lib ../../../alsa-lib-1.1.5/src/pcm/pcm_dsnoop.c:575:(snd_pcm_dsnoop_open) The dsnoop plugin supports only capture stream

ALSA lib ../../../alsa-lib-1.1.5/src/pcm/pcm_direct.c:1271:(snd1_pcm_direct_initialize_slave) unable to insta[1l6l8 4h8w. 3p5a2r7a0m3s]

fAsLlS-Ae slaiib- d.a.i/ .5.9/0.1./00a0l0s.ae-sliaib:- 1f.a1i.l5e/ds rtco/ pdcme/ripvcem _rdesqnuoiorpe.dc :6SC4K9R: p(rsantde_ cm_dsnoop_open) unable to initialize slave

ALSA lib ../../../alsa-lib-1.1.5/src/pcm/pcm_dsnoop.c:575:(snd_pcm_dsnoop_open) The dsnoop plugin supports only capture stream

ALSA lib ../../../alsa-lib-1.1.5/src/pcm/pcm_dsnoop.c:575:(snd_pcm_dsnoop_open) The dsnoop plugin supports only capture stream

[16848.392805] fsl-esai-dai 59010000.esai: ASoC: can't set 59010000.esai hw params: -22

ALSA lib ../../../alsa-lib-1.1.5/src/pcm/pcm_direct.c:1271:(snd1_pcm_direct_initialize_slave) unable to inst[a1l6l8 4

8w. 4p0a9r9a0m1s]

fAsLlS-Ae slaiib- d.a.i/ .5.9/.01.0/0a0l0s.ae-slaiib:- 1t.1h.e 5r/astrico/ picsm /opuctm_ dofs nroaonpg.ec :(614 9~:_(s16n)d

pcm_dsnoop_open) unable to initialize slave

ALSA lib ../../../alsa-lib-1.1.5/src/pcm/pcm_dsnoop.c:575:(snd_pcm_dsnoop_open) The dsnoop plugin supports only capture stream

ALSA lib ../../../alsa-lib-1.1.5/src/pcm/pcm_dsnoop.c:575:(snd_pcm_dsnoop_open) The dsnoop plugin supports only capture stream

[16848.449914] fsl-esai-dai 59010000.esai: ASoC: can't set 59010000.esai hw params: -22

ALSA lib ../../../alsa-lib-1.1.5/src/pcm/pcm_direct.c:1271:(snd1_pcm_direct_initialize_slave) unable to install hw params

ALSA lib ../../../alsa-lib-1.1.5/src/pcm/pcm_dsnoop.c:649:(snd_pcm_dsnoop_open) unable to initialize slave

ALSA lib ../../../alsa-lib-1.1.5/src/pcm/pcm_dsnoop.c:575:(snd_pcm_dsnoop_open) The dsnoop plugin supports only capture stream

Cannot connect to server socket err = No such file or directory

Cannot connect to server request channel

jack server is not running or cannot be started

JackShmReadWritePtr::~JackShmReadWritePtr - Init not done for -1, skipping unlock

JackShmReadWritePtr::~JackShmReadWritePtr - Init not done for -1, skipping unlock

Available Audio Devices : Index

imx-spdif: - (hw:0,0) : 0

imx-audio-sph0645: - (hw:1,0) : 1

USB Device 0x46d:0x821: Audio (hw:3,0) : 2

sysdefault : 3

dmix_48000 : 4

dmix_44100 : 5

dmix_32000 : 6

dmix_16000 : 7

dmix_8000 : 8

dsnoop_48000 : 9

dsnoop_32000 : 10

dsnoop_16000 : 11

asymed : 12

dsp0 : 13

dmix : 14

default : 15

*Which input (audio) device you want? Please provide index value (in number) : **1***

*You selected audio device : **1***

Expression 'alsa_snd_pcm_hw_params_set_buffer_size_near(pcm, hwParams, &alsaBufferFrames)' failed in '../portaudio/src/hostapi/alsa/pa_linux_alsa.c', line: 922

.

. <SAME ERROR AS ABOVE>

.

Cannot connect to server socket err = No such file or directory

Cannot connect to server request channel

jack server is not running or cannot be started

JackShmReadWritePtr::~JackShmReadWritePtr - Init not done for -1, skipping unlock

JackShmReadWritePtr::~JackShmReadWritePtr - Init not done for -1, skipping unlock

Say Something Now!. It will continuously convert speech to text and Wait till user pressed CTRL+C ...

unknown (prediction score = 44.53)

unknown (prediction score = 22.55)

stop (prediction score = 22.94)

go (prediction score = 48.45)

left (prediction score = 44.40)

yes (prediction score = 26.72)

right (prediction score = 97.46)

down (prediction score = 27.49)

stop (prediction score = 20.01)

yes (prediction score = 93.53)

no (prediction score = 50.74)

go (prediction score = 51.61)

up (prediction score = 51.55)

no (prediction score = 50.00)

go (prediction score = 26.50)

down (prediction score = 20.24)

down (prediction score = 46.60)

no (prediction score = 30.87)

no (prediction score = 20.47)

^CExiting Demo...

Exiting Demo...

root@imx8qxpaiml:~#

Here As we seen in above output, we got lots of error and warning regarding ALSA Lib. This is because ALSA lib try to configure capture only device to playback and vice versa. This errors want affect our actual behavior so kindly ignore that.

Also, we need to select Audio device 1 which is **"imx-audio-sph0645: - (hw:1,0)" DMIC.** We **need to speak bit loud and clear near to board** to capture audio perfectly and got confidence higher. If confidence is lower than we are simply, ignore that spoken word. We display top three predictions for spoken words if their confidence is at least greater than 20%. Some keyword like "**go**" and "**left**" have lower confidence due to their echo is similar to other spoken words like "**no**".

Here we don't provide mechanism for re-train but in actual speech recognition we need to continuously retrain our model with spoken words to increase accuracy and confidence. Due to that Google and other company like Amazon, Apple have better voice recognition system. (They have lots of spoken data and increasing it by each day.)

If we use good USB MIC instead of default DMIC then also we got better performance. As in external USB MIC, we have good feature like noise and echo cancellation. Due to that audio data input is much accurate and valid. If we want to test demo with USB MIC then we need to provide appropriate hardware index entry for that. For example, in our case USB MIC have index entry 2 (USB Device 0x46d:0x821: Audio (hw:3,0) : **2**) so we need to provide input 2 to test demo with USB.

In second demo, we use **Google Speech to Text** API to validate our MIC. Here we need internet connection. This model have more accuracy then our custom model. This Demo listen for 3 to 5 sec and convert those audio data to text. Here we can define our keywords and can perform basic operation based on that. For example, if we provide browser support in our firmware release then we identify keywords and perform action on browser like "**Anil open wikipedia**".

In this demo ANIL, WIKIPEDIA, YOUTUBE, GOOGLE are keywords.

In latest beta firmware, we still not provide support for browser (The only reason - browser is resource consuming) but if we have that then we can perform some task as mention below:

Say Something Now!. It will continuously convert speech to text and Wait till 'ENTER' pressed...

*Speech Recognition thinks you said **: Anil open Google***

Main Keyword detected...

Opening Google in browser...

Speech Recognition thinks you said : ***Anil open Wikipedia***

Main Keyword detected...

Opening WikiPedia in browser...

Speech Recognition thinks you said ***: Anil***

Main Keyword detected...

Speech Recognition thinks you said : ***Anil open YouTube***

Main Keyword detected...

Opening YouTube in browser...

Speech Recognition thinks you said : ***Anil open YouTube search latest song***

Main Keyword detected...

Opening YouTube in browser...

Speech Recognition thinks you said : ***Anil open Wikipedia search today's history***

Main Keyword detected...

Opening WikiPedia in browser...

5. Basler Camera Demo

In latest release, we provided support for basler USB camera. We have tested with **DAA2500-14UM (CS-MOUNT) - BASLER DART** camera.

Link: https://www.baslerweb.com/en/products/cameras/area-scan-cameras/dart/daa2500-14um-cs-mount/#tab=specs

However, this demo app can be run with any Basler camera. Here we configure pylon5 software for our board.

To view camera preview user need to select appropriate input i.e. 5.

```
root@imx8qxpaiml:~/ARROW_DEMOS#
root@imx8qxpaiml:~/ARROW_DEMOS# sh run_ml_demos.sh
########  Welcome to ML Demos [AI Corowd Count/Object detection/Face Recognition/Speech Recognition/Arm NN] ##########
Prerequisite: Have you run <setup_ml_demo.sh>?
Press: (y/n)

y
Choose the option from following
Press 1 : AI Crowd Count
Press 2 : Object Detection
Press 3 : Face Recognition
Press 4 : Speech Recognition
Press 5 : Basler Camera demo
Press 6 : Face Recognition using TensorFlow Lite demo
Press 7 : Object Recognition using Arm NN demo

Select: (1/2/3/4/5/6/7)

5
Welcome to Basler camera Pylon Viewer App.
In this Demo, we setup pylon software and run pylon viewer app provided by basler.
In Pylon viewer app we can select our USB based camera and get real-time video view with other controlling parameters.
Notice: PYLON_ROOT was already set. It got replaced with '/opt/pylon5'
run_ml_demos.sh: line 235: cd: pylon-5.1.0.12682-arm64/: No such file or directory
run_ml_demos.sh: line 236: /home/root/basler/pylon-5.1.0.12682-arm64/setup-usb.sh: No such file or directory
Unplug and Plug Basler USB camera.
Starting Pylon Viewer App...
QIconvCodec::convertFromUnicode: using Latin-1 for conversion, iconv_open failed
QIconvCodec::convertToUnicode: using Latin-1 for conversion, iconv_open failed
[  937.648414] usb 1-1.1: new full-speed USB device number 5 using cdns-usb3
[  937.776791] input: Logitech USB Receiver as /devices/platform/5b110000.usb3/xhci-cdns3/usb1/1-1/1-1.1/1-1.1:1.0/0003:046D:C534.0001/input/input1
[  937.849399] hid-generic 0003:046D:C534.0001: input: USB HID v1.11 Keyboard [Logitech USB Receiver] on usb-xhci-cdns3-1.1/input0
[  937.866755] input: Logitech USB Receiver as /devices/platform/5b110000.usb3/xhci-cdns3/usb1/1-1/1-1.1/1-1.1:1.1/0003:046D:C534.0002/input/input2
[  937.936679] hid-generic 0003:046D:C534.0002: input: USB HID v1.11 Mouse [Logitech USB Receiver] on usb-xhci-cdns3-1.1/input1
[  953.913175] capability: warning: `QThread' uses 32-bit capabilities (legacy support in use)
Exiting Demo...
root@imx8qxpaiml:~/ARROW_DEMOS#
root@imx8qxpaiml:~/ARROW_DEMOS#
root@imx8qxpaiml:~/ARROW_DEMOS#
```

Figure 16: Basler Camera logs

This demo setup for Basler camera and run pylon viewer app. (see figure 17)

App features (See rectangle box in attached figures with appropriate number):

1. On left - top side we see detected Basler camera.
2. On left - bottom side we can adjust camera features like brightness and other parameters.
3. From feature panel select continuous shot.
4. After selecting continuous shot we can see camera preview in preview area.
5. With Full screen feature, we can see camera preview on full screen.

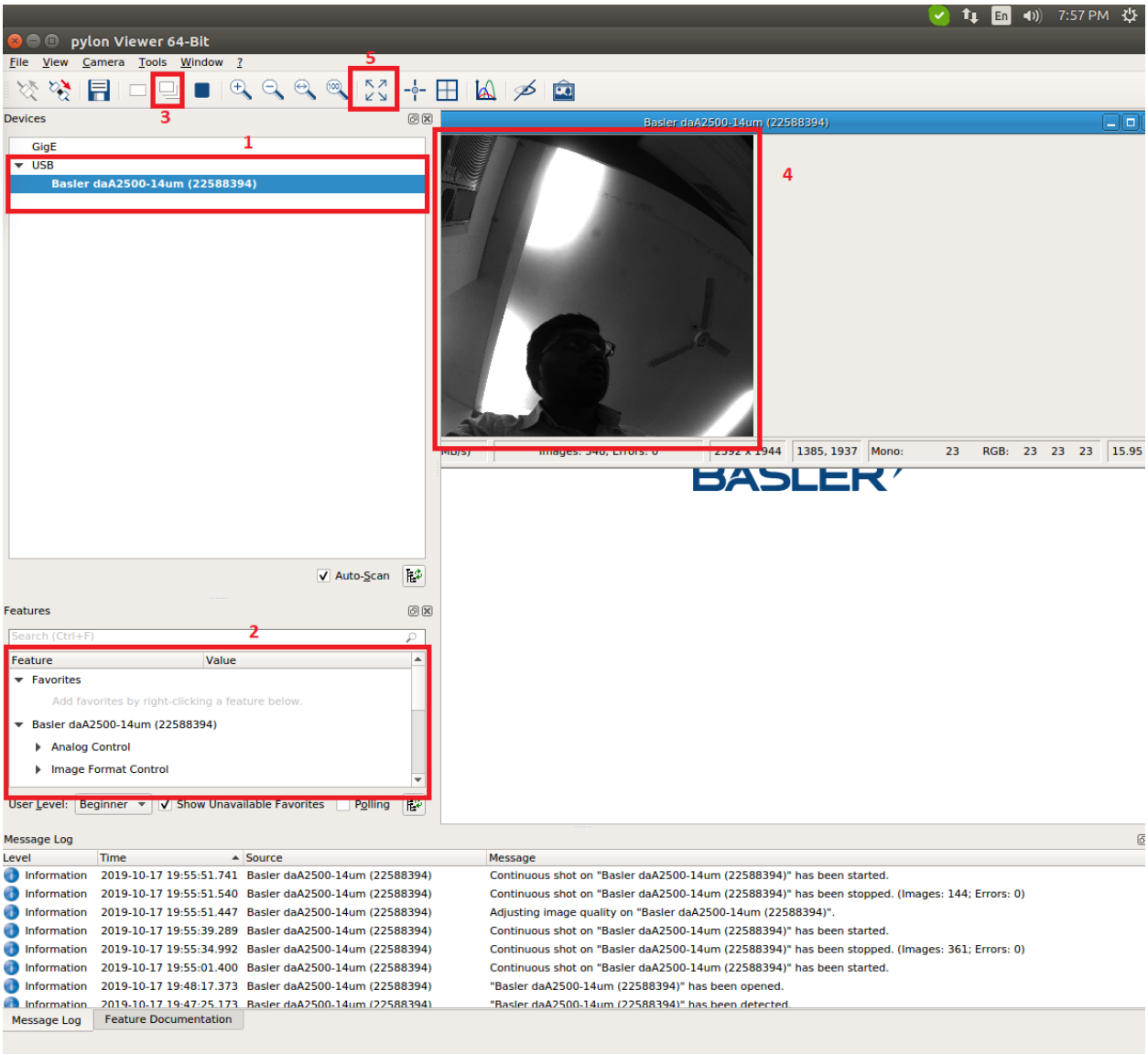

Figure 17: Basler Pylon Viewer App

Note: If We run App in maximize mode, then on left side we observed background display overlay. This is pylon app issue and it is observed on Linux as well as on board preview.

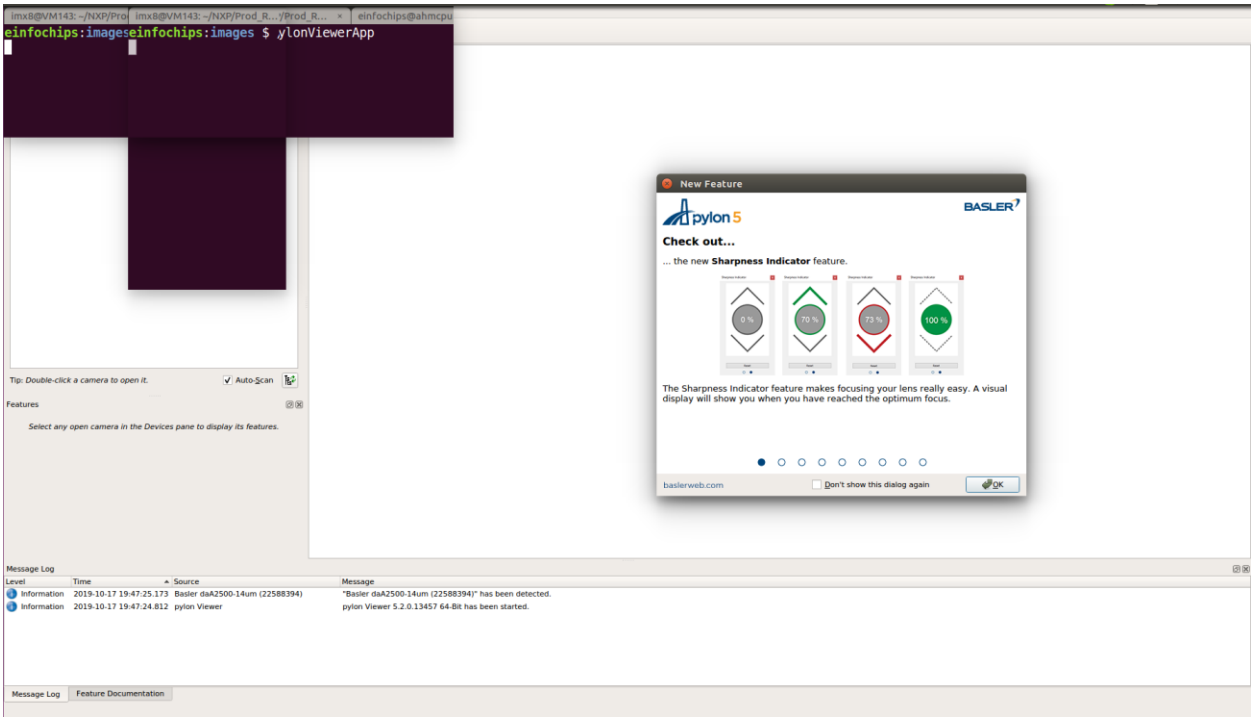

Figure 18: Pylon Viewer App display issue

6. Face Recognition using Tensorflow Lite demo

This application demo uses Haar Feature-based Cascade Classifiers for real time face detection. The pre-trained Haar Feature-based Cascade Classifiers for face, named as XML. TensorFlow Lite implementation for MobileFaceNets.

The MobileFaceNets is re-trained with a smaller batch size and input size to get a higher performance on a host PC. The trained model is loaded as a source file in this demo.

Steps to run Tf_based Face Recognition Demo:

Run **/run/media/mmcblk1p3/ARROW_DEMOS/run_ml_demos.sh script and select appropriate option** ,then select Node Entry e.g **/0/1/2/3/4**.

Figure 19: Tensorflow based Face Recognition demo run screen

sh /run/media/mmcblk1p3/ARROW_DEMOS/run_ml_demos.sh

######## Welcome to ML Demos [AI Corowd Count/Object detection/Face Recognition/Speech Recognition/Arm NN] ##########

Prerequisite: Have you run <setup_ml_demo.sh>?

Press: (y/n)

y

Choose the option from following

Press 1 : AI Crowd Count

Press 2 : Object Detection

Press 3 : Face Recognition

Press 4 : Speech Recognition

Press 5 : ARM NN Demo

Press 6 : Basler Camera demo

Press 7 : Face Recognition using TensorFlow Lite demo

Press 8 : Object Recognition using Arm NN demo

Select: (1/2/3/4/5/6/7/8)

7

Welcome to Face Recognition using TensorFlow Lite demo

Detecting Biggest Face in Real-Time

Pleae provide Camera Node Entry

Node entry e.g. /dev/video4 so enter 4 as numeric

4

Using Wayland-EGL

Using the 'xdg-shell-v6' shell integration

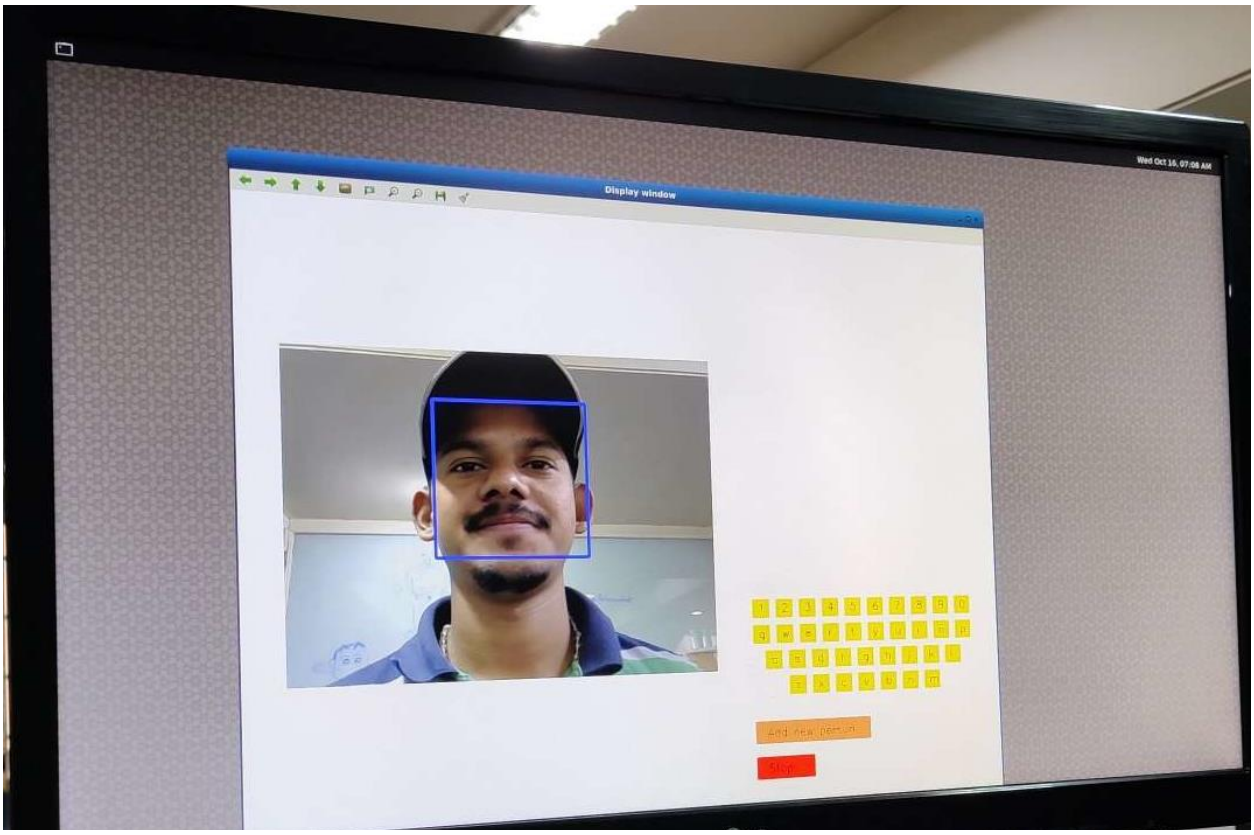

Figure 20: Tensorflow based Face Recognition demo output screen

When the demo is running, it will detect one biggest face at real time. Once the face is detected, you can click keyboards on the right of GUI to input the new person's name. Then, click 'Add new person' to add the face to data set.

In brief,

1. Detect face.

2. Input new person's name.

3. Click 'Add new person'.

Note: Once new faces are added, it will create a folder named 'data' in current directory. If you want to remove the new face from the data set, just delete it in 'data'.

7. Object Recognition using Arm NN Demo

In latest release, we have added support for ARM NN SDK and ARM Compute Library. This Demo run ARM NN example to test performance of ARM NN on our board.

This demo contains samples for running inference and predicting different objects. It also includes an extension that can recognize any given camera input/object.

Steps to run File_Based Demo:

Run **/run/media/mmcblk1p3/ARROW_DEMOS/run_ml_demos.sh** script and select option **8** ,then select option **1**.

```
root@imx8qxpaiml:/run/media/mmcblk1p3/ARROW_DEMOS# sh run_ml_demos.sh
########  Welcome to ML Demos [AI Corowd Count/Object detection/Face Recognition/Speech Recognition/Arm NN] ##########
Prerequisite: Have you run <setup_ml_demo.sh>?
Press: (y/n)

y
Choose the option from following
Press 1 : AI Crowd Count
Press 2 : Object Detection
Press 3 : Face Recognition
Press 4 : Speech Recognition
Press 5 : ARM NN Demo
Press 6 : Basler Camera demo
Press 7 : Face Recognition using TensorFlow Lite demo
Press 8 : Object Recognition using Arm NN demo

Select: (1/2/3/4/5/6/7/8)

8
Welcome to Object Recognition using Arm NN demo
This is a demo to show how much accuracy in detecting an Object like Cat, Dog, Shark
This will pick the image from data folder and provides the accuracy of respective image with this identity No.
List of Images with their identity is provided at https://github.com/ARM-software/armnn/blob/branches/armnn_18_11/tests/TfLiteMobilenetQuantized-Armnn/labels.txt
Please choose type of example method used in demo
Press 1: For File_Based Demo
Press 2: For MIPI Camera
1
File_Based Object-Recognition Demo
*******Command Executing*******Detecting Object******Predicting Accuracy********
```

Figure 21: Arm NN Object Recognition run screen

 # sh /run/media/mmcblk1p3/ARROW_DEMOS/run_ml_demos.sh

######## Welcome to ML Demos [AI Corowd Count/Object detection/Face Recognition/Speech Recognition/Arm NN] ##########

Prerequisite: Have you run <setup_ml_demo.sh>?

Press: (y/n)

y
Choose the option from following
Press 1 : AI Crowd Count

Press 2 : Object Detection

Press 3 : Face Recognition

Press 4 : Speech Recognition

Press 5 : ARM NN Demo

Press 6 : Basler Camera demo

Press 7 : Face Recognition using TensorFlow Lite demo

Press 8 : Object Recognition using Arm NN demo

Select: (1/2/3/4/5/6/7/8)

8

Welcome to Object Recognition using Arm NN demo

This is a demo to show how much accuracy in detecting an Object like Cat, Dog, Shark This will pick the image from data folder and provides the accuracy of respective image with this identity No.

List of Images with their identity is provided at https://github.com/ARM-software/armnn/blob/branches/armnn_18_11/tests/TfLiteMobilenetQuantized-Armnn/labels.txt

Please choose type of example method used in demo

Press 1: For File_Based Demo

Press 2: For MIPI Camera

1

File_Based Object-Recognition Demo

*******Command Executing*******Detecting Object******Predicting Accuracy********
ArmNN v20190200

= Prediction values for test #0

Top(1) prediction is 208 with confidence: 90.0606%

Top(2) prediction is 206 with confidence: 0.0772798%

Top(3) prediction is 176 with confidence: 0.0398233%

Top(4) prediction is 60 with confidence: 0.020664%

Top(5) prediction is 23 with confidence: 0.0152533%
= Prediction values for test #1

Top(1) prediction is 282 with confidence: 58.5935%

Top(2) prediction is 200 with confidence: 0.0584546%

Top(3) prediction is 139 with confidence: 0.0434935%

Top(4) prediction is 134 with confidence: 0.0408567%

Top(5) prediction is 133 with confidence: 0.0339192%

Prediction for test case 1 (282) is incorrect (should be 283)
= Prediction values for test #2

Top(1) prediction is 3 with confidence: 94.1774%

Top(2) prediction is 0 with confidence: 0.00358077%

Total time for 3 test cases: 4.955 seconds

Average time per test case: 1651.562 ms

```
File_Based Object-Recognition Demo
*******Command Executing*******Detecting Object******Predicting Accuracy********

Additional Options:
  +i                                    Set user specified inference model name.
                                        If not set, default name is used.
  +o                                    Set user specified output tensor name.
                                        If not set, default name is used.


Additional Options:
  +i                                    Set user specified inference model name.
                                        If not set, default name is used.
  +o                                    Set user specified output tensor name.
                                        If not set, default name is used.

ArmNN v20190200

= Prediction values for test #0
Top(1) prediction is 208 with confidence: 90.0606%
Top(2) prediction is 206 with confidence: 0.0772798%
Top(3) prediction is 176 with confidence: 0.0398233%
Top(4) prediction is 60 with confidence: 0.020664%
Top(5) prediction is 23 with confidence: 0.0152533%
= Prediction values for test #1
Top(1) prediction is 282 with confidence: 58.5935%
Top(2) prediction is 200 with confidence: 0.0584546%
Top(3) prediction is 139 with confidence: 0.0434935%
Top(4) prediction is 134 with confidence: 0.0408567%
Top(5) prediction is 133 with confidence: 0.0339192%
Prediction for test case 1 (282) is incorrect (should be 283)
= Prediction values for test #2
Top(1) prediction is 3 with confidence: 94.1774%
Top(2) prediction is 0 with confidence: 0.00358077%
Total time for 3 test cases: 4.955 seconds
Average time per test case: 1651.562 ms
```

Figure 22: Arm NN Object Recognition output screen

This is a demo output shows how much accuracy in detecting an Object like Cat, Dog, Shark. This will pick the image from data folder and provides the accuracy of respective image with this identity

Steps to run MIPI_Camera Demo:

Run **/run/media/mmcblk1p3/ARROW_DEMOS/run_ml_demos.sh** script and selec**t** option **8 ,**then select option **2**.

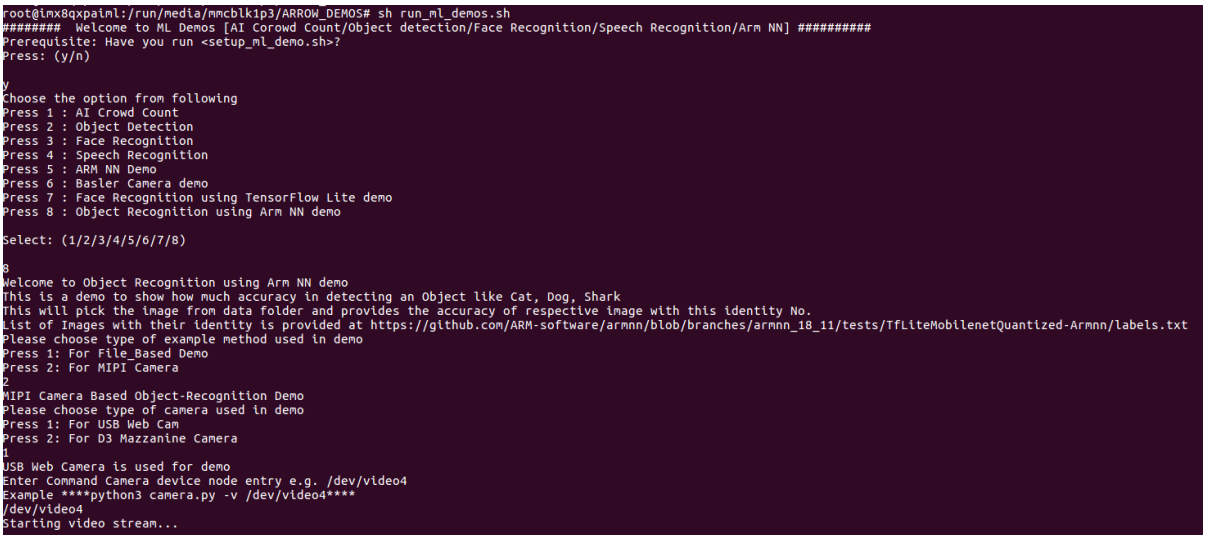

Figure 23: Arm NN Object Recognition using MIPI camera run screen

 # sh /run/media/mmcblk1p3/ARROW_DEMOS/run_ml_demos.sh

######### Welcome to ML Demos [AI Corowd Count/Object detection/Face Recognition/Speech Recognition/Arm NN] ############

Prerequisite: Have you run <setup_ml_demo.sh>?

Press: (y/n)

Y

Choose the option from following

Press 1 : AI Crowd Count

Press 2 : Object Detection

Press 3 : Face Recognition

.

.
Press 8 : Object Recognition using Arm NN demo

Select: (1/2/3/4/5/6/7/8)

8

Welcome to Object Recognition using Arm NN demo

This is a demo to show how much accuracy in detecting an Object like Cat, Dog, Shark

This will pick the image from data folder and provides the accuracy of respective image with this identity No.

List of Images with their identity is provided at https://github.com/ARM-software/armnn/blob/branches/armnn_18_11/tests/TfLiteMobilenetQuantized-Armnn/labels.txt

Please choose type of example method used in demo

Press 1: For File_Based Demo

Press 2: For MIPI Camera

2

MIPI Camera Based Object-Recognition Demo

Please choose type of camera used in demo

Press 1: For USB Web Cam

Press 2: For D3 Mazzanine Camera

1

USB Web Camera is used for demo

Enter Command Camera device node entry e.g. /dev/video4
Example ****python3 camera.py -v /dev/video4****

/dev/video4

Starting video stream...

Using Wayland-EGL

Using the 'xdg-shell-v6' shell integration

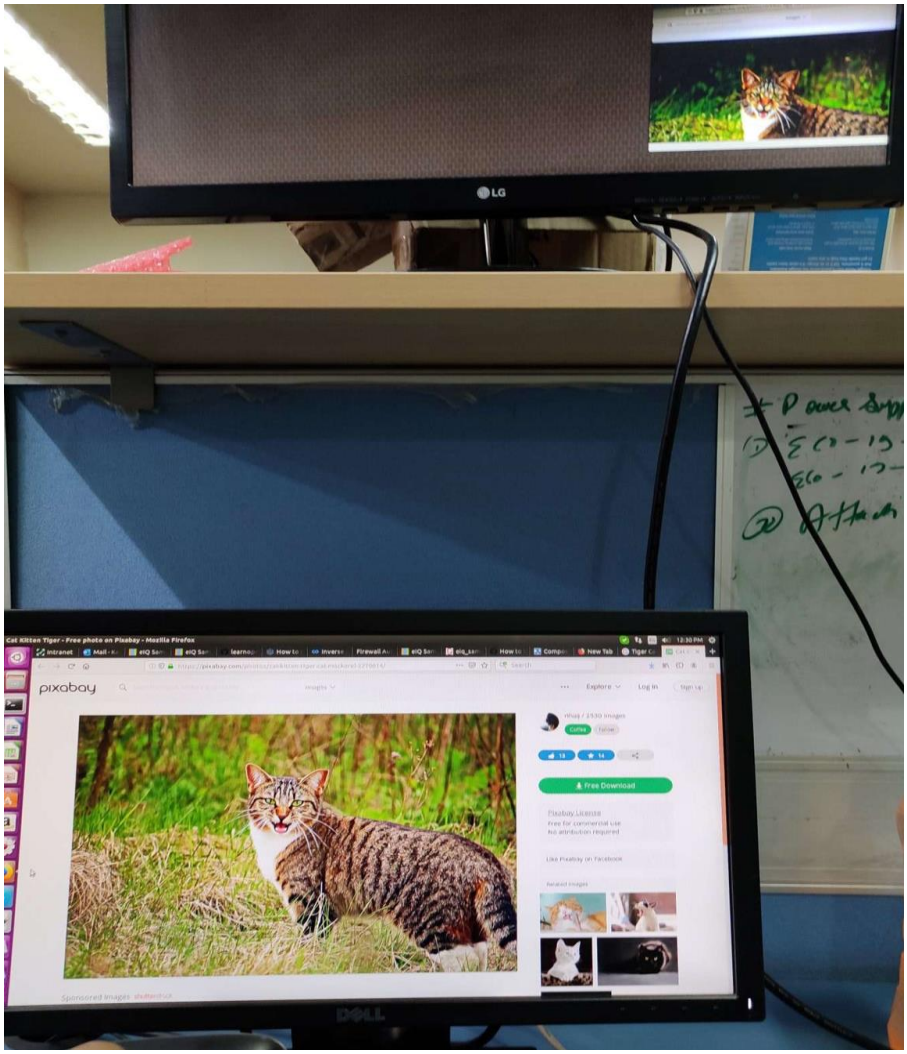

Figure 24: Arm NN Object Recognition using MIPI camera output screen

This runs the TfInceptionV3-Armnn test and parses the inference results to return any recognized object, not only the three expected types of animals.

Show the provided flash cards to the camera and wait for the detection message: Image captured, wait. The flash cards should not be twisted or curved on this step.

After a few seconds, the demo returns the detected object at this /opt/armnn/data/ path

TROUBLESHOOTING

➢ HDMI

- Although we have provided HDMI hot plug detection feature, we must connect HDMI before we boot up the board. Because we observed that if we do not connect HDMI before board boot up, hot plug feature not working and even after connecting HDMI we don't see any output on HDMI. We have to do reboot to get HDMI working.
- If we do not connect HDMI and run demo then sometimes we got below error.

```
****** Script Started *******
Setup is already completed. No need to do anything. Exiting...
Choose the option from following
Press 1: AI Crowd Count
Press 2: Object Detection
Press 3: Face Recognition
Press 4: Speech Recognition
Press 5: ARM NN Demo

Select: (1/2/3/4/5)

1
Welcome to AI Crowd Counting
This is a demo application using Python, QT, Tensorflow to be run on embedded devices for Crowd counting
You can choose Option for Live Mode (Camera)/Pre-captured Image Mode By clicking on GUI
Please choose type of camera used in demo
Press 1: For USB Web Cam
Press 2: For D3 Mazzanine Camera
2
D3 Mazzanine Camera is used for demo
ImportError: No module named 'numpy.core._multiarray_umath'
ImportError: No module named 'numpy.core._multiarray_umath'
NN> using tensorflow version:  1.3999999999999999
NN> using CPU-only (NO CUDA) with tensorflow
##### LOADING TENSORFLOW GRAPH
##### 160x120 used for live mode as these typically are close
##### 640x480 used for pre-captured large images of crowd as these are far
TF> using 640x480 resolution
Running on a compositor with no screens is not supported
Exiting Demo...
root@imx8qxpaiml:~#
```

Figure 25: No HDMI Connected Error

When we observed such error, we must connect HDMI and have to do reboot to work it.

➢ Camera

If we not connected any camera and start demo to capture frame then we might got error as shown in below attached figure.

To resolve camera error we can do following checks:

- Check correct node entry i.e. **"/dev/video7"** is provided while running demo in case of USB web cam.
- Please check that appropriate dtb file is set in uboot environment i.e. "**fsl-imx8qxp-aiml-mipi-ov5640.dtb**" in case of D3 camera.
- Please verify D3 camera or USB camera is working fine before running Demos. We can check that by gstreamer pipeline:
  gst-launch-1.0 v4l2src device=/dev/video0 ! video/x-raw,width=1280,height=720 ! glimagesink

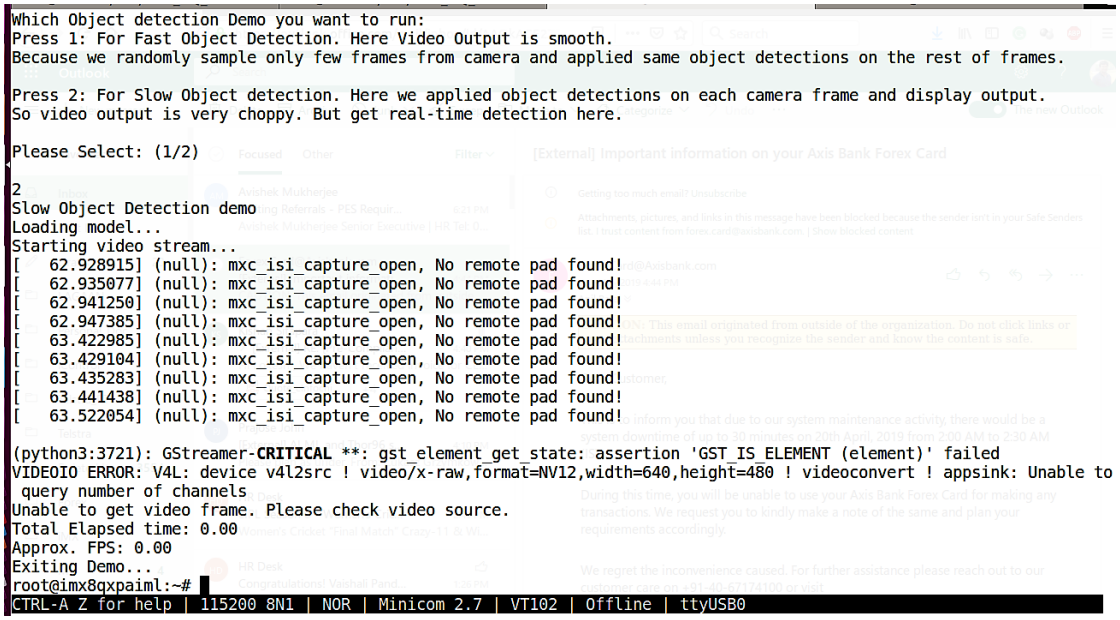

Figure 26: No Camera connected error

ML DEMOS REFERENCES

- https://github.com/ageitgey/face_recognition
- https://github.com/davisking/dlib
- https://cmusatyalab.github.io/openface/
- https://opencv-python-tutroals.readthedocs.io/en/latest/py_tutorials/py_objdetect/py_face_detection/py_face_detection.html#face-detection
- https://github.com/tensorflow/models/tree/master/research/object_detection
- https://www.tensorflow.org/tutorials/sequences/audio_recognition
- https://github.com/tensorflow/models
- https://github.com/tensorflow/models/blob/master/research/object_detection/g3doc/detection_model_zoo.md
- https://github.com/jrosebr1/imutils
- https://www.pyimagesearch.com/2018/06/18/face-recognition-with-opencv-python-and-deep-learning/
- https://github.com/chuanqi305/MobileNet-SSD
- https://github.com/Uberi/speech_recognition#readme
- https://medium.com/@ageitgey/quick-tip-speed-up-your-python-data-processing-scripts-with-process-pools-cf275350163a
- https://github.com/spmallick/learnopencv
- https://github.com/ARM-software/armnn
- https://www.pyimagesearch.com/2017/09/18/real-time-object-detection-with-deep-learning-and-opencv/
- https://github.com/opencv/opencv/tree/master/modules/dnn
- https://github.com/opencv/opencv/tree/master/samples/dnn
- https://heartbeat.fritz.ai/real-time-object-detection-on-raspberry-pi-using-opencv-dnn-98827255fa60
- https://github.com/ARM-software/ML-examples