User's Manual

PoseNet Summary Report

User's Manual: Software

RZ/G2E

All information contained in these materials, including products and product specifications, represents information on the product at the time of publication and is subject to change by Renesas Electronics Corp. without notice. Please review the latest information published by Renesas Electronics Corp. through various means, including the Renesas Electronics Corp. website (http://www.renesas.com).

Renesas Electronics
www.renesas.com

Rev.0.10 Aug, 2020

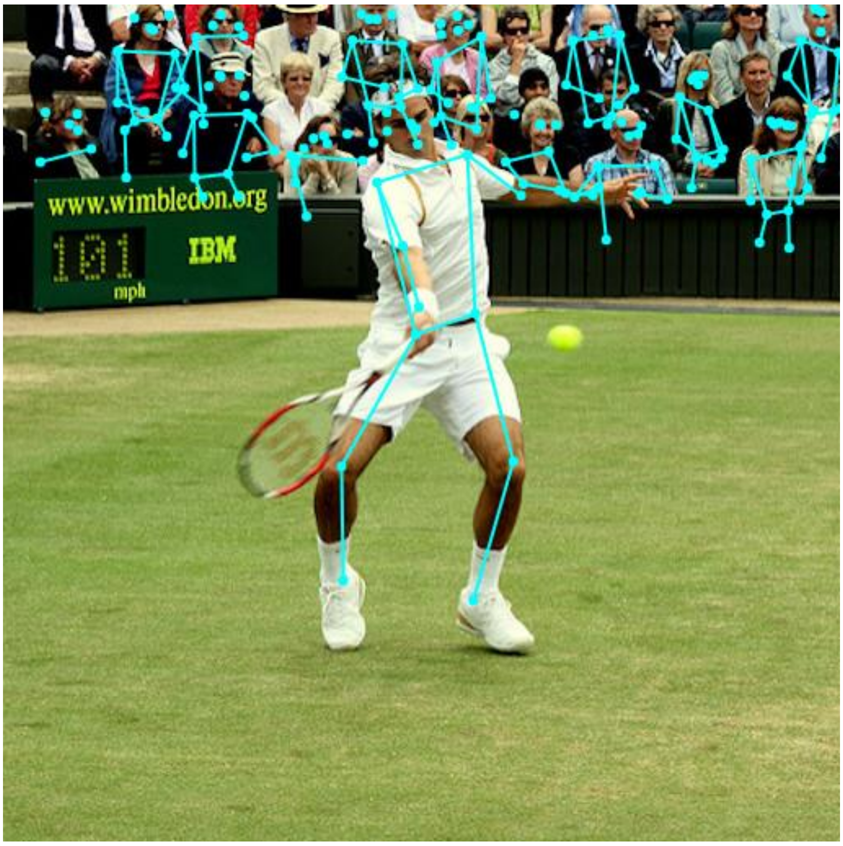
Table of Contents

1. Overview

1.1 Overview

Artificial intelligence is proliferating new business grounds with impressive success. One of the key factors contributing to AI’s effective implementation is the availability of agile and dynamic libraries like TensorFlow.

Posenet is one of TensorFlow applications. It is a vision model that can be used to estimate the pose of a person in an image or video by estimating where key body joints are.



Pose estimation has many uses, from interactive installations that react to the body to augmented reality, animation, fitness uses, and more.

1.2 Summary

This document give the detail report about:

- Prepare environment to build Posenet app on RZ/G2E board
- Implement Posenet on RZ/G2E board
- Output result
- Issue and restriction
- Future features

2. Terminology

The following table shows the terminology related to this module.

Table 2.1 Terminology

Terms	Explanation
BSP	Board support package
tf	Tensorflow
tflite	Tensorflow lite
tfjs	Tensorflow json
SDK	Software Development Kit

3. Operating environment

3.1 Background

This section lists all required hardware, software to build and run Posenet app.

3.2 Hardware Environment

The following table lists the hardware needed to use posenet app.

Table 3.1 Hardware Environment

Name	information
RZ/G2E EK874 board	Revision 3
RZ/G2E CPU	Arm Cortex A53 Dual MPCore 1.2 GHz
Camera output	Resolution: 640x480
Linux PC	Core i5, RAM 8GB

3.3 Software Environment

The following table lists software needed to use posenet app.

Table 3.2 Software Environment

Name	Version
onnx	1.7
onnxruntime	1.0.0 or 1.4.0
tflite2onnx	0.2.0
opencv	4.1.1
BSP for RZ/G2E board	1.0.4

3.4 Attached items

This document will attach below items:

Table 3.3 Attached Items

Name	summary
model_conversion.zip	model conversion tool
opencv_4.1.1.bb	Recipe file for opencv
0001-Update-maximum-number-of-supported-camera-from-8-to-.patch	Patch file for opencv
posenet_camera.zip	Source code of Posenet camera app
environment_conversion.txt	List of software and version
cam_output_129x129.avi	Result with input size 129x129
cam_output_257x257.avi	Result with input size 257x257
cam_output_513x513.avi	Result with input size 513x513

4. Model conversion

4.1 Background

PoseNet is application of TensorFlow. Currently, ONNX has no official model converted from TensorFlow, so need to use model conversion tool to convert tf model to onnx model.

4.2 Required software for model conversion tool

For running conversion tool, need to install required software on Linux PC.

Copy environment_conversion.txt file in attachment folder and run below command to install required software:

```
#pip install -r environment_conversion.txt
```

4.3 Working flow of model conversion tool:

- Download tfjs models (.json file)
- Convert tfjs models to tf models (.pb file)
- Convert tf models to tflite models (.tflite file)
- Convert tflite models to onnx models (.onnx file)

4.4 Usage of model conversion tool

4.4.1 Get instruction by running following command:

Command to get instruction as below:

```
#python3 convert.py --hint True
```

Currently, convert tool support below:

- model: mobilenet
- stride: 8 or 16
- Image size width = image size height, (width - 1) should be divisible by stride
  - Example: when we select stride = 16, width and height should be selected as below: width = height = X and (X - 1) % 16 should be 0, so we can choose X = 257 or 129 or 513.

4.4.2 Command for model conversion:

Command to convert model as below:

```
#python3 convert.py --model mobilenet --stride <stride> --tensor_size <1 x 3 x
width x height>
```

Example command with stride = 16, width = height = 129

```
#python3 convert.py --model mobilenet --stride 16 --tensor_size 49923
```

Example command with stride = 16, width = height = 257

```
#python3 convert.py --model mobilenet --stride 16 --tensor_size 198147
```

Example command with stride = 16, width = height = 513

```
#python3 convert.py --model mobilenet --stride 16 --tensor_size 789507
```

4.5 Output of model conversion tool:

After run conversion command, the output of this tool is onnx model

Example output with stride = 16, width = height = 129

- mobilenet_stride16_imagesize129.onnx

Example output with stride = 16, width = height = 257

- mobilenet_stride16_imagesize257.onnx

Example output with stride = 16, width = height = 513

- mobilenet_stride16_imagesize513.onnx

5. Yocto build environment for PoseNet App

5.1 Background

This section gives instruction to prepare environment to build and run Posenet app.

Building Yocto environment to run RZ/G2E board

Building SDK environment to build Posenet app

5.2 Package for Yocto environment

Prepare following directories with commit in Yocto directory

Table 6.1 Package for Yocto build

Package	commit
meta-gplv2	f875c60ecd6f30793b80a431a2423c4b98e51548
meta-linaro	75dfb67bbb14a70cd47afda9726e2e1c76731885
meta-openembedded	352531015014d1957d6444d114f4451e241c4d23
meta-qt5	c1b0c9f546289b1592d7a895640de103723a0305
poky	7e7ee662f5dea4d090293045f7498093322802cc
meta-rzg2	254953bd4f3f263f826d2208c82e7ad895bbaaa7
meta-renesas-ai	09461265862253e24215473cb8c66ea49c8ee09d

5.3 Patch file for opencv

When running RZ/G2E board with camera, the device name of camera is /dev/video8 (It is 9th video). Maximum number of supported camera on opencv version 4.1.1 is 8. So it get error while opencv cannot get the output of camera. Patch file is used to extend video device number.

Replace current opencv_4.1.1.bb in meta-renesas-ai/meta-rzg-ai/recipes-support/opencv by our opencv_4.1.1.bb include patch file for extending camera device number.

Note: opencv_4.1.1.bb and 0001-Update-maximum-number-of-supported-camera-from-8-to-.patch in the attached folder.

5.4 Build Yocto procedure

Run below commands to prepare environment for building Yocto and SDK:

```
# source poky/oe-init-build-env

# cp ../meta-renesas-ai/meta-onnxruntime/templates/ek874/* conf/

# ln -sf /data2/downloads
```

Update below variables in conf/local.conf

- BB_NO_NETWORK = "1“
- IMAGE_INSTALL_append = "onnxruntime-staticdev onnxruntime-dev onnxruntime-examples opencv opencv-samples libopencv-core libopencv-imgproc libopencv-photo libopencv-imgcodecs libopencv-videoio libopencv-highgui“

Run below commands to build Yocto and SDK:

```
#bitbake core-image-weston

#bitbake core-image-weston-sdk -c populate_sdk
```

5.5 Output environment

5.5.1 Image

Environment for G2E board in below directory:

- build/tmp/deploy/images/

5.5.2 SDK

SDK toolchain in below directory:

- build/tmp/deploy/sdk/poky-glibc-x86_64-core-image-weston-sdk-aarch64-toolchain-2.4.3.sh

Run the SDK toolchain and enter target directory for SDK environment

```
#./build/tmp/deploy/sdk/poky-glibc-x86_64-core-image-weston-sdk-aarch64-
toolchain-2.4.3.sh
```

Example for entering target directory: /data2/hoaphan/onnxruntime2/sdk

6. Build PoseNet app

6.1 Background

This section gives guideline to build Posenet app

6.2 Steps to build Posenet app

Extract the posenet_camera.zip to Linux PC

Source SDK environment before building posenet app

Example command to source SDK environment:

```
# source /data2/hoaphan/onnxruntime2/sdk/environment-setup-aarch64-poky-linux
```

Run below command to build posenet app:

```
#make
```

After building Posenet app, there is Posenet app as below:

- `/poseNet_camera`

7. Usage of PoseNet app

7.1 Background

This section gives guideline to use Posenet app

7.2 Required environment

7.2.1 Hardware environment:

RZ/G2E board (run with image and rootfs from 5.1.1)

Camera: All kind of camera that RZ/G2E can support. For detail information, refer 3.2

7.2.2 Software environment:

Copy Posenet app from step 6.2 to folder in RZ/G2E board

Example location: /posenet_camera/poseNet_camera

Copy onnx model from converting model step to folder in RZ/G2E board as below structure:

posenet_camera

|--------models/<onnx model>

|--------poseNet_camera

Example:

posenet_camera_cpp

|--------models/mobilenet_stride16_imagesize257.onnx

|--------poseNet_camera

7.3 Option run of PoseNet app

7.3.1 List all supported model of Posenet

To list all supported model of posenet, run below command:

```
# ./poseNet_camera -a
```

7.3.2 Instruction of posenet app

To get the instruction of posenet app, run below command:

```
# ./poseNet_camera -h
```

7.3.3 Select model architecture

```
-model <model architecture>
```

There are 2 model architecture:

- Mobilenet
- Resnet50 (not support)

Currently, Posenet app support mobilenet architecture only

7.3.4 Select model index

```
-model_index <model index>
```

Posenet app support 3 model as below:

Model 1:

Stride: 16

Input image size: 129x129

Input tensor size: 49923

Model 2:

Stride: 16

Input image size: 257x257

Input tensor size: 198147

Model 3:

Stride: 16

Input image size: 513x513

Input tensor size: 789507

7.3.5 Select camera index

```
-cam_index <camera index>
```

Check the device name of camera on board to update cam_index variable

Example: /dev/video<camera index>

7.3.6 Enable/disable measure time and print poses score

Base on demand, user can enable or disable measure time and print poses score as below variable:

```
-measure_time <option>
```

```
-print_poses_score <option>
```

<option> = 1 to enable

<option> = 0 to disable

7.3.7 Example command:

```
# ./poseNet_camera -model mobilenet -model_index 2 -cam_index 8 -
measure_time 0 -print_poses_score 0
```

8. Output result

8.1 Background

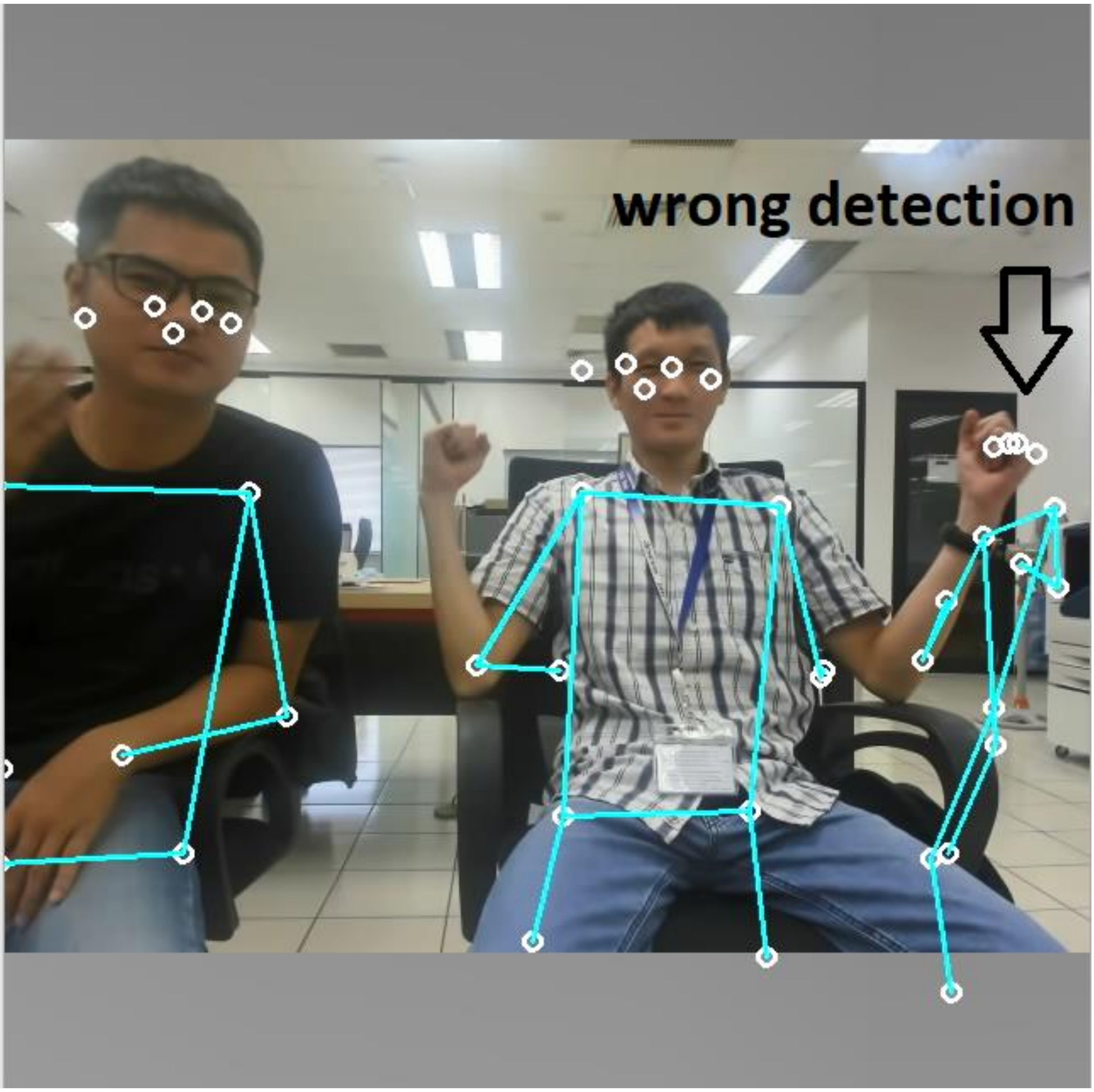
This section gives output of running Posenet app and the performance judgement.

8.2 Output

8.2.1 Output result of model 1 (MobileNet, stride: 16, image size: 129x129)

Video file: cam_output_129x129.avi

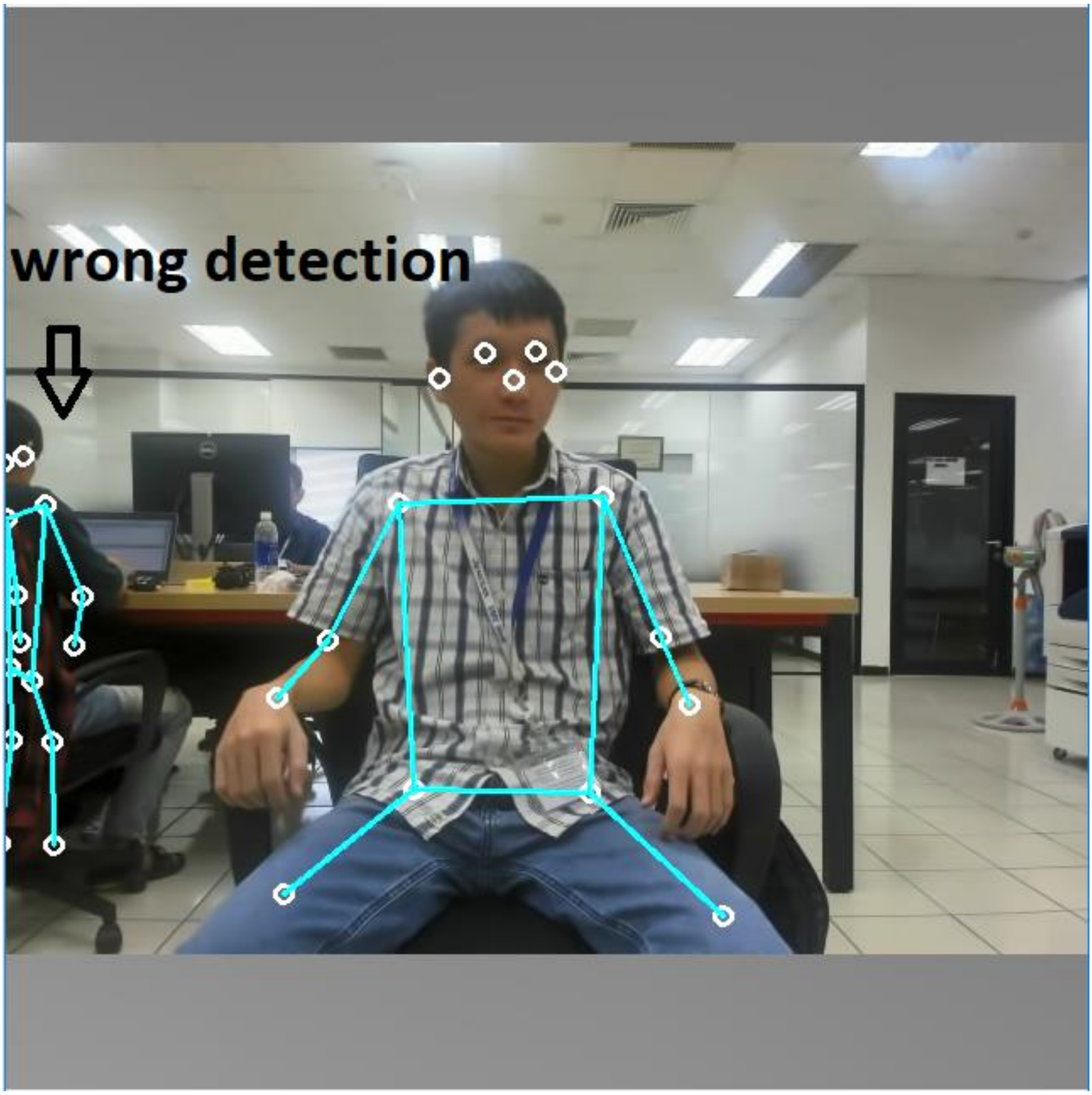
Image capture:



8.2.2 Output result of model 2 (MobileNet, stride: 16, image size: 257x257)

Video file: cam_output_257x257.avi

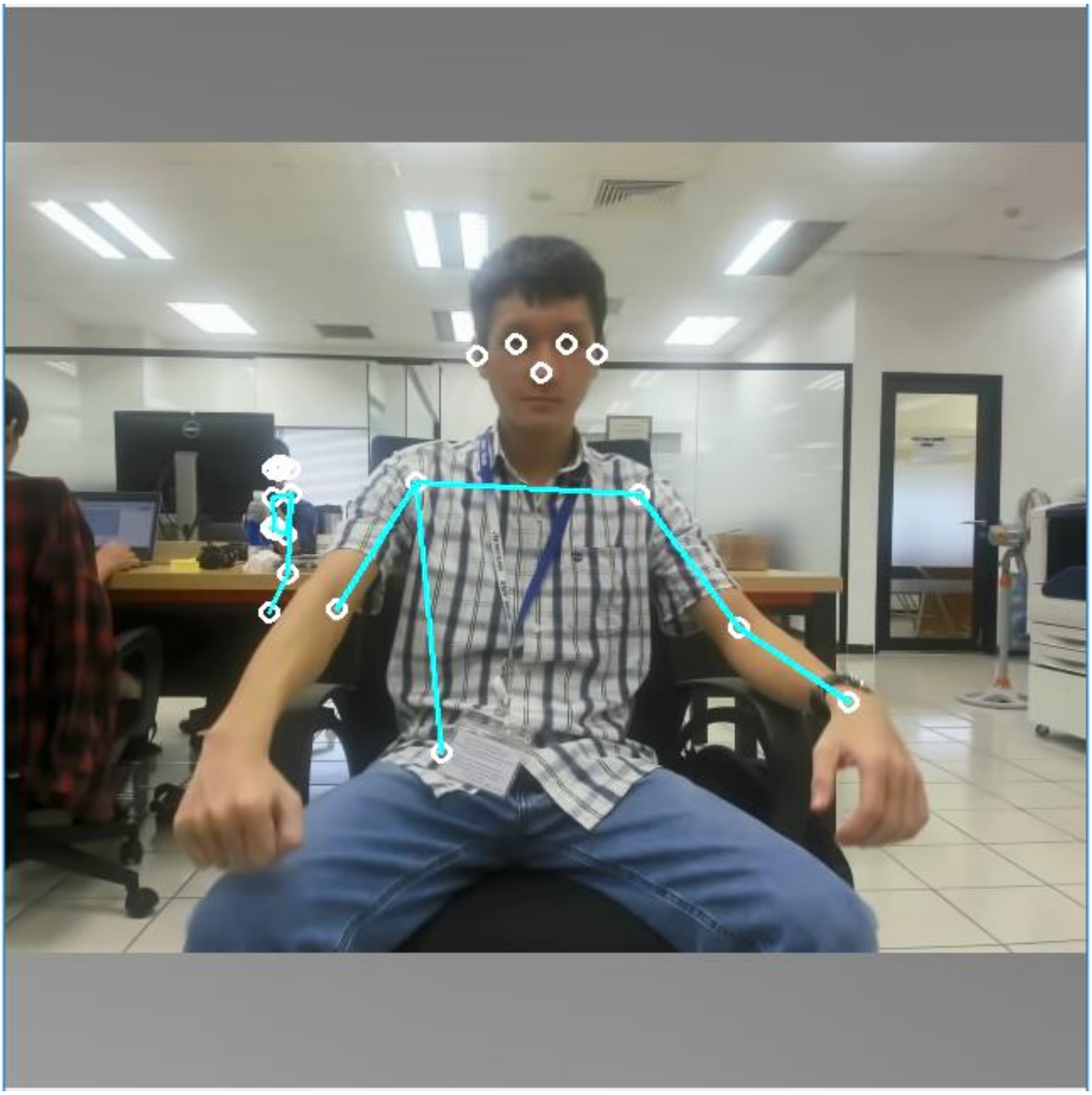
Image capture:



8.2.3 Output result of model 3 (MobileNet, stride: 16, image size: 513x513)

Video file: cam_output_513x513.avi

Image capture:



8.3 Performance measurement

Below table show performance report for posenet app with difference model

Table 9.1 performance

model	MobileNet (129 x 129)	MobileNet (257 x 257)	MobileNet (513 x 513)
Preprocessing (ms)	61.255	55.617	93.41
Run model (ms)	239.39	793.026	2784.874
Postprocessing (ms)	29.616	23.624	30.34
Total (ms)	330.261	872.267	2908.624
Frame per second (FPS)	3.02790823	1.14643796	0.343805181
Speed	high	mid	low
Correctness	low	mid	high

9. Issues and solutions

9.1 Background

This section lists all issues in the development process of Posenet app.

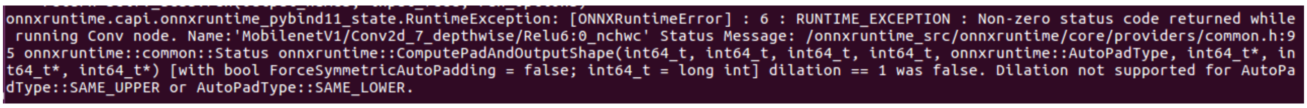
9.2 Can not convert TensorFlow models with dynamic shape

Description: Can't convert Tensor Flow model with dynamic shape to ONNX model. In this case, converted models architecture are MobileNet and ResNet.

Working flow of model conversion:

- Get tfjs model from TF hub (file .json)
- Convert tfjs to saved model format (file .pb)
- Use tfonnx tool to convert saved model to onnx format (https://github.com/onnx/tensorflow-onnx)
- Use ONNX Runtime to load and run converted model.

Issue: Get error in class ONNX Runtime::ComputePadAndOutputShape



```
onnxruntime.capi.onnxruntime_pybind11_state.RuntimeException: [ONNXRuntimeError] : 6 : RUNTIME_EXCEPTION : Non-zero status code returned while
 running Conv node. Name:'MobilenetV1/Conv2d_7_depthwise/Relu6:0_nchwc' Status Message: /onnxruntime_src/onnxruntime/core/providers/common.h:9
5 onnxruntime::common::Status onnxruntime::ComputePadAndOutputShape(int64_t, int64_t, int64_t, int64_t, onnxruntime::AutoPadType, int64_t*, in
t64_t*, int64_t*) [with bool ForceSymmetricAutoPadding = false; int64_t = long int] dilation == 1 was false. Dilation not supported for AutoPa
dType::SAME_UPPER or AutoPadType::SAME_LOWER.
```

Root cause: there are some limitations in conversion between Tensorflow and ONNX, especially for dynamic shape

Solution: Set input shape of model to fixed value. Dynamic input shape model is changed to static input shape model.

Example:

Table 10.1 input shape

	Model with dynamic shape	Model with static shape	Note
Input shape	[1, -1, -1, 3]	[1, 513, 513, 3]	The value -1 will be replace with dynamic value when running model
	[1, -1, -1, 3]	[1, 257, 257, 3]	
	[1, -1, -1, 3]	[1, 129, 129, 3]	

Working flow of model conversion with Solution:

- Get tfjs model from TF hub (file .json)
- Convert tfjs to saved model format (file .pb)
- Convert saved model format to TFLite with input shape specified [1, 513, 513, 3] or [1, 257, 257, 3] or [1, 219, 219, 3]
- Use tflite2onnx Python library (https://pypi.org/project/tflite2onnx) to convert TFLite to ONNX format
- Use ONNX Runtime to load and run converted model.

Note:

- We can't use tf2onnx due to it requires saved model format (.pb) as input. But saved model format can't set input shape with specified.

- The counter measure just apply for MobileNet. About ResNet we have another issue described in next page (10.2).

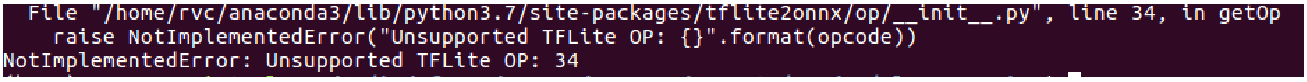
9.3 Can not convert ResNet model with static shape

Description: Can’t convert ResNet model with static shape

Working flow of model conversion with counter measure:

- Get tfjs model from TF hub (file .json)
- Convert tfjs to saved model format (file .pb)
- Convert saved model format to TFLite with input shape specified [1, 513, 513, 3] or [1, 257, 257, 3] or [1, 219, 219, 3]
- Use tflite2onnx Python library (https://pypi.org/project/tflite2onnx) to convert TFLite to ONNX format
- Use ONNX Runtime to load and run converted model.

Issue: Get error when convert TFLite to ONNX use tflite2onnx.



```
  File "/home/rvc/anaconda3/lib/python3.7/site-packages/tflite2onnx/op/__init__.py", line 34, in getOp
    raise NotImplementedError("Unsupported TFLite OP: {}".format(opcode))
NotImplementedError: Unsupported TFLite OP: 34
```

Root cause: The model conversion tool does not support TFLite operations

Solution: Currently, we have not converted ResNet to ONNX yet. We also skip it due to low priority. In the future, if we need to convert ResNet to ONNX. There are 2 potential solutions:

1) Waiting the new version of convert tool tflite2onnx

2) Try use other format such as Keras (.hdf5) , graph model instead of TFLite.

10. Restrictions

10.1 Background

This section lists all restrictions of Posenet app.

10.2 Model architecture

Description: PoseNet app doesn’t support ResNet architecture:

Root cause: ResNet can't be run by ONNX runtime after converting TF to ONNX. Because ONNX Runtime has not supported some operations as TF yet.

10.3 Input image shape

Description: Posenet app doesn’t support dynamic image shape

Image shape: (n image, Channel , Height, Weight)

Root cause: Currently, we can convert TF model to onnx model with dynamic shape but getting errors when running by ONNX Runtime. So we work around by changing dynamic image shape model to static image shape model: (1, 3, 513, 513), (1, 3, 257, 257) and (1, 3, 129, 129).

10.4 Input stride

Description: Mobilenet architecture supports 3 stride: 8, 16, 32. But model conversion tool supports 2 stride: 8 and 16, not support stride 32

Root cause: there are no tfjs model with stride 32.

11. Future features

11.1 Background

Currently, Posenet app have some issues and restrictions. So these issues and restrictions will be handle in the future.

11.2 Future features

In the future, the Posenet app will be update below features:

- Support dynamic image shape
- Support Resnet architecture
- Support all stride 8, 16, 32

12. Reference

Posenet app use below references:

- https://www.tensorflow.org/lite/models/pose_estimation/overview
- https://github.com/tensorflow/tfjs-models/tree/master/posenet
- https://github.com/atomicbits/posenet-python

REVISION HISTORY	Posenet summary report User’s Manual: Software

Rev.	**Date**	**Description**	
		Page	**Summary**
0.10	Aug. 26, 2020	-	First Edition issued

PoseNet summary report
User’s Manual: Software

Publication Date: Rev.0.10 Aug 26, 2020

Published by: Renesas Electronics Corporation

PoseNet Summary Report

Renesas Electronics Corporation