CVGesture

Performance Report

2018-01-25

OPEN AI LAB

Revision Record

Date	Rev	Change Description	Author
2017-10-13	0.1.0	QA Testing	Luo Hao
2017-12-25	0.2.0	QA Testing	Luo Hao
2018-01-25	0.3.0	QA Testing	Luo Hao

catalog

1 Introduction

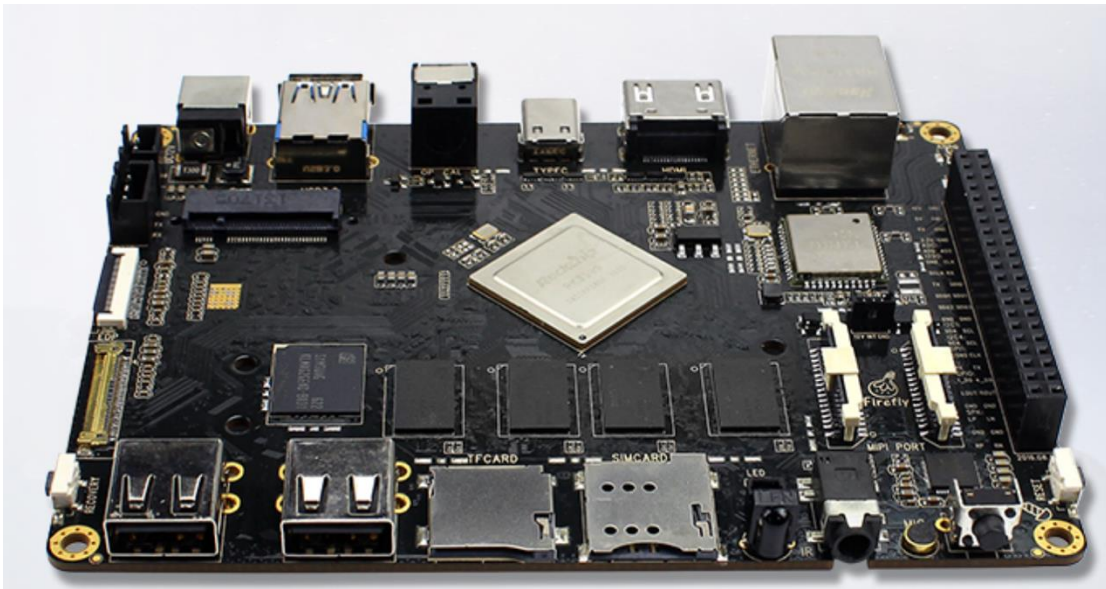This Report is tested on RK3399 platform. The report includes performance and accuracy information.

2 Test Environment

Hardware SoC: firefly

http://www.t-firefly.com/product/rk3399.html

- ➢ GPU: Mali T864 (800MHz)
- ➢ RAM: 4G
- ➢ CPU: Dual-core Cortex-A72 up to 2.0GHz (real frequency is 1.8GHz); Quad-core Cortex-A53 up to 1.5GHz (real frequency is 1.4GHz)

Operating System: Ubuntu 16.04

Software: OpenCV 3.3.0

3 Definition of Gestures

In order to test the performance of the application, there should be some definitions and limitations to different gestures:

- ● Palm: with five fingers open (do not close five fingers together, as fig.3), facing directly to the camera (as fig.1). The angle of rotation of hand in the direction of front and back should not go beyond 30 degrees. The angle of rotation of hand in the direction of left and right should not go beyond 45 degrees. The standard is the same for both left and

right hand.

● Fist: with fist clenched, facing directly to the camera (as fig.2). The angle of rotation of hand in the direction of front, back, left and right should not go beyond 30 degrees. The standard is the same for both left and right hand.

Figure 1. Palm facing directly to camera

Figure 2. Fist facing directly to camera

Figure 3. Wrong palm gesture
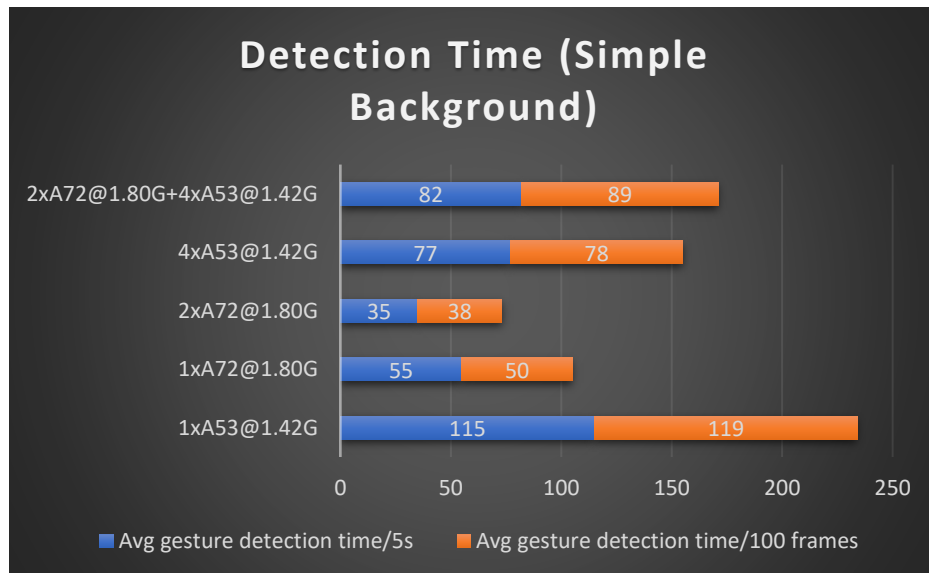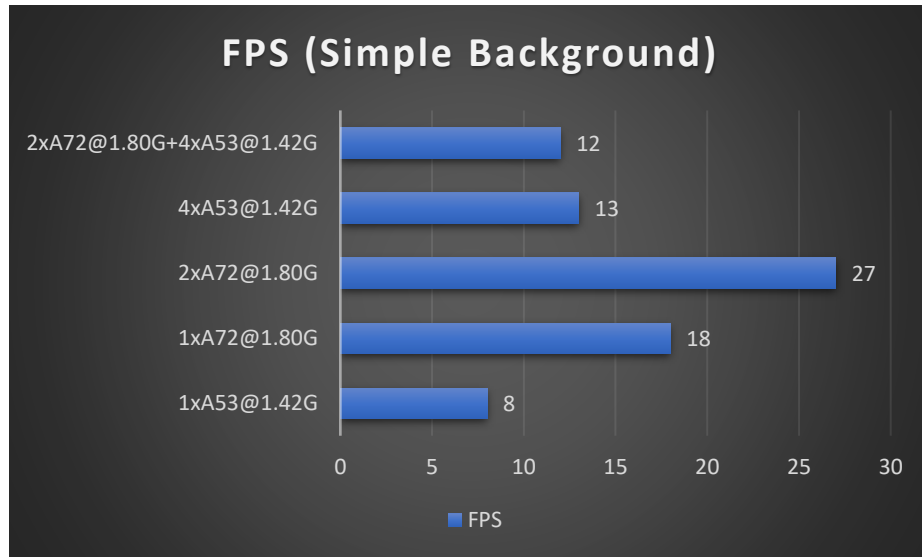
4 Performance on Different Cores

Calculate the FPS(Frame rate Per Second) in five seconds, and print the result in terminal. Skip the result of first five seconds. Recognize two gestures: palm and fist. The palm and fist detection time are also averaged in five seconds and 100 frames, skip the result of first five seconds. As the complexity of background will significantly affect the performance, the performance under different backgrounds will be given.

4.1 Simple background

Simple background means that the background does not contain object of irregular shape like a plain wall. Performance under simple background will be stronger.

FPS on different cores, 640x480 resolution

	FPS	Avg. gesture detection time (ms) in 5 seconds	Avg gesture detection time (ms) in 100 frames
1xA53@1.42G	8	115	119
1xA72@1.80G	18	55	50
2xA72@1.80G	27	35	38
4xA53@1.42G	13	77	78
2xA72@1.80G+4xA53@1.42G	12	82	89

FPS (Simple Background)

Configuration	FPS
2xA72@1.80G+4xA53@1.42G	12
4xA53@1.42G	13
2xA72@1.80G	27
1xA72@1.80G	18
1xA53@1.42G	8

Detection Time (Simple Background)

| Configuration	Avg gesture detection time/5s	Avg gesture detection time/100 frames
2xA72@1.80G+4xA53@1.42G	82	89
4xA53@1.42G	77	78
2xA72@1.80G	35	38
1xA72@1.80G	55	50
1xA53@1.42G	115	119

4.2 Complex background

Complex background means that the background contains many objects of irregular shape. Performance under complex background will be weaker.

FPS on different cores, 640x480 resolution

	FPS	Avg. gesture detection time (ms) in 5 seconds	Avg. gesture detection time (ms) in 100 frames
1xA53@1.42G	6.8	145	150
1xA72@1.80G	12	81	87

|			
2xA72@1.80G	18	54	52
4xA53@1.42G	12	81	85
2xA72@1.80G+4xA53@1.42G	12	79	86
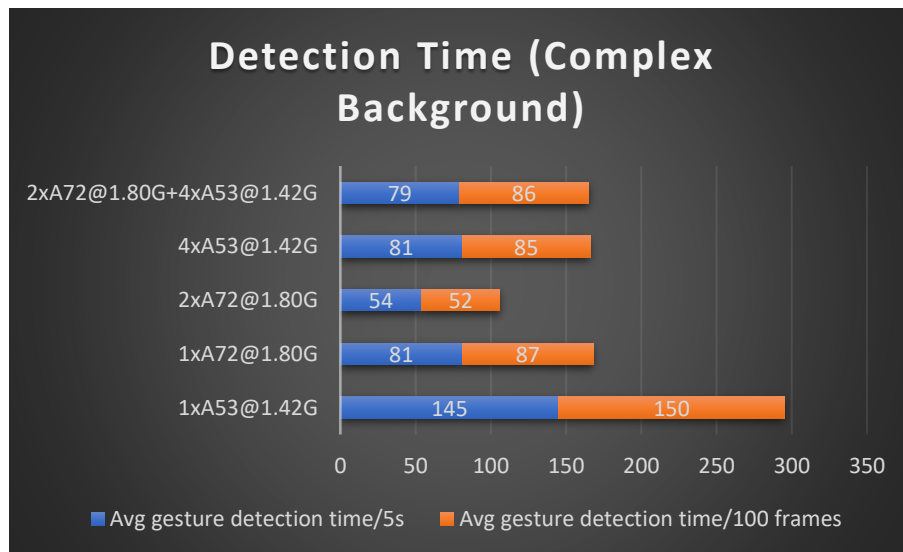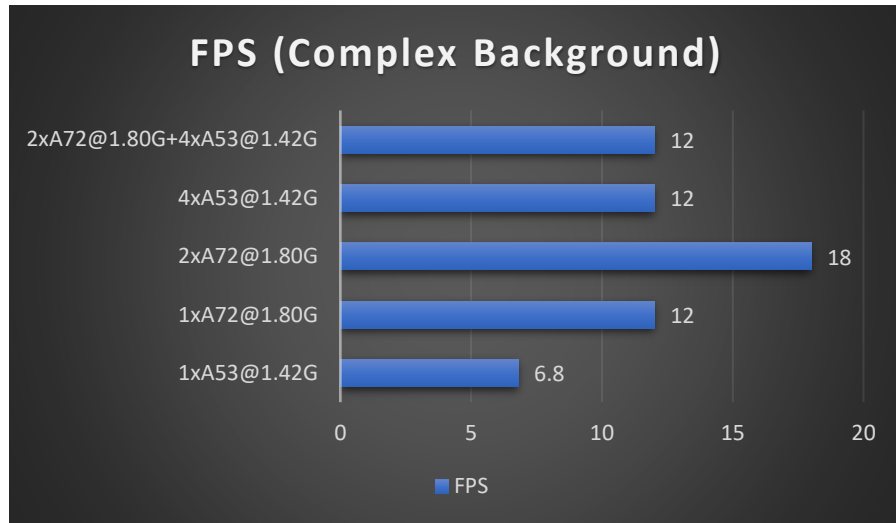
5 Accuracy test

To test accuracy of the classifiers, a small number of images of palms and fists are collected from the internet and labeled with ground truth. Following indices are used to evaluate detection and recognition accuracy:

For detection:

- IoU: Intersection over Union, intersection means the overlap area of detection and ground truth and union mean the union area of detection and ground truth (as shown below)

$$IoU = \frac{\text{Area of Overlap}}{\text{Area of Union}}$$
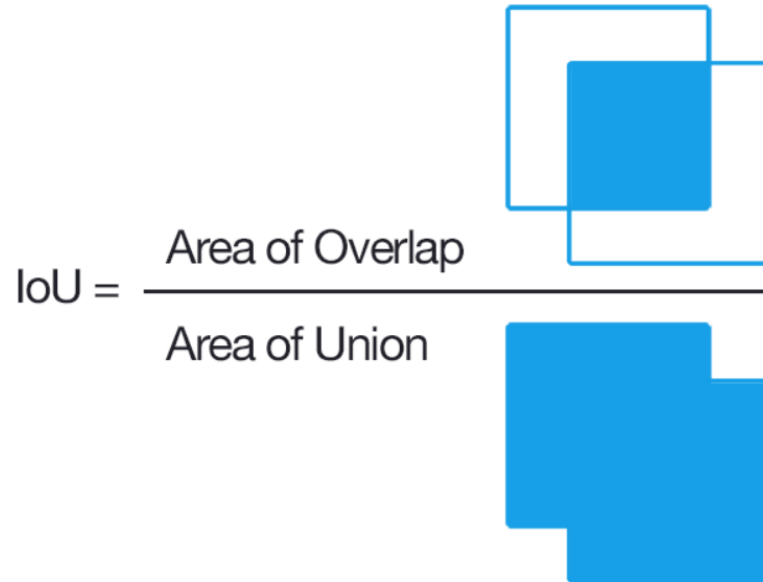
For recognition (as shown below):

- Precision: Ture positives / (Ture positives + False positives)
- Recall: Ture positives / (Ture positives + False negatives)
- Accuracy: (Ture positives + Ture negatives) / Total
- Ture positives: IoU of recognition >= 0.5 and the label of class is correct

True class

		p	n
Y		True Positives	False Positives
Hypothesized class			
N		False Negatives	True Negatives
Column totals:		P	N

$$\text{fp rate} = \frac{FP}{N} \qquad \text{tp rate} = \frac{TP}{P}$$

$$\text{precision} = \frac{TP}{TP+FP} \qquad \text{recall} = \frac{TP}{P}$$

$$\text{accuracy} = \frac{TP+TN}{P+N}$$

$$\text{F-measure} = \frac{2}{1/\text{precision}+1/\text{recall}}$$

The accuracy results of the classifiers are given below, the results are averaged from 20 different confidence level (setting *minNeighbors* parameter in *cv.detectMultiScale* from 1 to 20) and the min:

Avg. IoU	Avg. Precision	Avg. Recall	Avg. Accuracy
0.75	0.90	0.70	0.81

The ROC curve and PR curve are also given below:
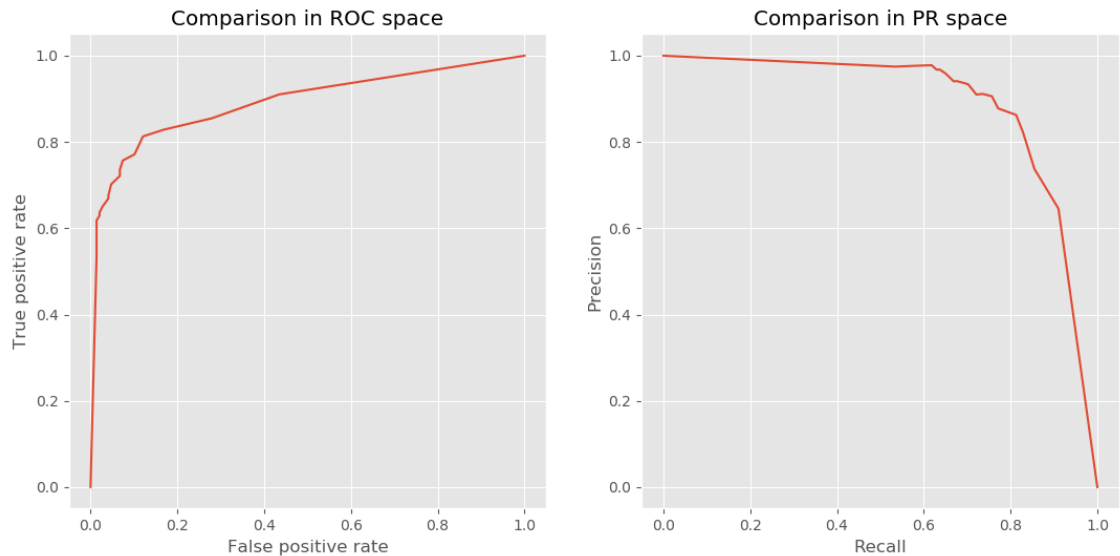
6 Real case accuracy
6.1 Factors affect accuracy

The detect and recognition function used in OpenCV is shown as below:

```
void CascadeClassifier::detectMultiScale( const Mat& image, vector<Rect>&
objects,
                                          double scaleFactor, int minNeighbors,
                                          int flags, Size minObjectSize, Size
maxObjectSize)
```

Of all the parameters, 3 parameters will affect detection accuracy: scaleFactor, minNeighbors and minObjectSize. 2 of the 3 parameters will affect detection speed at the same time: scaleFactor and minObjectSize. To achieve real-time detection speed, scaleFactor is often set to 1.1 and minObjectSize is set to 80 or 90.

6.2 Ideal case

Test in chapter 5 is using following parameter settings:

ScaleFactor: 1.1, minObjectSize: 20

6.3 Real case

Test is using following parameter settings:

ScaleFactor: 1.1, palm.minObjectSize: 90, fist.minObjectSize: 80, minNeighbors: 7

Avg. IoU	Avg. Precision	Avg. Recall	Avg. Accuracy
0.77	0.94	0.56	0.77

As some of the images in test set is small, using such minObjectSize parameter cannot achieve best accuracy result.

7 Conclusion

From the above test cases, we can deduce that :

- the performance on 2xA72 is the best
- the performance on 4xA53+2xA72 is similar versus 4xA53

The algorithm should run on A72 or 4xA53, single A53 core cannot meet the performance requirement.

8 Testing Issues

There are many known factors that will significantly affect the performance of the application:

- The **version of OpenCV should be 3.3.0**, using OpenCV2 will slow down the detection
- The performance of the application depends on the complexity of background, running under **simple background like a plain wall** will be significantly faster