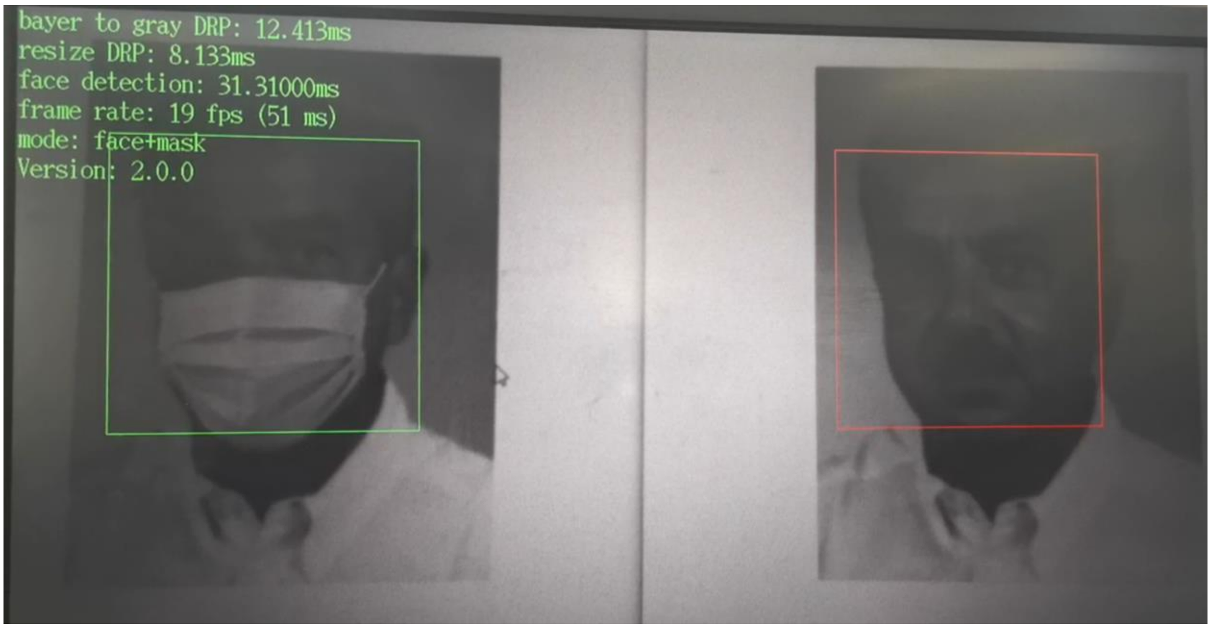
RZ/A2M

Face Detection Application Note



```
bayer to gray DRP: 12.413ms
resize DRP: 8.133ms
face detection: 31.31000ms
frame rate: 19 fps (51 ms)
mode: face+mask
Version: 2.0.0
```

Introduction

This document describes the RZ/A2M Face detection demo software.

In the example, we connect the Sony IMX219 CMOS sensor through MIPI interface, input a 1280x720 resolution image, and use DRP in RZ/A2M to perform Simple ISP processing and image scaling processing on the input image, and then run a light and efficient mask face detection model. It can achieve detect speed of 30FPS in the face detection mode, and a detection speed of 20FPS in the mode of distinguishing whether to wear a mask.

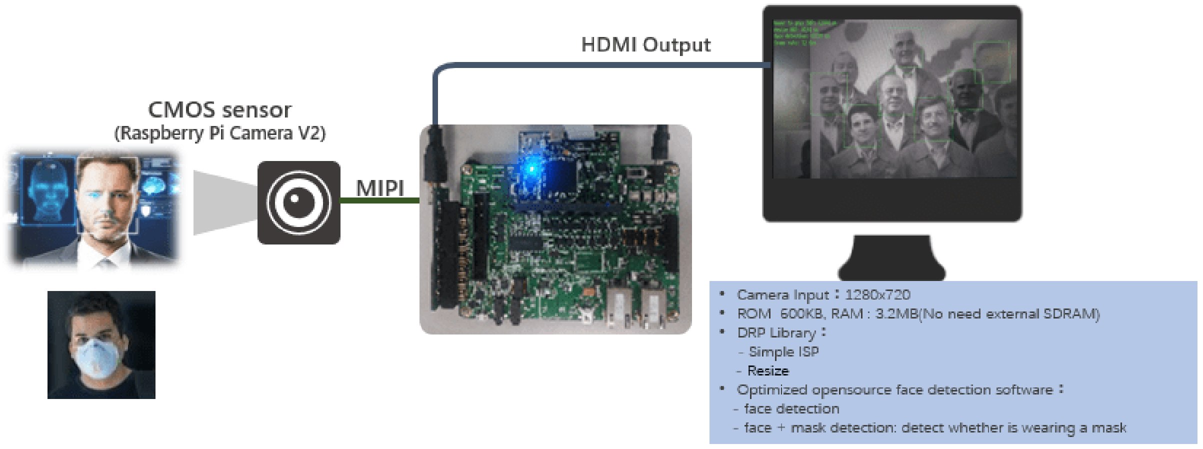
Target Device

RZ/A2M

Contents

1. Overview

The demo detects face or mask taken by a camera. The result is shown on a display (HDMI monitor).



- Camera Input : 1280x720
- ROM 600KB, RAM : 3.2MB(No need external SDRAM)
- DRP Library :
  - Simple ISP
  - Resize
- Optimized opensource face detection software :
  - face detection
  - face + mask detection: detect whether is wearing a mask

The Face detection demo software is available for 2 different platforms, please refer to below link for general information about the EVK and GR-MANGO platform:

1. RZ/A2M Evaluation Board Kit
   https://www.renesas.com/eu/en/products/microcontrollers-microprocessors/rz-cortex-a-mpus/rza2m-evaluation-kit-rza2m-evaluation-kit
2. Gadget Renesas board "GR-MANGO"
   https://www.renesas.com/eu/en/products/gadget-renesas/boards/gr-mango

Used e²studio version: e²studio 2021.04 (2021-04 (21.4.0))

In this sample, we provided 2 different e2 studio projects accordingly.
e²studio project name: "rza2m_face_detection_freertos_gcc_evk"
e²studio project name: "rza2m_face_detection_freertos_gcc_grmango"

There are a few platform-dependant differences. These are marked with the e²studio project name. The user switches are used to select different options:

	rza2m_face_detection_freertos_gcc_evk	rza2m_face_detection_freertos_gcc_grmango
select the image and information shown at the display (*1)	SW3	SW2
reset	n.a.	SW1

(*1) Three different face detection modes
1. Face+Mask: Run two AI model, one is Face detection, another one is Mask detection
2. Face: Only Run face detection
3. Mask: Only run mask dedection

2. Operation Confirmation Conditions

2.1. Evaluation Board Kit setup

Please setup the development kit board as described in

rza2m_face_detection_freertos_gcc_evk/doc/readme-e.txt

Especially the setting for DIP switches and jumpers needs to be followed.

Additional information can be found in the:

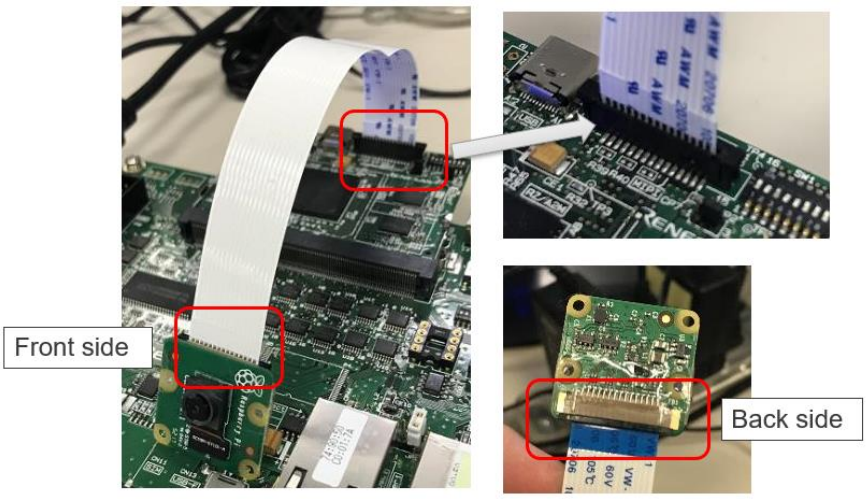
rza2m_face_detection_freertos_gcc_evk/doc/ r01qs0027ej0108-rza2m-quick-guide-gcc.pdf

rza2m_face_detection_freertos_gcc_evk/doc/r20ut4535ej0102-e2studio.pdf

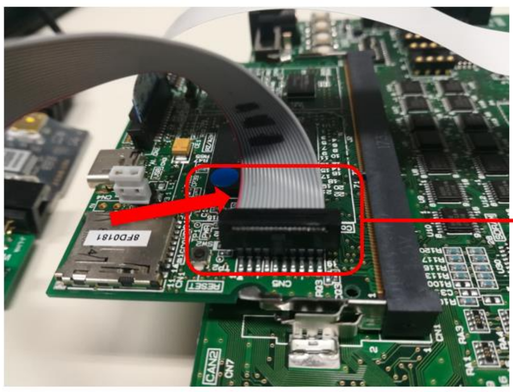
.

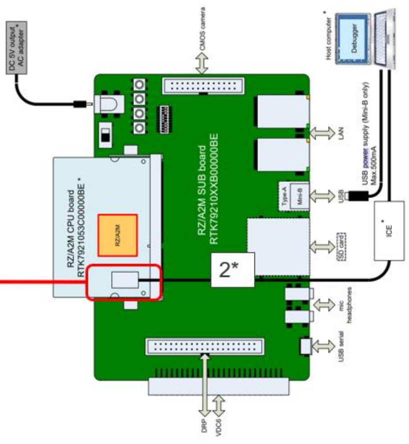
2.1.1. Evaluation Board Kit setup (quick start)

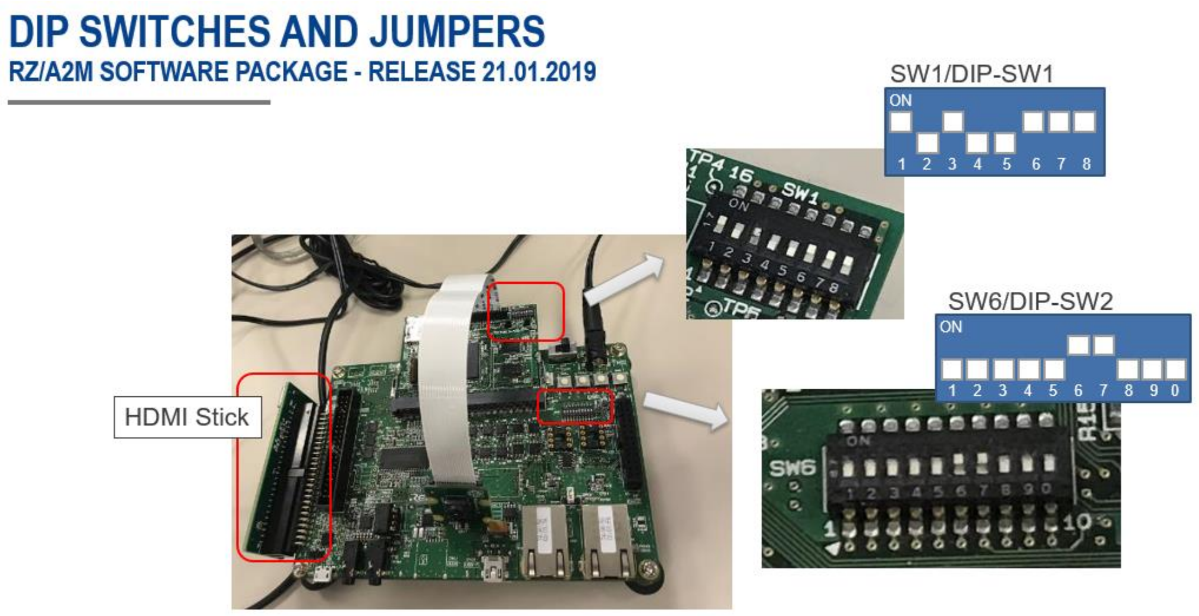
MIPI CAMERA CONNECTION



J_LINK CONNECTION





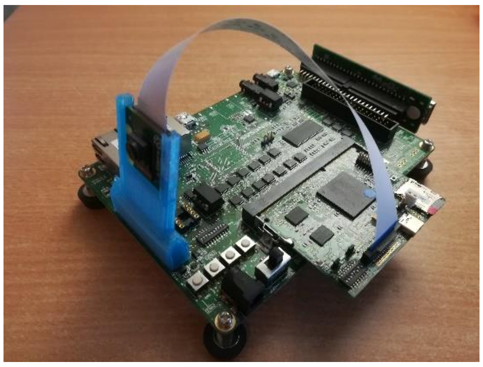


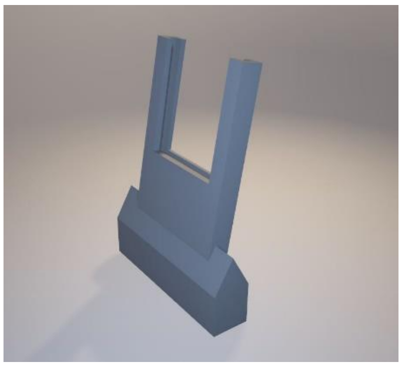
2.1.2. Raspberry PI camera holder (3D model)

The following camera holder is quite useful to stabilize the camera.

RZ/A2M development kit gadget: Raspberry PI camera holder (3D model)

https://renesasrulz.com/rz/m/rz_files/3399





2.2. GR-MANGO board setup

Please setup the development kit board as described in
rza2m_face_detection_freertos_gcc_grmango/doc/readme-e.txt

"(4) Downloading sample code

Connect GR-MANGO CN1 and PC via USB cable.
PC detects the GR-MANGO as MBED drive.
Drag-and-drop the binary file
(e.g. rza2m_face_detection_freertos_gcc_grmango/HardwareDebug/
rza2m_face_detection_freertos_gcc_grmango.bin
or pre-compiled binary
rza2m_face_detection_freertos_gcc_grmango/bin/
rza2m_face_detection_freertos_gcc_grmango.bin)
to the MBED drive.
(5) Executing sample code

Connect the camera to GR-MANGO CN13 and connect display to CN9 via HDMI cable.

Ensure that complete to download program, and push reset button on GR-PEACH."

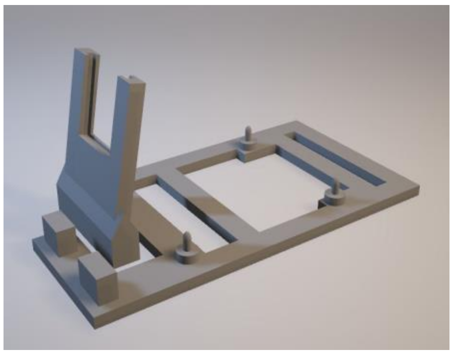
Additional information can be found in the:
rza2m_face_detection_freertos_gcc_grmango/doc/r01an5595ej0200-rza2m-swpkg-grmango-gcc.pdf
rza2m_face_detection_freertos_gcc_grmango/doc/ r01qs0042ej0101-rza2m-quick-guide-grmango-gcc.pdf

2.2.1. Raspberry PI camera holder (3D model)

The following camera holder is quite useful to stabilize the camera.

RZ/A2M GR-MANGO board gadget: board mount with Raspberry PI camera holder (3D model)

https://renesasrulz.com/rz/m/rz_files/3402



3. Software

3.1. Folder Structure

```
rza2m_face_detection_freertos_gcc_evk or
rza2m_face_detection_freertos_gcc_grmango
+---bootloader
+---doc
+---FlashTools
+---generate
|   +---compiler
|   +---configuration
|   +---drivers
|   +---gr_mango_boot         (RZA2M_GR_MANGO only)
|   +---os_abstraction
|   +---sc_drivers
|   |   +---r_cbuffer
|   |   +---r_ceu
|   |   +---r_drp
|   |   |   +---doc
|   |   |   +---drp_lib_custom
|   |   |   +---drp_lib
|   |   |   +---inc
|   |   |   \---src
|   |   +---r_mipi
|   |   +---r_ostm
|   |   +---r_riic
|   |   +---r_rvapi
|   |   +---r_scifa
|   |   \---r_vdc
|   \---system
\---src
    +---config_files
    +---FreeRTOS
    +---renesas
    |   +---application
    |   |   +---common
    |   |   |   +---camera
    |   |   |   +---perform
    |   |   |   +---port_settings
    |   |   |   \---render
    |   |   +---inc
    |   |   |   +---camera
    |   |   |   \---lcd
    |   \---hwsetup
    \---user_prog
+---pico-master
|   +---gen
|   |   +---sample
|   |   |   +---caltechfaces
|   |   |   +---caltechfaces_mask
|   |   |   \---mask
|   +---rnt
|   |   +---cascades
|   |   \---samples
```

3.2. Display type definition

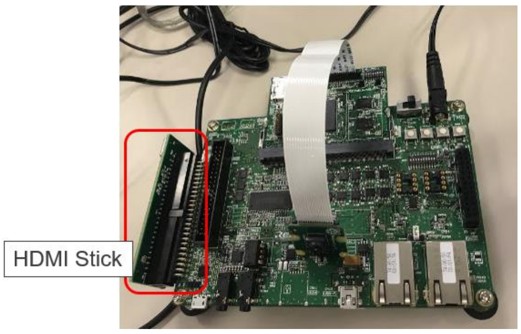
The RZ/A2M Evaluation Board Kit offers different options to connect a display.
To change the display type, please open header file

rza2m_face_detection_freertos_gcc_evk/src/renesas/application/inc/lcd_panel.h

and map LCD_PANEL with one of the pre-defined options (e.g. LCD_PANEL_RSK or LCD_PANEL_DVI).

LCD_PANEL_RSK is the label for a Renesas touch display that can be connected to the Evaluation board kit directly.

LCD_PANEL_DVI is the label for HDMI displays that can be connected via the HDMI interface stick that needs to be connected to the development board.



RZ/A2M Evaluation Board Kit and GR-MANGO board:

If LCD_PANEL_DVI (HDMI) is selected, please check header file

rza2m_face_detection_freertos_gcc_evk/src/renesas/application/inc/lcd/pc_monitor.h

and modify the define for SELECT_MONITOR to select the most suitable configuration for the connected monitor.

3.3. DRP libraries

The Software uses a mixture of released DRP libraries and modified DRP libraries that have been developed at Renesas Electronics China.

The libraries are given in

/rza2m_barcode_type_freertos_gcc_*/generate/sc_drivers/r_drp

/rza2m_barcode_type_freertos_gcc_*/generate/sc_drivers/r_drp/drp_custom

Please check the documents in sub-directories

r_drp/doc,

for detailed information.

4. Image Pre-Processing

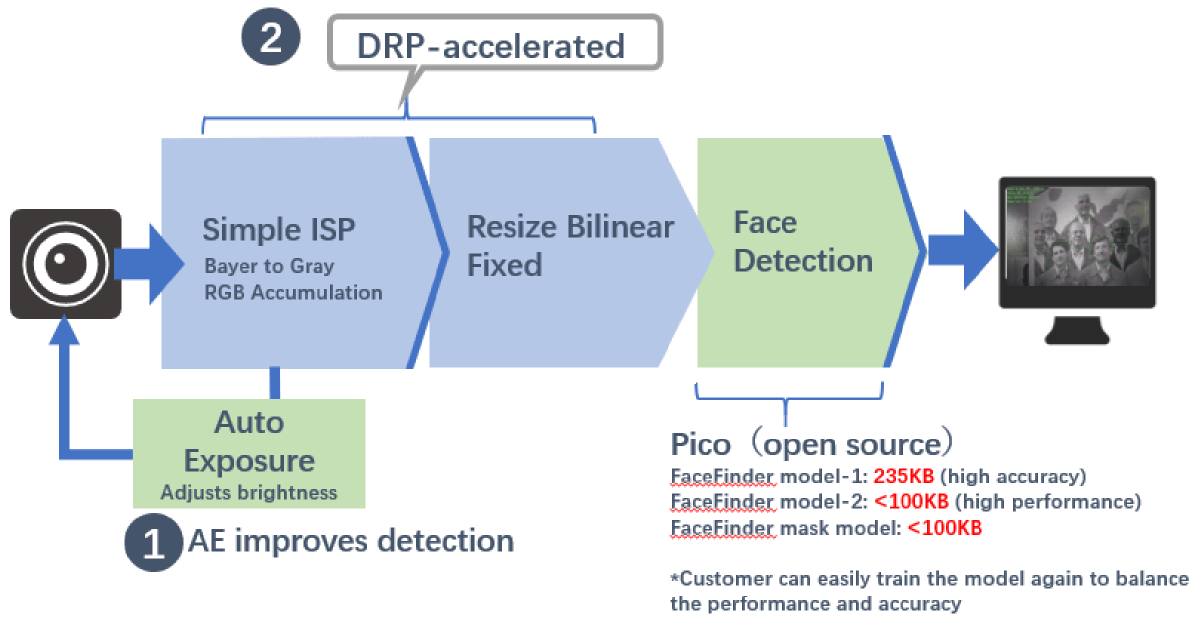
4.1. The overview of face and mask detection

The processing of the portion show in blue is realized by DRP hardware acceleration, and the Simple ISP library converts the Bayer format data of the CMOS sensor into grayscale data, and counts the average brightness of three configurable areas in a frame of image to adjust the automatic exposure parameters.

The second DRP library implements image scaling, compressing the 1280x720 resolution grayscale image into a 640x360 size image, which will greatly improve the face detection speed.

The green part in the picture is a light mask and face detection model run by the Cortex A9 processor, which is used to calibrate whether the current frame has a human face and whether it has a mask.

In this example, we will not rely on external RAM and only use RZ/A2M's 4MB on-chip high-speed RAM.


2
DRP-accelerated
Simple ISP
Bayer to Gray
RGB Accumulation
Resize Bilinear Fixed
Face Detection
Auto Exposure
Adjusts brightness
1 AE improves detection
Pico (open source)
FaceFinder model-1: 235KB (high accuracy)
FaceFinder model-2: <100KB (high performance)
FaceFinder mask model: <100KB
*Customer can easily train the model again to balance the performance and accuracy

4.2. Pre-Processing step “simple_isp_2_tiles”

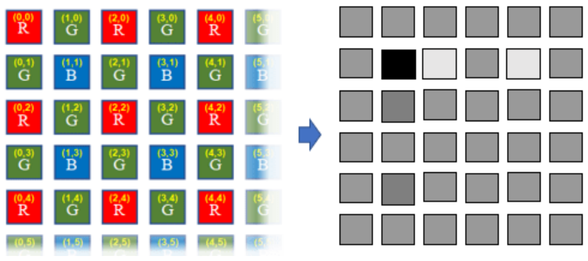
r_drp_simple_isp_2_tiles.dat

V1.10

86944 Bytes

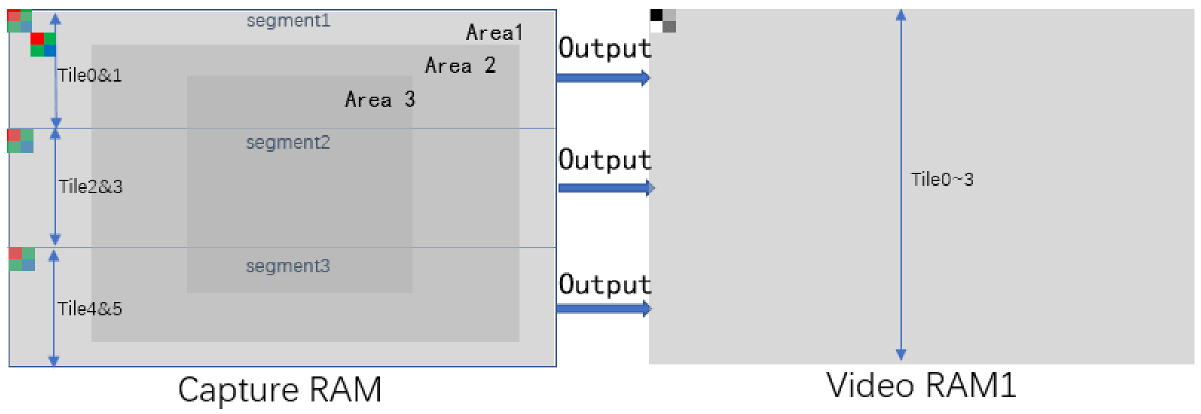
2 tile

Since face detection only needs to use grayscale images, we need to convert the Bayer format image of the CMOS sensor into a grayscale image. At this time, we load a simple_isp_2_tiles DRP library. This DRP library will have the following characteristics:



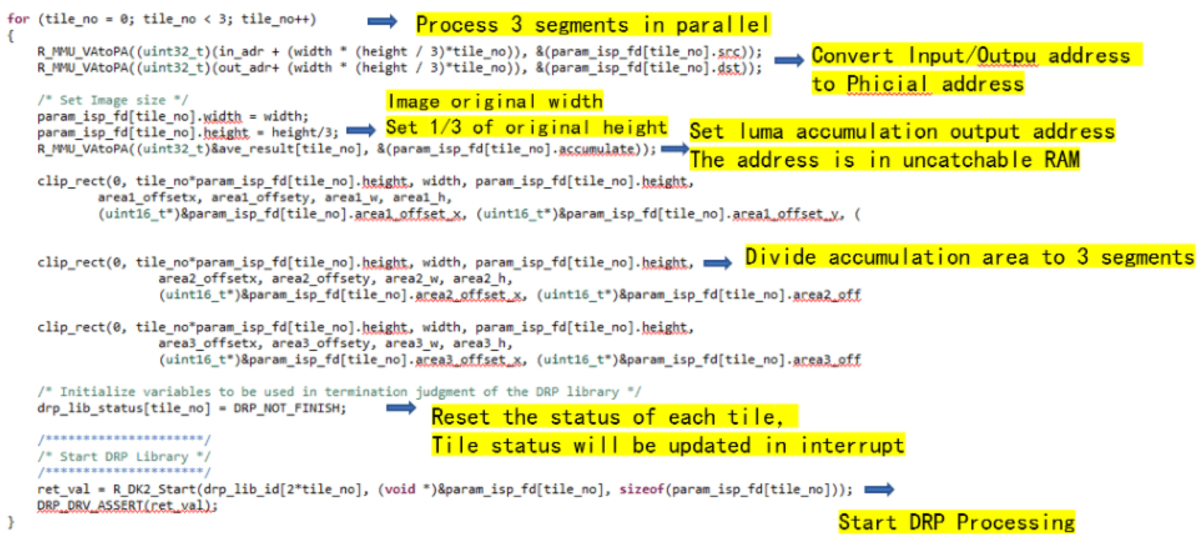
- Occupies 2 tiles of DRP hardware resources
- Realize Bayer to grayscale
- Accumulate the brightness values of all pixels in 3 independent areas
- Support multi-segment parallel processing

This library features multi-segment parallelized processing, we can load it into 3 sets of DRP tiles. Among them, the simple_isp_2_tiles library of Tile 0 and 1 handles the top 1/3 of the image, the simple_isp_2_tiles library of Tiles 2 and 3 handles the middle 1/3 of the image, and the simple_isp_2_tiles library of Tile 4 and 5 handles the bottom 1/3 of the image. These three parts of images are processed in parallel at the same time, which increases the processing speed by 3 times.



Because the DRP library provides a very convenient API interface, the above functions can be realized by simple programming operations.

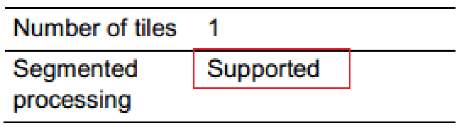
Sample Code



```
for (tile_no = 0; tile_no < 3; tile_no++)
{
    R_MMU_VAtoPA((uint32_t)(in_adr + (width * (height / 3)*tile_no)), &(param_isp_fd[tile_no].src));
    R_MMU_VAtoPA((uint32_t)(out_adr+ (width * (height / 3)*tile_no)), &(param_isp_fd[tile_no].dst));

    /* Set Image size */
    param_isp_fd[tile_no].width = width;
    param_isp_fd[tile_no].height = height/3;
    R_MMU_VAtoPA((uint32_t)&ave_result[tile_no], &(param_isp_fd[tile_no].accumulate));

    clip_rect(0, tile_no*param_isp_fd[tile_no].height, width, param_isp_fd[tile_no].height,
            area1_offsetx, area1_offsety, area1_w, area1_h,
            (uint16_t*)&param_isp_fd[tile_no].area1_offset_x, (uint16_t*)&param_isp_fd[tile_no].area1_offset_y, (


    clip_rect(0, tile_no*param_isp_fd[tile_no].height, width, param_isp_fd[tile_no].height,
                area2_offsetx, area2_offsety, area2_w, area2_h,
                (uint16_t*)&param_isp_fd[tile_no].area2_offset_x, (uint16_t*)&param_isp_fd[tile_no].area2_off

    clip_rect(0, tile_no*param_isp_fd[tile_no].height, width, param_isp_fd[tile_no].height,
                area3_offsetx, area3_offsety, area3_w, area3_h,
                (uint16_t*)&param_isp_fd[tile_no].area3_offset_x, (uint16_t*)&param_isp_fd[tile_no].area3_off

    /* Initialize variables to be used in termination judgment of the DRP library */
    drp_lib_status[tile_no] = DRP_NOT_FINISH;

    /*********************/
    /* Start DRP Library */
    /*********************/
    ret_val = R_DK2_Start(drp_lib_id[2*tile_no], (void *)&param_isp_fd[tile_no], sizeof(param_isp_fd[tile_no]));
    DRP_DRV_ASSERT(ret_val);
}
```

Process 3 segments in parallel
Convert Input/Outpu address to Phicial address
Image original width
Set 1/3 of original height
Set luma accumulation output address The address is in uncatchable RAM
Divide accumulation area to 3 segments
Reset the status of each tile, Tile status will be updated in interrupt
Start DRP Processing

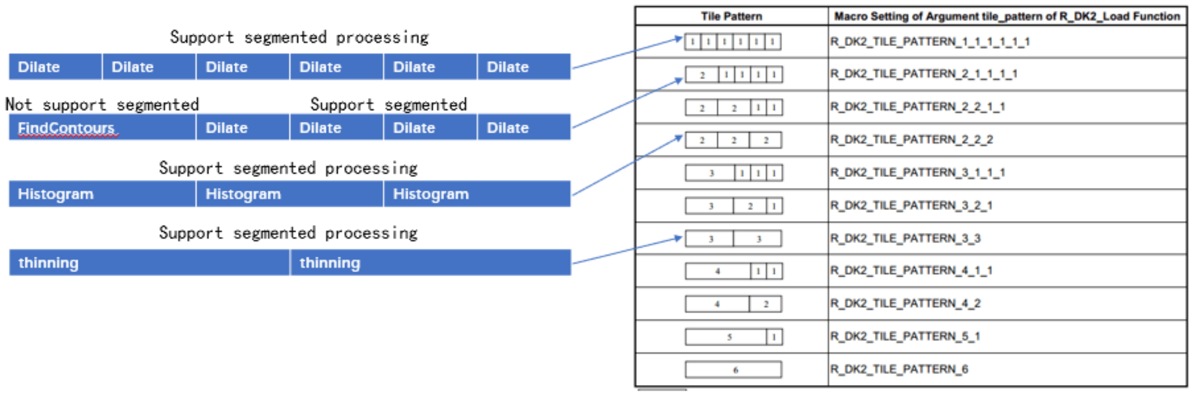
Users can decide how to load the DRP library according to the Number of tiles and Segmented Processing attributes in the application documentation of the DRP library



Number of tiles	1
Segmented processing	Supported

- Number of tiles: Indicates that the DRP library needs to occupy how many hardware tiles
- Segmented processing: indicates that the DRP task can be split into multiple tiles for parallel execution

Totally, there are 11 configurations to organize the DRP library in the Tile. You can choose which loading method to use according to the Number of tiles and Segmented attributes of DRP library in flexible way. below are some examples:



Tile Pattern	Macro Setting of Argument tile_pattern of R_DK2_Load Function
1 | 1 | 1 | 1 | 1 | 1	R_DK2_TILE_PATTERN_1_1_1_1_1_1
2 | 1 | 1 | 1 | 1	R_DK2_TILE_PATTERN_2_1_1_1_1
2 | 2 | 1 | 1	R_DK2_TILE_PATTERN_2_2_1_1
2 | 2 | 2	R_DK2_TILE_PATTERN_2_2_2
3 | 1 | 1 | 1	R_DK2_TILE_PATTERN_3_1_1_1
3 | 2 | 1	R_DK2_TILE_PATTERN_3_2_1
3 | 3	R_DK2_TILE_PATTERN_3_3
4 | 1 | 1	R_DK2_TILE_PATTERN_4_1_1
4 | 2	R_DK2_TILE_PATTERN_4_2
5 | 1	R_DK2_TILE_PATTERN_5_1
6	R_DK2_TILE_PATTERN_6

4.3. Pre-Processing step "resize bilinear fixed"

r_drp_resize_bilinear_fixed.dat

V1.00

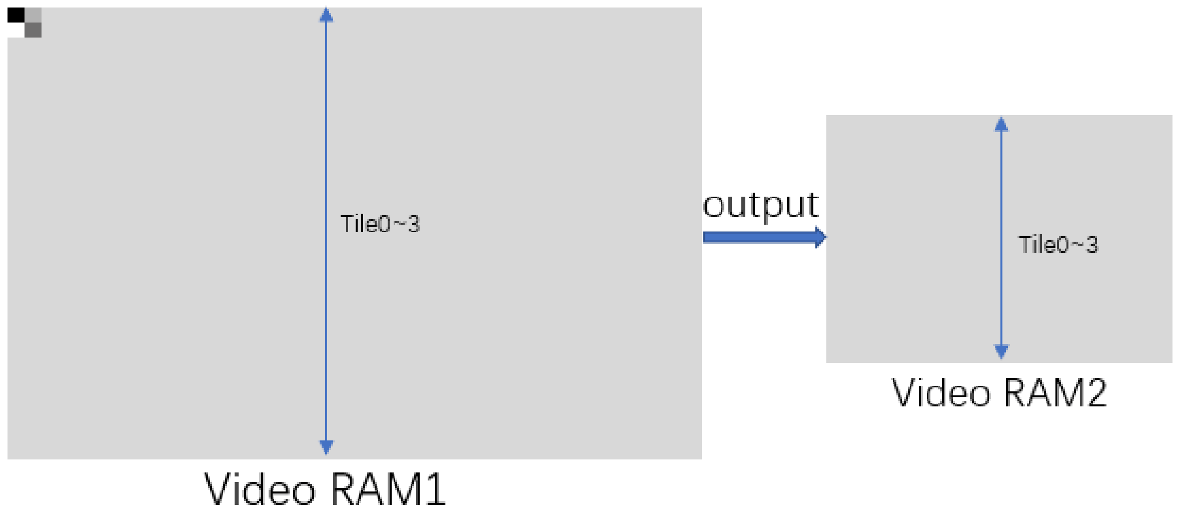
115744 Bytes

3 tile

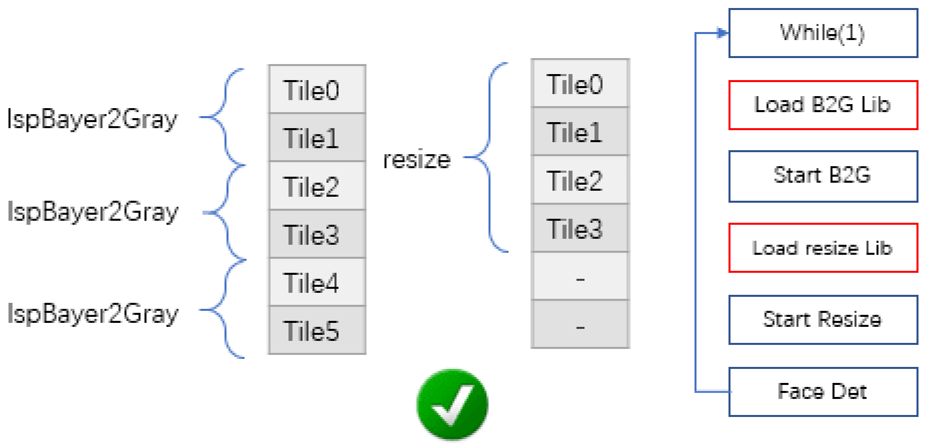
After getting a frame of grayscale image, we load a DRP library by resize bilinear_fixed to zoom this frame of image. Key features of the DRP library:

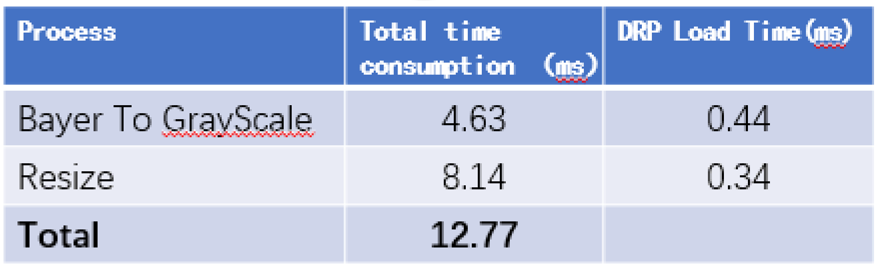
- Input 8bpp grayscale image
- Support ⅛ ¼ ½ 1x 2x 4x 8x 16x fixed zoom ratio
- Separate control of horizontal and vertical scaling
- Input width range 128~1280, input height range 8~960
- Occupies 4 tiles hardware resources without supporting segmented

After processing step 2, we read the grayscale image from Video RAM1, reduce the width and height to the original ½, and write the image to Video RAM2 for the next face detection.



The execution time of these two steps is about 4.6ms and 8.2ms. The parallel processing and the loading speed of the DRP library of less than 1ms have greatly optimized the execution speed of image preprocessing before face recognition.



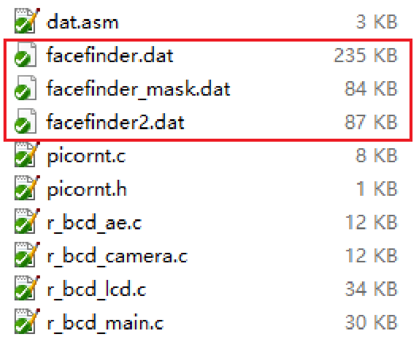


Process	Total time consumption (ms)	DRP Load Time (ms)
Bayer To GrayScale	4.63	0.44
Resize	8.14	0.34
Total	**12.77**	

5. Face Detection

We have integrated a pre-trained AI model in the sample project.

- facefinder.dat
- facefinder_mask.dat
- facefinder2.dat



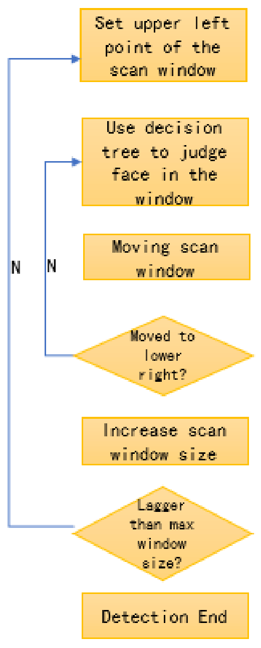
dat.asm	3 KB
facefinder.dat	235 KB
facefinder_mask.dat	84 KB
facefinder2.dat	87 KB
picornt.c	8 KB
picornt.h	1 KB
r_bcd_ae.c	12 KB
r_bcd_camera.c	12 KB
r_bcd_lcd.c	34 KB
r_bcd_main.c	30 KB

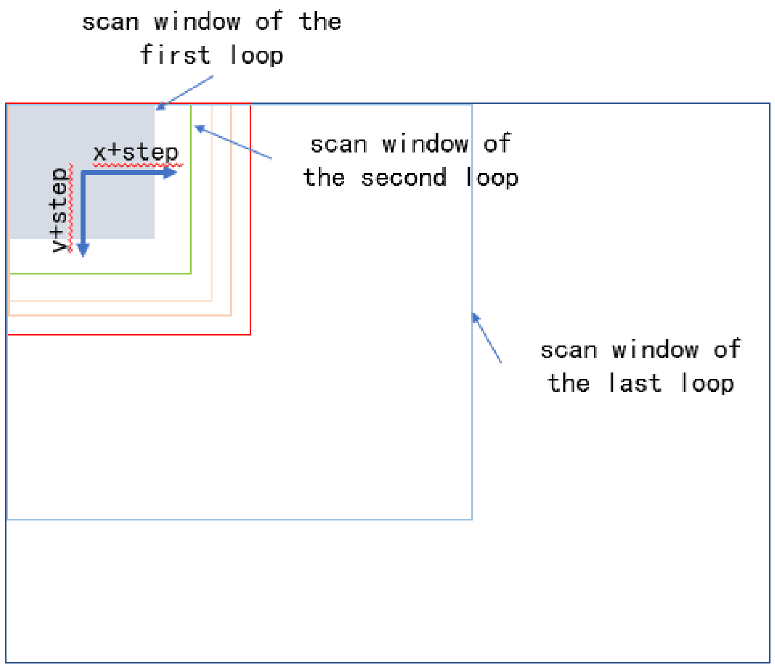
facefinder has higher accuracy than facefinder2, but it takes longer time, you can select either one according to your applications.

The AI model is loaded to RAM area during system boot, you can update the dat.asm file to change your own AI model.

```
.balign 32
.global g_cascade
g_cascade:
@.incbin "./facefinder.dat"
@/*facefinder2.dat is faster but may miss detection in some caess */
.incbin "./facefinder2.dat"

.balign 32
.global g_cascade_mask
g_cascade_mask:
@/*facefinder_mask only can detecth the face is wearing a mask */
.incbin "./facefinder_mask.dat"

.end
```

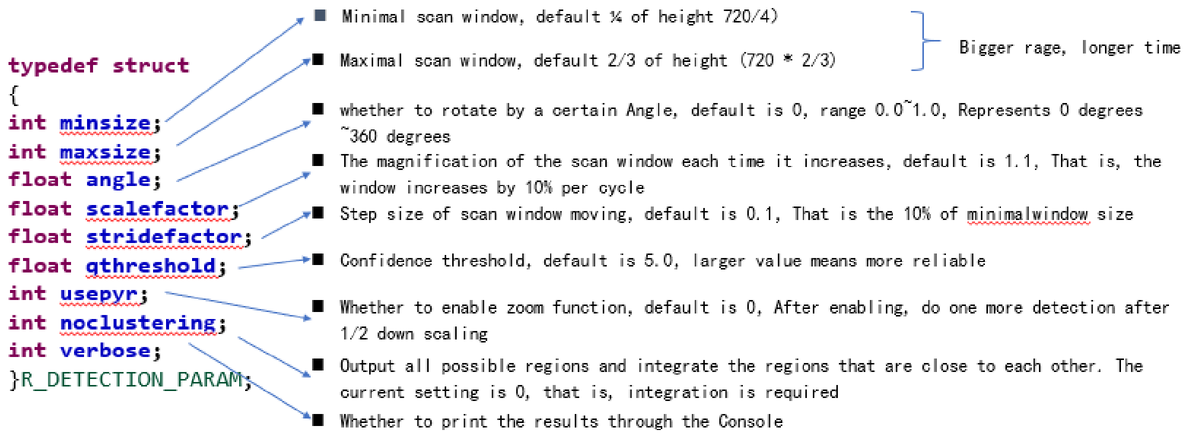
In the face detection process, we use a sliding window to scan the image generated in section 4.3 step by step, first use the smallest sliding window, and then gradually increase the size of the window. Use the decision tree model to detect whether there is a face in each sliding window.


Set upper left point of the scan window
Use decision tree to judge face in the window
Moving scan window
Moved to lower right?
Increase scan window size
Lagger than max window size?
Detection End
N
N


scan window of the first loop
x+step
y+step
scan window of the second loop
scan window of the last loop

6. Parameter Tunning

The following is the configuration of some key parameters, we can balance between detection accuracy and performance through parameter adjustment.



```
typedef struct
{
int minsize;
int maxsize;
float angle;
float scalefactor;
float stridefactor;
float qthreshold;
int usepyr;
int noclustering;
int verbose;
}R_DETECTION_PARAM;
```

- Minimal scan window, default ¼ of height 720/4)
- Maximal scan window, default 2/3 of height (720 * 2/3)

Bigger rage, longer time

- whether to rotate by a certain Angle, default is 0, range 0.0~1.0, Represents 0 degrees ~360 degrees
- The magnification of the scan window each time it increases, default is 1.1, That is, the window increases by 10% per cycle
- Step size of scan window moving, default is 0.1, That is the 10% of minimalwindow size
- Confidence threshold, default is 5.0, larger value means more reliable
- Whether to enable zoom function, default is 0, After enabling, do one more detection after 1/2 down scaling
- Output all possible regions and integrate the regions that are close to each other. The current setting is 0, that is, integration is required
- Whether to print the results through the Console

The following is a 1280x720 resolution input, which detects 1 face, 3 faces and 7 faces in the screen respectively. In the mask face mode, the detection speed can be above 15fps.

Scan Window Size: 180x180 ~ 480x480
Face number: 1
Model: facefinder2.dat
facefinder_mask.dat

Items	Performance
Bayer2Gray	4.63 ms
Resize	8.14 ms
Detect Mode (face + mask)	26~44 ms
Time/Frame Rate	39~58 ms / 17~24 fps
Detect Mode (face)	14~22 ms
Time/Frame Rate	27~34 ms / 29~35 fps
Detect Mode (mask)	14~24 ms
Time/Frame Rate	27~38 ms / 26~35 fps

Scan Window Size : 180x180 ~ 480x480
Face number : 3
Model : facefinder2.dat
facefinder_mask.dat

Items	Performance
Bayer2Gray	4.63 ms
Resize	8.14 ms
Detect Mode (face + mask)	42~60 ms
Time/Frame Rate	56~72 ms / 13~17 fps
Detect Mode (face)	21~30 ms
Time/Frame Rate	36~43 ms / 23~28 fps
Detect Mode (mask)	23~30 ms
Time/Frame Rate	36~43 ms / 22~26 fps

Scan Window Size : 180x180 ~ 480x480
Face number : 7
Model : facefinder2.dat
facefinder_mask.dat

Items	Performance
Bayer2Gray	4.63 ms
Resize	8.14 ms
Detect Mode (face + mask)	51 ms
Time/Frame Rate	64 ms / 15 fps
Detect Mode (face)	24 ms
Time/Frame Rate	33 ms / 26 fps
Detect Mode (mask)	35 ms
Time/Frame Rate	48 ms / 19 fps

7. AI Model Training

If you want to train your own AI model, you can run the script in Linux PC or Windows Bash environment. Ubuntu18.04 is used in below steps.

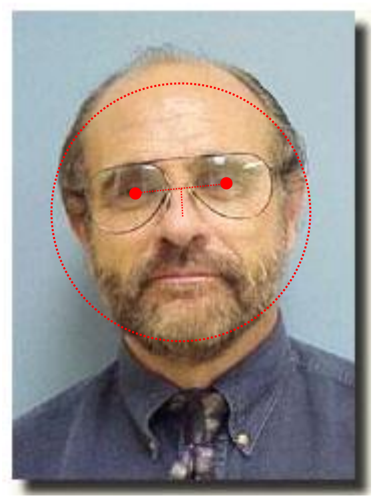
Before running the script, please make sure the python3 has been installed.

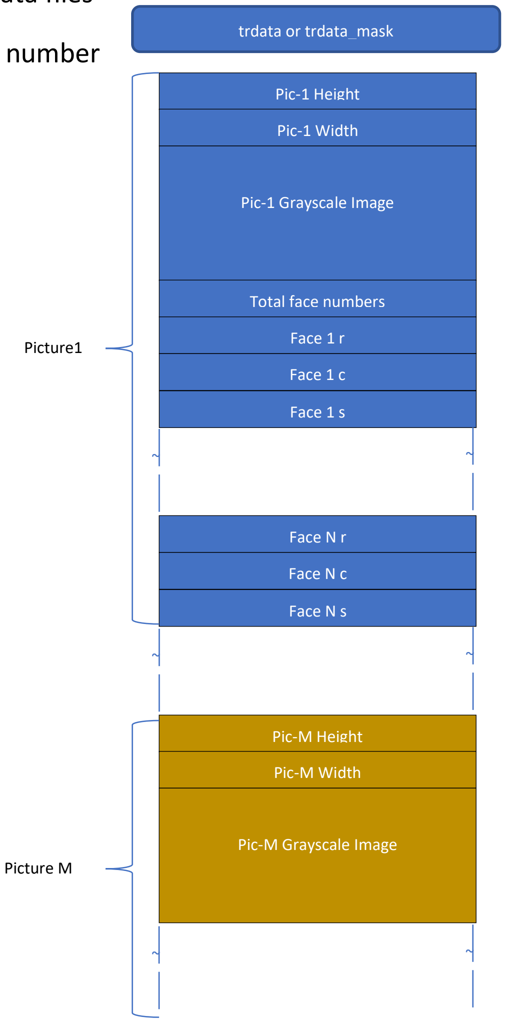
7.1 Training Data Generation

```
$cd ./pico-master/gen/sample
$ python3 ./caltechfaces.py caltechfaces > trdata
$ python3 ./caltechfaces.py caltechfaces_mask > trdata_mask
Parameter1: Picture folder name, includes face image and label file
Parameter2: Generated image data file which is used for the training
```

After the training data generating. You will get 2 image data files

- trdata -> file size is around 7.6GB, total marked face number is around 77K
- trdata_mask: file size is around 4.6GB





r: Circle center y

c: Circle center x

s: Circle center diameter

7.2 Training AI model

The basic training command is picolrn, there are 3 kinds of method to generate AI model according to different parameters.

Example 1:

```
$./picolrn <trdata> <cascade>
Parameter 1: The input image data file
Parameter 2: The output model file
# It will train the AI model with default parameters:
trp=0.98, fpr=0.4, ntree=16, ndepth=6, cascade_nums=5
e.g. : $./picolrn ./sample/trdata ./facefinder
# Use trdata as training file in sample folder, output model name is facefinder
$./picolrn ./sample/trdata_mask ./facefinder_mask > log.txt
# Use trdata_mask as traiing file in sample folder, output model name is facefinder_mask, save
the log in log.txt
```

Example 2:

```
$./picolrn <trdata> <output-cascade> <tpr> <fpr> <ntrees> <cascade_nums>
e.g. $./picolrn sample/trdata_mask ./facefinder_20191112 0.99f 0.45f 24 8
Parameter 1: The input image data file
Parameter 2: The output model file
Parameter 3:  tpr, the true-positive rate of each training stage
Parameter 4:  fpr, the false-positive rate of each training stage
Parameter 5:  ntree, how many decision trees will be added in each training stage
Parameter 6: How many stage will be trained
# Train the model with custom parameters.
```

Example 3:

```
$./picolrn <current-cascade> <trdata> <tpr> <fpr> <ntrees> <cascade_nums> <new-cascade>
e.g. $./picolrn ./facefinder_20191112 ./sample/trdata_mask  0.99f 0.45f 24 8 ./facefinder_new
Parameter 1: The original trainied AI model file
Parameter 2~7: same as example 2
# append N new stage to an existing trained AI model
```

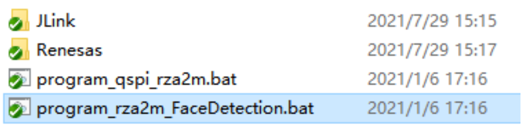
After the training, please copy the AI model file such as “facefinder_new” to your e2studio project:

src\renesas\application

Then update the dat.asm and compile the application again.

8. Pre-compiled binary Downloading

In case of using EVK, we provide a flash script to write bootloader and application binaries to QSPI flash. Please connect the JLINK debugger, then double click “program_rza2m_FaceDetection.bat” to download the images.



JLink	2021/7/29 15:15
Renesas	2021/7/29 15:17
program_qspi_rza2m.bat	2021/1/6 17:16
program_rza2m_FaceDetection.bat	2021/1/6 17:16

In case of using GR-MANGO, please connect microUSB cable between GR-MANGO and your PC, then copy the rza2m_face_detection_freertos_gcc_grmango.bin from bin folder to MBED disk.