IP Camera

HTTP CGI Specifications

for Intelligent Video Analytics features

V1.0.20160710

Version History

Ver	Revision	Date
V1.0	First version, should be used with camera's normal HTTP CGI	2016.07.10
V1.1	ADD CGI: To get the MJPEG of the detected face	2016.08.25
V1.1	ADD CGI: To get the standard Motion JPEG stream	2016.08.25
V1.2	Updated: Face Detection with Face ID and Confidence	2016.11.07
V1.3	Updated: the CGI command	2017.01.10

Contents

1. NOTICE

This document describes the specifications of the new CGI interface specialized in IP cameras' intelligent video analytics features. The interfaces are the part of the IP Camera CGI interfaces, so these interfaces should be used together with IP Camera's other normal CGI interfaces at the same time.
The document mainly describe related intelligent video analytics features CGI interfaces only.

2. COMMAND LIST

2.1 GET and SET Global Parameters for the feature

2.1.1 Parameters Description

The related parameters are defined like the below:

width: the width of the image processed
height: the height of the image processed.
fps: the frame rate of the video processed, the valid value is (5,10,15)
bsize: the minimum resolution value of the image processed, the valid value is (5,10,15)

Note:
1. both width and height value are read only parameters, it cannot be SET.
2. FPS value will impact the accuracy of the analytics result. The bigger of the value, the better accuracy, but it will bring the CPU load increasing, the performance will be impacted.
3. bsize value will implac the sensitivity of the analytics, the smaller of the value, the better sensitivity, but it will bring the CPU load increasing, the performance will be impacted.

2.1.2 GET Global Parameters for the feature

Request URL	`/action/get?subject=vaglobal`
Request Body	NONE

Response	`<?xml version="1.0" encoding="utf-8"?>`
	`<response>`
	`    <vaglobal>`
	`        <width>320</width>`
	`        <height>180</height>`
	`        <fps>10</fps>`

	`        <bsize>10</bsize>`
	`    </vaglobal>`
	`</response>`

2.1.3 SET Global Parameters for the feature

Request URL	`/action/set?subject=vaglobal`

Request Body	`<?xml version="1.0" encoding="utf-8"?>`
	`<request>`
	`    <vaglobal>`
	`        <width>320</width>`
	`        <height>180</height>`
	`        <fps>10</fps>`
	`        <bsize>10</bsize>`
	`    </vaglobal>`
	`</request>`

Response	`200: Succeed in Set`
	`400: Error of Request`
	`403: No Right to Set`
	`500: Failed to Set`

2.2 GET and SET People Counting Line Parameters

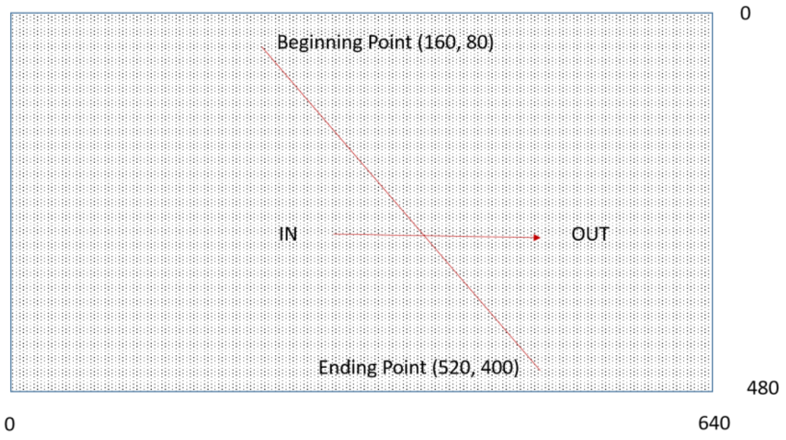
2.2.1 Parameters Description

The related parameters are defined like the below:

enable: enable or disable the people counting, 0: disable, 1: enable
line.begin: the beginning coordinate of the boundary of the zone, format: (x, y);
line.end: the ending coordinate of the boundary of the zone, format: (x, y)

Note:

1. the coordinate value is the relative value which is percentage value of the zone size which need people counting.
   For example, if the image of the zone resolution is 640 * 480, and the beginning coordinate was wanted to be set from (320, 240), thus the line.begin value should be (50, 50).
2. Below is the value of the line.begin and line.end calculated example.



On the above pictures shown, the line.begin value is (25,17), line.end value is (81,83)

2.2.2 GET Line Parameters of People Counting

Request URL	`/action/get?subject=counter`
Request Body	NONE

Response	`<?xml version="1.0" encoding="utf-8"?>`
	`<response>`
	`    <counter>`
	`        <enable>1</enable>`
	`        <line>`
	`           <begin>25,17</begin>`
	`           <end>81,83</end>`
	`        </line>`
	`    </counter>`
	`</response>`

2.2.3 SET Line Parameters of People Counting

Request URL	`/action/set?subject=counter`

Request Body	`<?xml version="1.0" encoding="utf-8"?>`
	`<request>`
	`    <counter>`
	`        <enable>1</enable>`
	`        <line>`
	`           <begin>25,17</begin>`
	`           <end>85,83</end>`
	`        </line>`
	`    </counter>`
	`</request>`

Response	`200: Succeed in Set`
	`400: Error of Request`
	`403: No Right to Set`
	`500: Failed to Set`

2.3 GET and SET Line Crossing Parameters

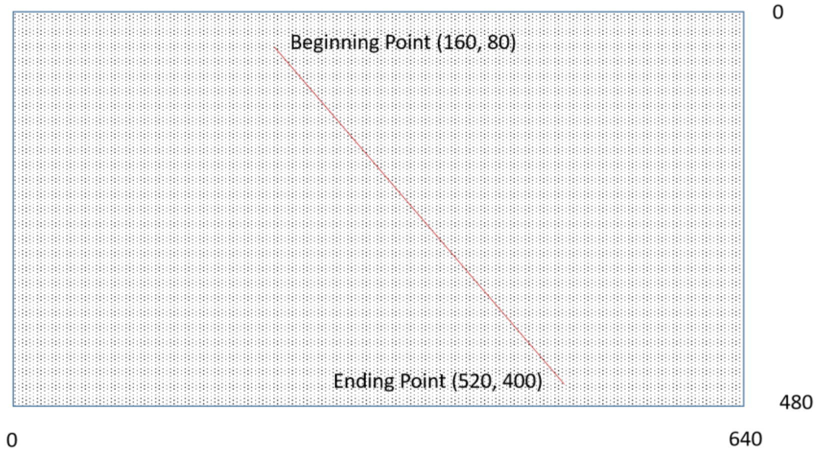
2.3.1 Parameters Description

The related parameters are defined like the below:

enable: enable or disable the line crossing detection, 0: disable, 1: enable
line.begin: the beginning coordinate of the detected line, format: (x, y);
line.end: the ending coordinate of the detected line, format: (x, y)

Note:

1. The coordinate value is the relative value which is percentage value of the crossing line. For example, if the image of the zone resolution is 640 * 480, and the beginning coordinate was wanted to be set from (320, 240), thus the line.begin value should be (50, 50).
2. Below is the value of the line.begin and line.end calculated example.



On the above pictures shown, the line.begin value is (25,17), line.end value is (81,83)

2.3.2 GET Line Crossing Parameters

Request URL	`/action/get?subject=crossline`
Request Body	NONE

Response	`<?xml version="1.0" encoding="utf-8"?>`
	`<response>`

	<crossline> <enable>1</enable> <line> <begin>25,17</begin> <end>81,83</end> </line> </crossline> </response>

2.3.3 SET Line Crossing Parameters

Request URL	`/action/set?subject=vaglobal`
Request Body	<?xml version="1.0" encoding="utf-8"?> <response> <crossline> <enable>1</enable> <line> <begin>25,17</begin> <end>81,83</end> </line> </crossline> </response>
Response	200: Succeed in Set 400: Error of Request 403: No Right to Set 500: Failed to Set

2.4 GET and SET Intrusion Parameters

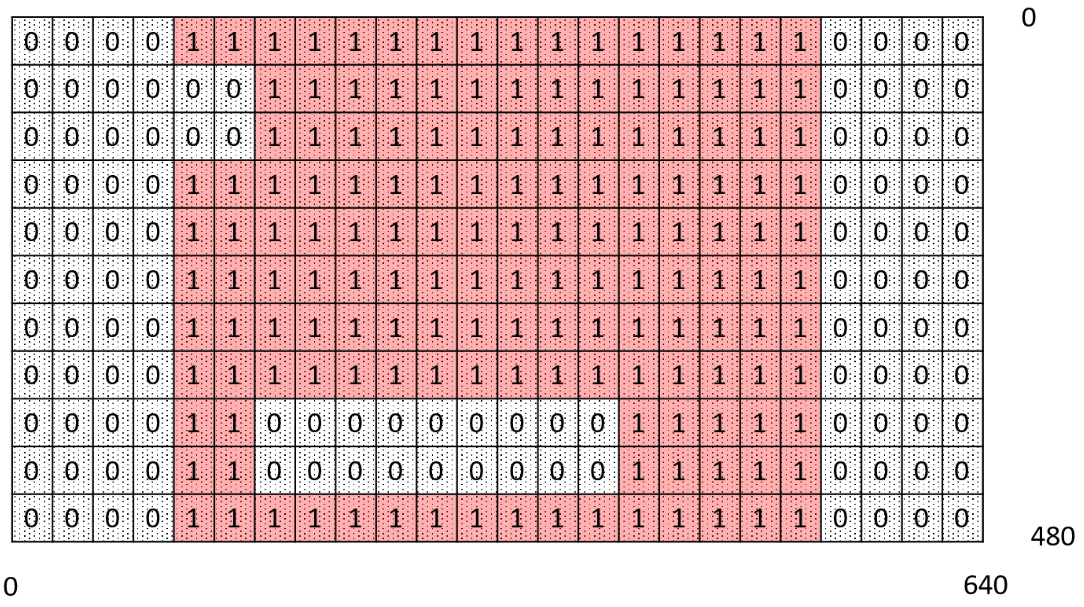
2.4.1 Parameters Description

The related parameters are defined like the below:
enable: enable or disable intrusion detection, 0: disable, 1: enable
areamask.row: the value of the raw of the macro zone in intrusion detection region.
areamask.col: the value of the column of the macro zone in intrusion detection region.
Areamask.mask: the mask value of the macro zone in the intrusion detection region.

NOTE:

1. Intrusion Region: the intrusion region is full size of the image which can be divided into several (row * col) detection macro zones, each macro zone can be set separately to be valid or invalid intrusion zone.

2. The value of area.mask identify if the macro zone is valid intrusion zone, 1: YES, 0: NO. Each bit represent each macro.
3. The format of areamask.mask is several groups which was built by every 4 bytes which is shown by hexadecimal strings. If less than 4 bytes, 0 will be alignment. Each group should be separated by ",".
4. Example like the below:



0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0
0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0
0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0
0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0
0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0
0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0
0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0
0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0
0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	1	1	1	1	1	0	0	0	0
0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	1	1	1	1	1	0	0	0	0
0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0

The red colored zone is the valid intrusion zones, the macro zone was identified: 1;
For the above image, the value will be like the below:
areamask.row : 11
areamask.col: 24
areamask.mask: 0ffff0,03fff0,03fff0,0ffff0,0ffff0,0ffff0,0ffff0,0ffff0,0c01f0,0c01f0,0ffff0

2.4.2 GET Intrusion Parameters

Request URL	`/action/get?subject=intrusion`
Request Body	NONE

Response	`<?xml version="1.0" encoding="utf-8"?>`
	`<response>`
	`    <intrusion>`
	`        <enable>1</enable>`
	`        <areamask>`
	`           <row>11</row>`
	`           <col>24</col>`
	`<mask>0ffff0,03fff0,03fff0,0ffff0,0ffff0,0ffff0,0ffff0,0ffff0,0c01f0,0`
	`c01f0,0ffff0</mask>`
	`        </areamask>`
	`    </intrusion>`

	</response>

2.4.3 SET Intrusion Parameters

Request URL	/action/set?subject=intrusion
Request Body	<?xml version="1.0" encoding="utf-8"?> <request> <intrusion> <enable>1</enable> <areamask> <row>11</row> <col>24</col> <mask>0ffff0,03fff0,03fff0,0ffff0,0ffff0,0ffff0,0ffff0,0ffff0,0c01f0,0c01f0,0ffff0</mask> </areamask> </intrusion> </request>
Response	200: Succeed in Set 400: Error of Request 403: No Right to Set 500: Failed to Set

2.5 GET and SET Human Detection and Face Detection

2.5.1 Parameters Description

The related parameters are defined like the below:

Person: enable or disable human/person detection, 0: disable, 1: enable
Face.active: enable or disable face detection, 0: disable, 1: enable
Face.confidence: the confidence value, value range: [0, 100]
Face.fsize: the size of the face, the value is (5,15,20);
Face.roi: the zone which can detect the face, the format is (left, top, right, bottom), the unit is percentage;

Note:

1. Human detection is now disabled.
2. Because of the limited performance of the camera, now the others features will be disabled when the face detection feature is enabled.
3. Starting face detection will take much higher CPU load;
4. The face size can be adjusted according to the distance of the camera with the face,

generally the more longer distance of the face, the smaller value of the face, but the CPU load will be higher.

5. To set the face detection zone (ROI) can lower the CPU load much. Generally the size of the zone is not bigger than 60% of the image. The bigger size of the zone, the much higher CPU load.

2.5.2 GET Human and Face Detection Parameters

Request URL	`/action/get?subject=human`
Request Body	NONE

Response	<?xml version="1.0" encoding="utf-8"?>
	<response>
	    <human>
	        <person>1</person>
	        <face>
	          <active>1</active>
	          <confidence>50</confidence>
	          <fsize>1</fsize>
	          <roi>10,10,50,30</roi>
	        </face>
	    </human>
	</response>

2.5.3 SET Human and Face Detection Parameters

Request URL	`/action/set?subject=human`

Request Body	<?xml version="1.0" encoding="utf-8"?>
	<request>
	    <human>
	        <person>1</person>
	        <face>
	          <active>1</active>
	          <confidence>50</confidence>
	          <fsize>1</fsize>
	          <roi>10,10,50,30</roi>
	        </face>
	    </human>
	</request>

Response	200: Succeed in Set 400: Error of Request 403: No Right to Set 500: Failed to Set

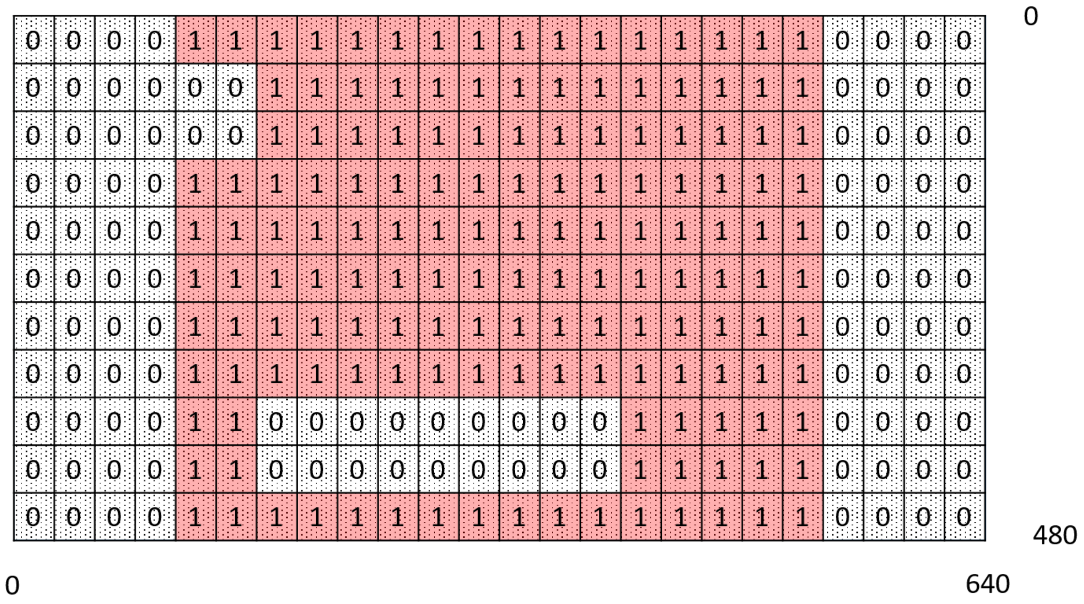
2.6 GET and SET Object Left and Removed Detection

2.6.1 Parameters Description

The related parameters are defined like the below:
left: enable or disable object left detection, 0: disable, 1: enable
removed: enable or disable object removed detection, 0: disable, 1: enable
areamask.row: the value of the raw of the macro zone in detection region.
areamask.col: the value of the column of the macro zone in detection detection region.
areamask.mask: the mask value of the macro zone in the detection region.

NOTE:

1. Detection Region: the detection region is full size of the image which can be divided into several (row * col) detection macro zones, each macro zone can be set separately to be valid or invalid detection zone.
2. The value of area.mask identify if the macro zone is valid detection zone, 1: YES, 0: NO. Each bit represent each macro.
3. The format of areamask.mask is several groups which was built by every 4 bytes which is shown by hexadecimal strings. If less than 4 bytes, 0 will be alignment. Each group should be separated by ",".
4. Example like the below:



0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0
0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0
0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0
0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0
0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0
0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0
0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0
0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0
0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	1	1	1	1	1	0	0	0	0
0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	1	1	1	1	1	0	0	0	0
0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0

The red colored zone is the valid detection zones, the macro zone was identified: 1;
For the above image, the value will be like the below:
areamask.row : 11
areamask.col: 24
areamask.mask: 0ffff0,03fff0,03fff0,0ffff0,0ffff0,0ffff0,0ffff0,0ffff0,0c01f0,0c01f0,0ffff0

2.6.2 GET Object Left and Removed Detection

Request URL	/action/get?subject=leftremoved
Request Body	NONE

Response	<?xml version="1.0" encoding="utf-8"?>
	<response>
	    <leftremoved>
	        <left>1</left>
	        <removed>1</removed>
	        <areamask>
	            <row>11</row>
	            <col>24</col>
	<mask>0ffff0,03fff0,03fff0,0ffff0,0ffff0,0ffff0,0ffff0,0ffff0,0c01f0,0c01f0,0ffff0</mask>
	         </areamask>
	    </leftremoved>
	</response>

2.6.3 SET Object Left and Removed Detection

Request URL	/action/set?subject=leftremoved

Request Body	<?xml version="1.0" encoding="utf-8"?>
	<request>
	    <leftremoved>
	        <left>1</left>
	        <removed>1</removed>
	<areamask>
	            <row>11</row>
	            <col>24</col>
	<mask>0ffff0,03fff0,03fff0,0ffff0,0ffff0,0ffff0,0ffff0,0ffff0,0c01f0,0c01f0,0ffff0</mask>
	</areamask>
	    </leftremoved>
	</request>

Response	200: Succeed in Set 400: Error of Request 403: No Right to Set 500: Failed to Set

2.7 GET and SET Loitering Detection

2.7.1 Parameters Description

The related parameters are defined like the below:

enable: enable or disable loitering detection, 0: disable, 1: enable

second: the minimum loitering time of the object stayed in the detection zone. Unit: second, the valid value: (5,10,15)

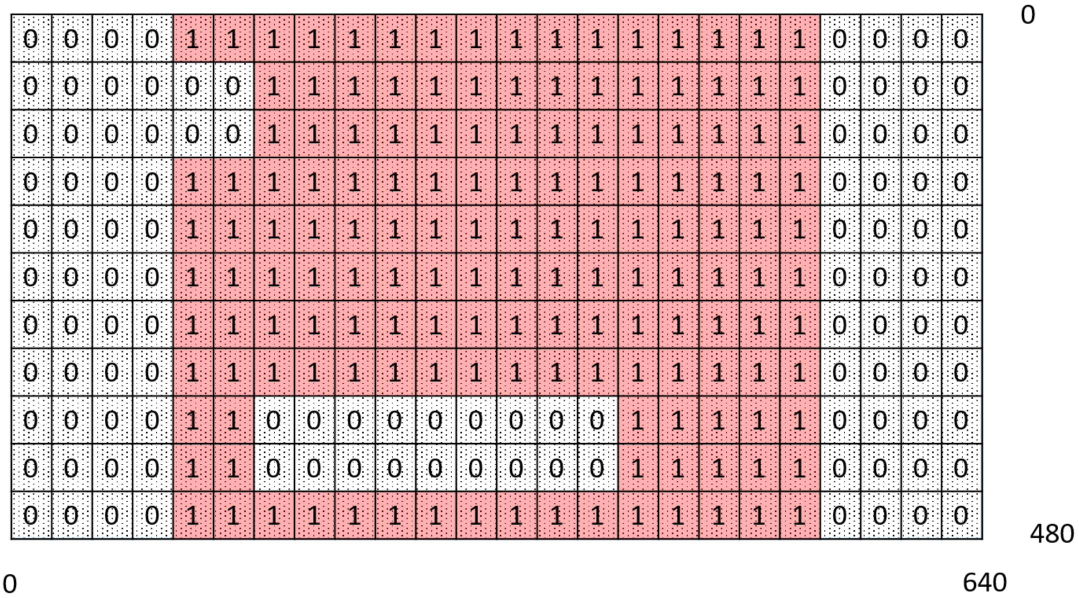
areamask.row: the value of the raw of the macro zone in detection region.

areamask.col: the value of the column of the macro zone in detection region.

areamask.mask: the mask value of the macro zone in the detection region.

NOTE:

1. Detection Region: the detection region is full size of the image which can be divided into several (row * col) detection macro zones, each macro zone can be set separately to be valid or invalid detection zone.
2. The value of area.mask identify if the macro zone is valid detection zone, 1: YES, 0: NO. Each bit represent each macro.
3. The format of areamask.mask is several groups which was built by every 4 bytes which is shown by hexadecimal strings. If less than 4 bytes, 0 will be alignment. Each group should be separated by ",".
4. Example like the below:



0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0
0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0
0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0
0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0
0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0
0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0
0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0
0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0
0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	1	1	1	1	1	0	0	0	0
0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	1	1	1	1	1	0	0	0	0
0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0

The red colored zone is the valid detection zones, the macro zone was identified: 1;

For the above image, the value will be like the below:

areamask.row : 11

areamask.col: 24

areamask.mask: 0ffff0,03fff0,03fff0,0ffff0,0ffff0,0ffff0,0ffff0,0ffff0,0c01f0,0c01f0,0ffff0

2.7.2 GET Loitering Detection

Request URL	/action/get?subject=loitering
Request Body	NONE
Response	<?xml version="1.0" encoding="utf-8"?> <response> <loitering> <enable>1</enable> <second>5</second> <areamask> <row>11</row> <col>24</col> <mask>0ffff0,03fff0,03fff0,0ffff0,0ffff0,0ffff0,0ffff0,0ffff0,0c01f0,0c01f0,0ffff0</mask> </areamask> </loitering> </response>

2.7.3 SET Loitering Detection

Request URL	/action/set?subject=loitering
Request Body	<?xml version="1.0" encoding="utf-8"?> <request> <loitering> <enable>1</enable> <second>5</second> <areamask> <row>11</row> <col>24</col> <mask>0ffff0,03fff0,03fff0,0ffff0,0ffff0,0ffff0,0ffff0,0ffff0,0c01f0,0c01f0,0ffff0</mask> </areamask> </loitering> </request>
Response	200: Succeed in Set 400: Error of Request 403: No Right to Set 500: Failed to Set

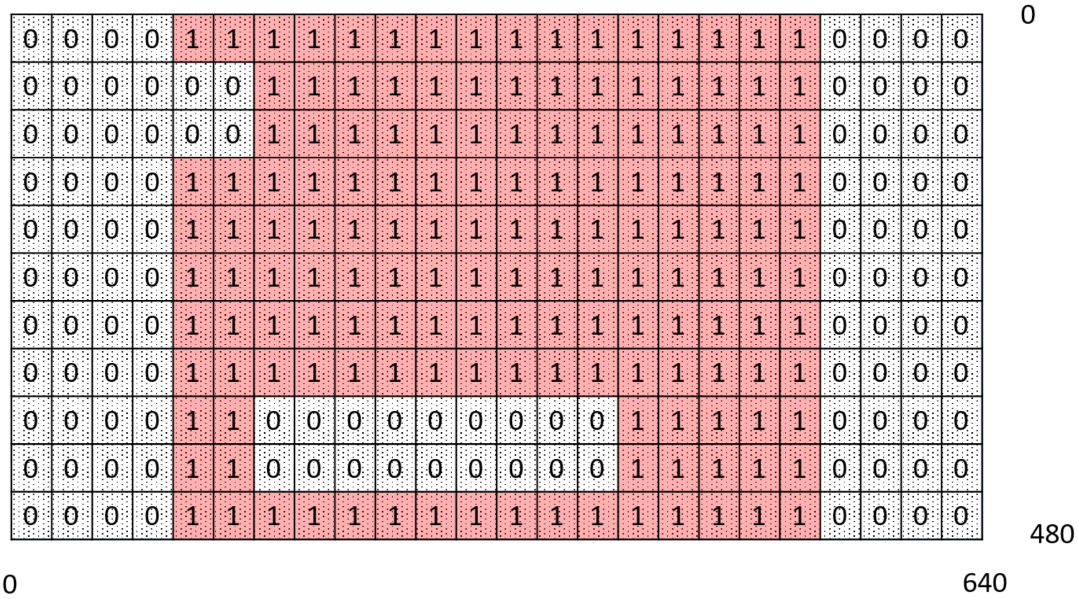
2.8 GET and SET Heatmap Detection

2.8.1 Parameters Description

The related parameters are defined like the below:
enable: enable or disable heatmap detection, 0: disable, 1: enable
second: the minimum time of the object stayed in the detection zone. Unit: second, the valid value: (5,10,15)
areamask.row: the value of the raw of the macro zone in detection region.
areamask.col: the value of the column of the macro zone in detection region.
areamask.mask: the mask value of the macro zone in the detection region.

NOTE:

1. Detection Region: the detection region is full size of the image which can be divided into several (row * col) detection macro zones, each macro zone can be set separately to be valid or invalid detection zone.
2. The value of area.mask identify if the macro zone is valid detection zone, 1: YES, 0: NO. Each bit represent each macro.
3. The format of areamask.mask is several groups which was built by every 4 bytes which is shown by hexadecimal strings. If less than 4 bytes, 0 will be alignment. Each group should be separated by ",".
4. Example like the below:



0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0
0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0
0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0
0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0
0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0
0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0
0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0
0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0
0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	1	1	1	1	1	0	0	0	0
0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	1	1	1	1	1	0	0	0	0
0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0

The red colored zone is the valid detection zones, the macro zone was identified: 1;
For the above image, the value will be like the below:
areamask.row : 11
areamask.col: 24
areamask.mask: 0ffff0,03fff0,03fff0,0ffff0,0ffff0,0ffff0,0ffff0,0ffff0,0c01f0,0c01f0,0ffff0

2.8.2 GET Heatmap Detection

Request URL	/action/get?subject=heatmap
Request Body	NONE

Response	<?xml version="1.0" encoding="utf-8"?>
	<response>
	    <heatmap>
	        <enable>1</enable>
	        <second>5</second>
	<areamask>
	          <row>11</row>
	          <col>24</col>
	<mask>0ffff0,03fff0,03fff0,0ffff0,0ffff0,0ffff0,0ffff0,0ffff0,0c01f0,0c01f0,0ffff0</mask>
	         </areamask>
	    </heatmap>
	</response>

2.8.3 SET Heatmap Detection

Request URL	/action/set?subject=heatmap

Request Body	<?xml version="1.0" encoding="utf-8"?>
	<request>
	    <heatmap>
	        <enable>1</enable>
	        <second>5</second>
	<areamask>
	          <row>11</row>
	          <col>24</col>
	<mask>0ffff0,03fff0,03fff0,0ffff0,0ffff0,0ffff0,0ffff0,0ffff0,0c01f0,0c01f0,0ffff0</mask>
	</areamask>
	    </heatmap>
	</request>

Response	200: Succeed in Set 400: Error of Request 403: No Right to Set 500: Failed to Set

2.9 GET and SET Wrong Direction Detection Parameters

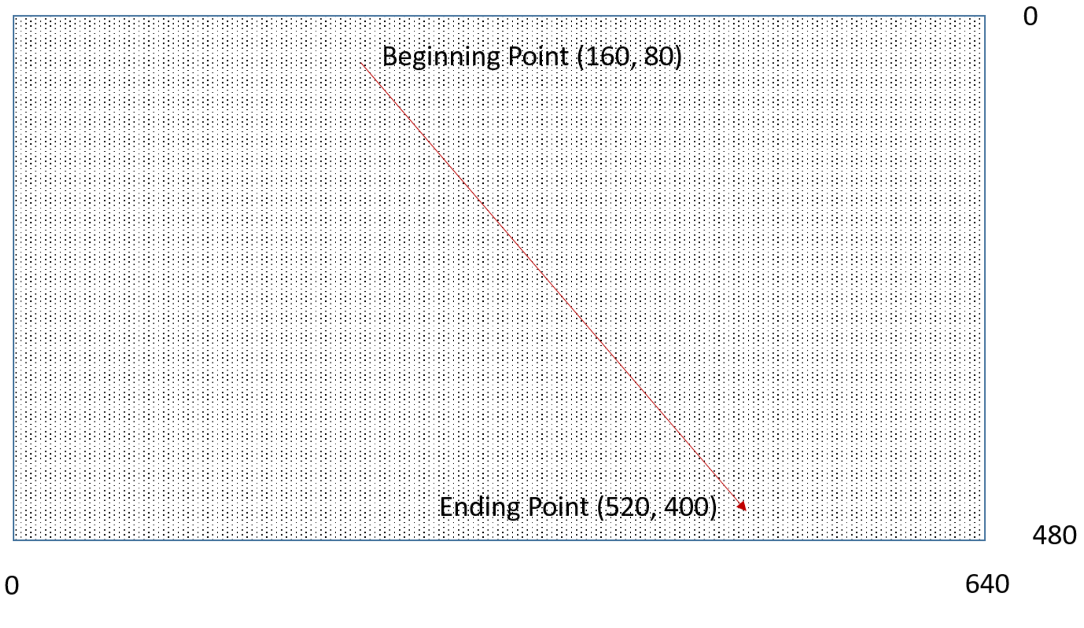
2.9.1 Parameters Description

The related parameters are defined like the below:

enable: enable or disable the wrong direction detection, 0: disable, 1: enable
line.begin: the beginning coordinate of the direction, format: (x, y);
line.end: the ending coordinate of the direction, format: (x, y)

Note:

1. The coordinate value is the relative value which is percentage value of the zone size which need people counting.
   For example, if the image of the zone resolution is 640 * 480, and the beginning coordinate was wanted to be set from (320, 240), thus the line.begin value should be (50, 50).
2. Below is the value of the line.begin and line.end calculated example.



On the above pictures shown, the line.begin value is (25,17), line.end value is (81,83)

2.9.2 GET the Wrong Direction Detection Parameters

Request URL	/action/get?subject=wrongdir
Request Body	NONE
Response	<?xml version="1.0" encoding="utf-8"?>

	<response>
	    <wrongdir>
	        <enable>1</enable>
	        <line>
	           <begin>25,17</begin>
	           <end>81,83</end>
	        </line>
	    </wrongdir>
	</response>

2.9.3 SET the Wrong Direction Detection Parameters

Request URL	/action/set?subject=wrongdir

Request Body	<?xml version="1.0" encoding="utf-8"?>
	<request>
	    <wrongdir>
	        <enable>1</enable>
	        <line>
	           <begin>25,17</begin>
	           <end>85,83</end>
	        </line>
	    </wrongdir>
	</request>

Response	200: Succeed in Set 400: Error of Request 403: No Right to Set 500: Failed to Set

2.10 The interface to Manage IVA

2.10.1 Parameters Description

The related parameters are defined like the below:

counter: RESET the counting result of the people counting, 0: NO,   1: YES
heatmap: RESET the statistics result of the heatmap counting, 0: NO,   1: YES

2.10.2 The interface to manage IVA

Request URL	/action/set?subject=vactrl

Request Body	NONE

Response	<?xml version="1.0" encoding="utf-8"?>
	<request>
	    <vactrl>
	        <reset>
	           <counter>1</counter>
	           <heatmap>1</heatmap>
	        </reset>
	    </vactrl>
	</request>

2.11 GET the Result of IVA

2.11.1 Parameters Description

The related parameters are defined like the below:

lseqno: the sequence number of the latest triggered event.
vaevent.id: ID number of the triggered event
vaevent.etype: the type of the triggered event: 1, Object Left; 2, Object Removed; 3, Crossing Line; 4, Intrusion; 5, Loitering; 6, Wrong Direction Way
vaevent.time: the triggered time of the event, 20160708T162004 means it's the time: Year 2016, Month: 07, Day: 08, Hour: 16, Minute: 20, Second: 04
counter.in: the quantity of the object IN;
counter.out: the quantity of the object OUT;
heatmap.count: the statistics result of the heatmap.

Note:
1. If there are many events which are happened at the same time, thus several "vaevent" will be returned, and all of the "lseqno" are valid.
2. If there is no any new event triggered, there will be no any "vaevent" and "lseqno" returned any more.
3. Upon getting the events, the "last" value should be set in the request URL. It means that the current "last" value is the value of the last returned and valid "lseqno". If it's the first time to get, the value of "last" can be set "-1";

2.11.2 GET the Result of IVA

Request URL	/action/get?subject=varesult&last=-1
Request Body	last: the sequence number of the current event got, if it's -1, it means

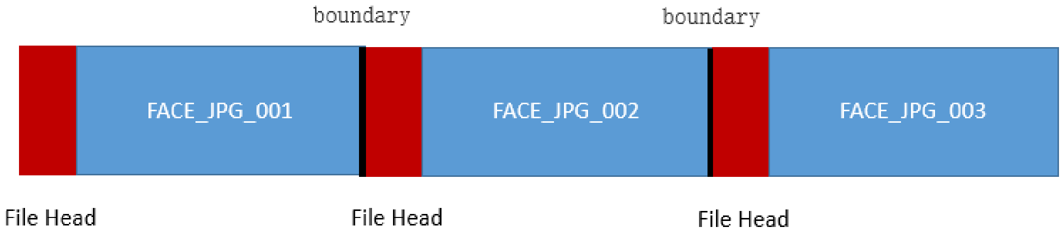
	that it’s to get all of the current event list.

Response	<?xml version="1.0" encoding="utf-8"?>
	<request>
	    <varesult>
	        <lseqno>1</lseqno>
	        <vaevent>
	           <id>1</id>
	           <etype>0</etype>
	           <time>20160204T120802</time>
	        </vaevent>
	        <vaevent>
	           <id></id>
	           <etype></etype>
	           <time>20160204T120802</time>
	        </vaevent>
	        ......
	        <counter>
	            <in>10</in>
	            <out>10</out>
	        </counter>
	        <heatmap>
	           <count>10</count>
	        </heatmap>
	    </varesult>
	</request>

2.12 GET the Snapshot of the detected Face

2.12.1 Parameters Description

The standard Motion JPEG Stream will be responded when ask the HTTP request. These images sending is using the standard MJPEG-Streamer technology, and there is the images stream sending from the cameras definition:



The boundary stream between the images: brovotechmjpegstreamboundary which is the string to show SOI (start of Image) or EOI (end of image)

File Head Definition:

File Type: image/jpeg

File-Length:

File-Name: face_20160826T134457_[46_50_82_86].jpg

20160826T134457: the image snapshotted time: YYYYMMDDThhmmss

[46_50_82_86]: the line and columns of the image in the full picture: lef_top_right_bottom

Here is the example response information if you ask the request:

HTTP/1.0 200 ok

Server: Brovotech/2.0.0

Connection: Keep-Alive

Content-Type:multipart/x-mixed-replace;boundary=------brovotectmjpegstreamboundary

--------------------------brovotectmjpegstreamboundary----------

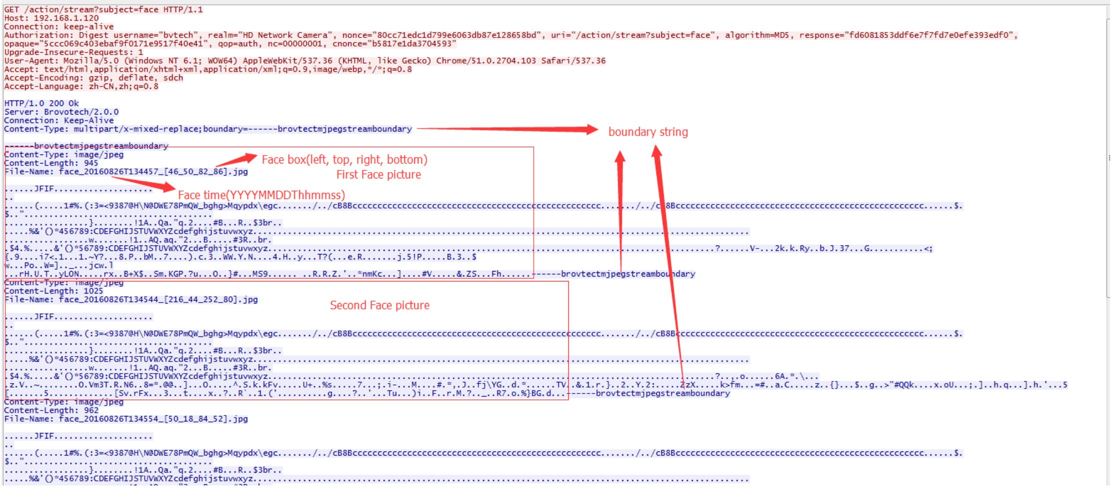
Content-Type: image/jpeg

Content-Length:945

File-Name: face_20160826T134457_[46_50_82_86].jpg

…………JFIF………………………………………………………..

The first face picture



```
GET /action/stream?subject=face HTTP/1.1
Host: 192.168.1.120
Connection: keep-alive
Authorization: Digest username="bvtech", realm="HD Network Camera", nonce="80cc71edc1d799e6063db87e128658bd", uri="/action/stream?subject=face", algorithm=MD5, response="fd6081853ddf6e7f7fd7e0efe393edf0",
opaque="5ccc069c403ebaf9f0171e9517f40e41", qop=auth, nc=00000001, cnonce="b5817e1da3704593"
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/51.0.2704.103 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Encoding: gzip, deflate, sdch
Accept-Language: zh-CN,zh;q=0.8

HTTP/1.0 200 Ok
Server: Brovotech/2.0.0
Connection: Keep-Alive
Content-Type: multipart/x-mixed-replace;boundary=------brovtectmjpegstreamboundary

------brovtectmjpegstreamboundary
Content-Type: image/jpeg
Content-Length: 945
File-Name: face_20160826T134457_[46_50_82_86].jpg

......JFIF...................
..
......(.....1#%.(:3=<9387@H\N@DWE78PmQW_bghg>Mqypdx\egc......./../cB8Bcccccccccccccccccccccccccccccccccccccccccccccccccc......./../cB8Bcccccccccccccccccccccccccccccccccccccccccccccccccc......$.
$.."....................................
.................}........!1A..Qa."q.2....#B...R..$3br..
.....%&'()*456789:CDEFGHIJSTUVWXYZcdefghijstuvwxyz................................................................................
.................w.......!1..AQ.aq."2...B.....#3R..br.
.$4.%.....&'()*56789:CDEFGHIJSTUVWXYZcdefghijstuvwxyz........................................................................?......V-...2k.k.Ry..b.J.37...G..........<;
{.9....i7<.1...1.~Y?...8.P..bM..7....).c.3..WW.Y.N....4.H..y...T?(...e.R.......j.5!P.....B.3..$
w...Po..W=].._...jcw.l
...rH.U.T..yLON.....rx..B+X$..Sm.KGP.?u...O..}#...MS9......  ..R.R.Z.'..*nmKc...]....#V.....&.ZS...Fh......------brovtectmjpegstreamboundary
Content-Type: image/jpeg
Content-Length: 1025
File-Name: face_20160826T134544_[216_44_252_80].jpg

......JFIF...................
..
......(.....1#%.(:3=<9387@H\N@DWE78PmQW_bghg>Mqypdx\egc......./../cB8Bcccccccccccccccccccccccccccccccccccccccccccccccccc......./../cB8Bcccccccccccccccccccccccccccccccccccccccccccccccccc......$.
$.."....................................
.................}........!1A..Qa."q.2....#B...R..$3br..
.....%&'()*456789:CDEFGHIJSTUVWXYZcdefghijstuvwxyz................................................................................
.................w.......!1..AQ.aq."2...B.....#3R..br.
.$4.%.....&'()*56789:CDEFGHIJSTUVWXYZcdefghijstuvwxyz........................................................................?..,.o......6A.*.\...
.z.V..~........O.Vm3T.R.N6..8=*.@@..]...O.....^.S.k.kFv.....U+..%s.....7...;.i-...M....#.*,.J..fj\YG..d.*......TV..&.1.r.}..2..Y.2:.....ZzX.....k>fm...=#..a.C.....z..{}...$..g..>"#QQk....x.oU...;.]..h.q...].h.'...5
[.......5............[Sv.rFx...3...t....x..?..R`..1.('..........g....?..'...Tu...)i..F..r.M.?.._..R7.o.%}BG.d...------brovtectmjpegstreamboundary
Content-Type: image/jpeg
Content-Length: 962
File-Name: face_20160826T134554_[50_18_84_52].jpg

......JFIF...................
..
......(.....1#%.(:3=<9387@H\N@DWE78PmQW_bghg>Mqypdx\egc......./../cB8Bcccccccccccccccccccccccccccccccccccccccccccccccccc......./../cB8Bcccccccccccccccccccccccccccccccccccccccccccccccccc......$.
$.."....................................
.................}........!1A..Qa."q.2....#B...R..$3br..
.....%&'()*456789:CDEFGHIJSTUVWXYZcdefghijstuvwxyz................................................................................
```

boundary string
Face box(left, top, right, bottom)
First Face picture
Face time(YYYYMMDDThhmmss)
Second Face picture

The user can use the HTTP CGI to get all JPEG images to the same directory.

2.12.2 GET Snapshot images stream of the detected Face

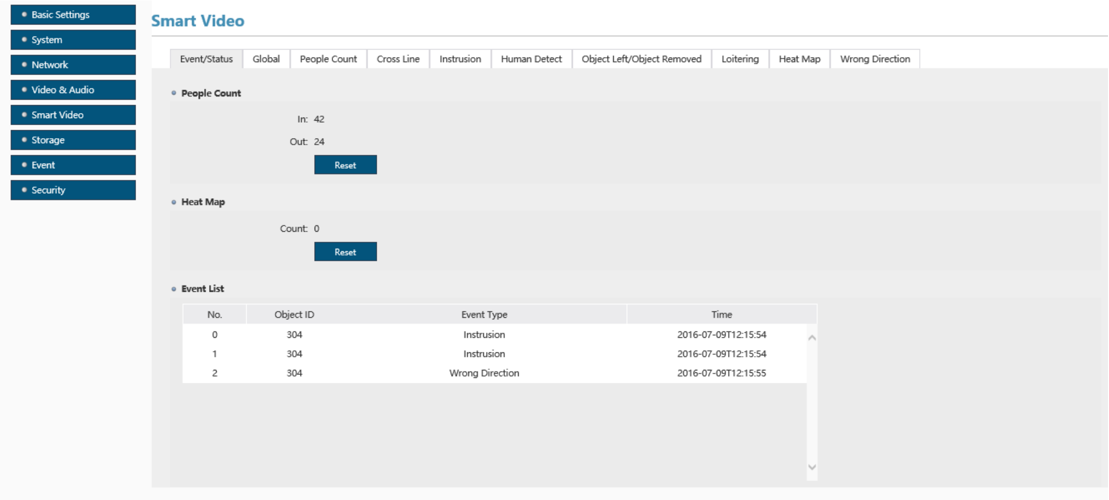
Request URL	/action/stream?subject=face
Request Body	None.
Response	200: Succeed in Request 400: Error of Request 403: No Right to Request

	`500: Error in Server`

3. WEB INTERFACES REFERENCE

3.1 WEB Interface Description

The camera support built-in web server for the clients to configure and acquire the IVA parameters and results. Below are the examples of the related interfaces.



People Count

In: 42

Out: 24

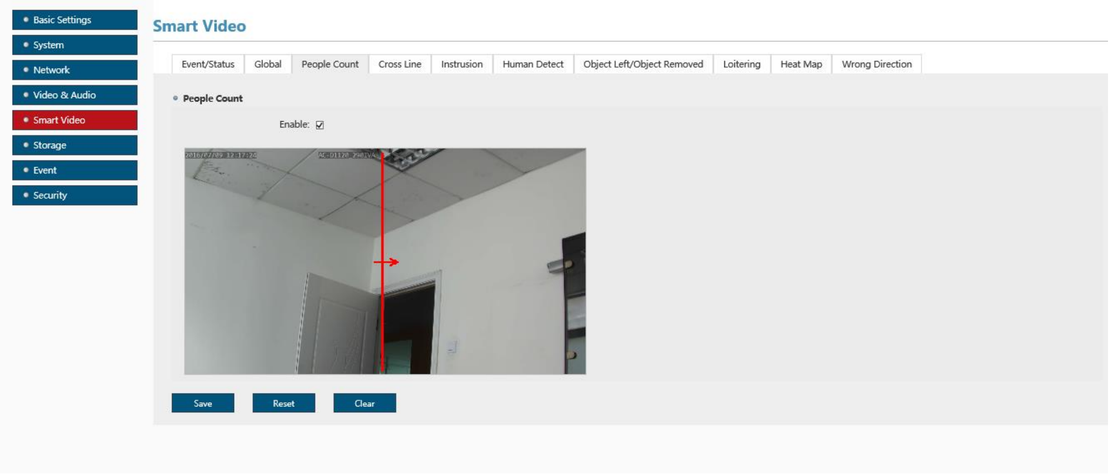
Heat Map

Count: 0

Event List

No.	Object ID	Event Type	Time
0	304	Instrusion	2016-07-09T12:15:54
1	304	Instrusion	2016-07-09T12:15:54
2	304	Wrong Direction	2016-07-09T12:15:55

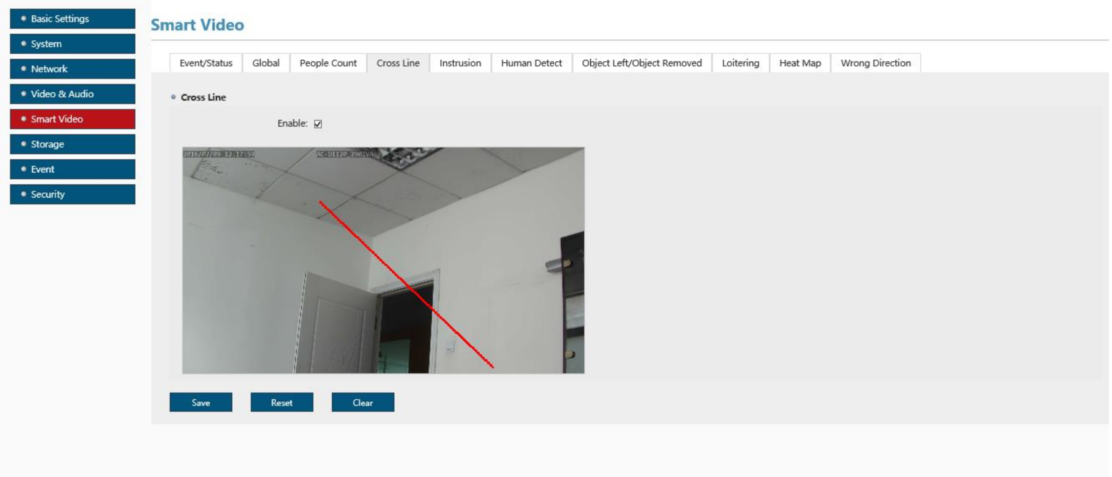
Fig.1: Smart Video Configure Window and IVA result



People Count

Enable: ☑

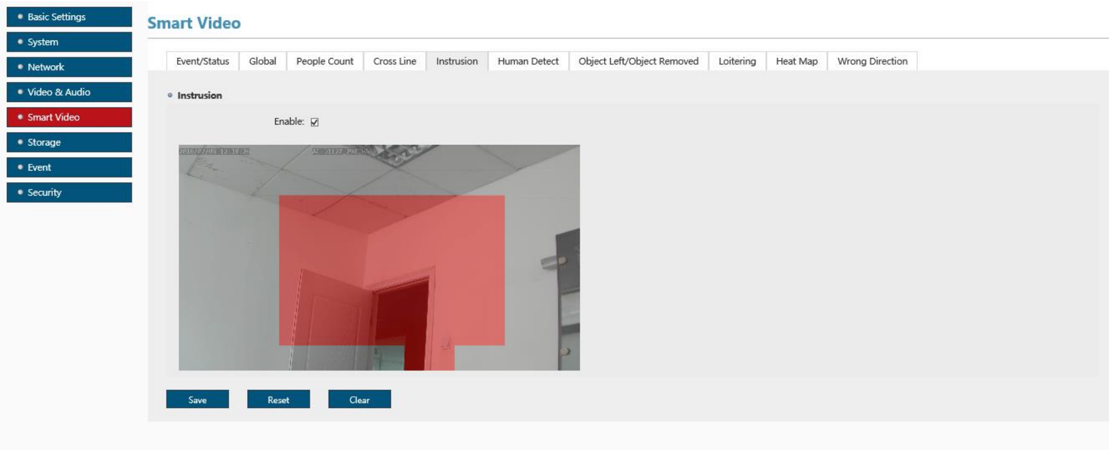
Fig.2: People Counting



Cross Line

Enable: ☑

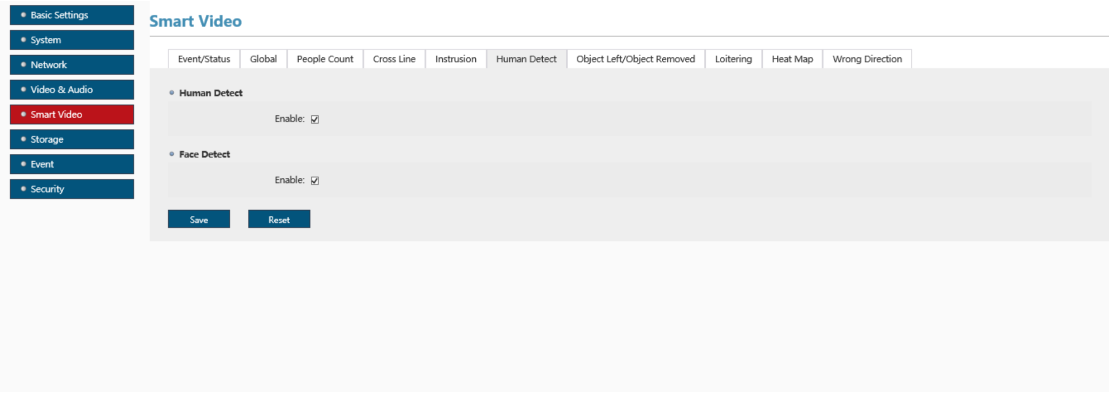
Fig.3: Line Crossing



Instrusion

Enable: ☑

Fig. 4: Intrusion Detection



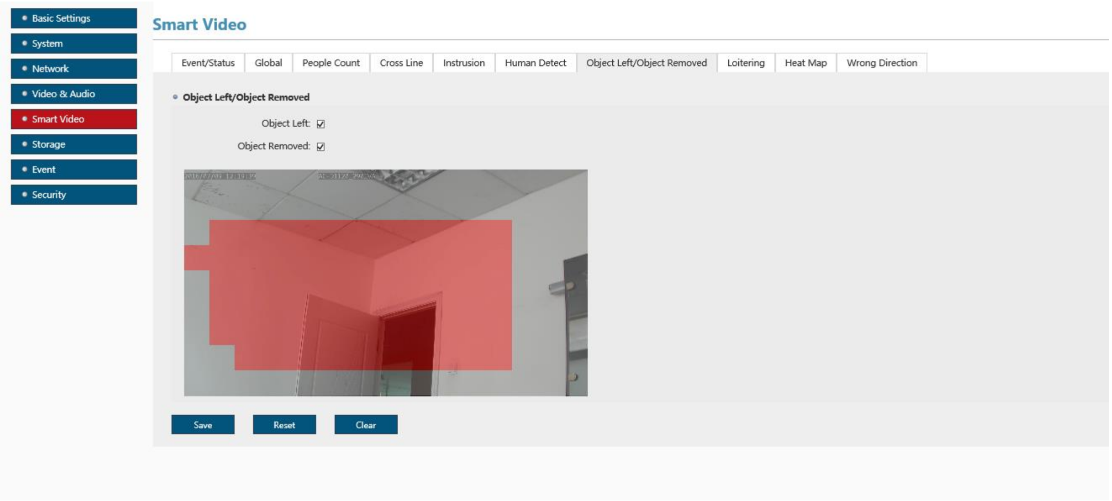
Human Detect

Enable: ☑

Face Detect

Enable: ☑

Fig. 5: Human Detection and Face Detection



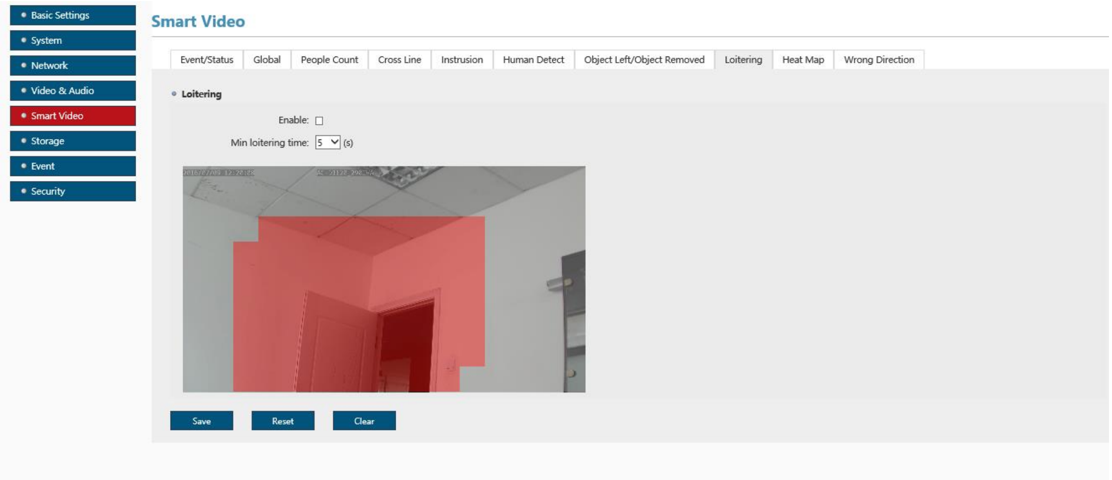
Smart Video

Object Left/Object Removed

Object Left: ☑

Object Removed: ☑

Fig. 6: Human and Face Detection



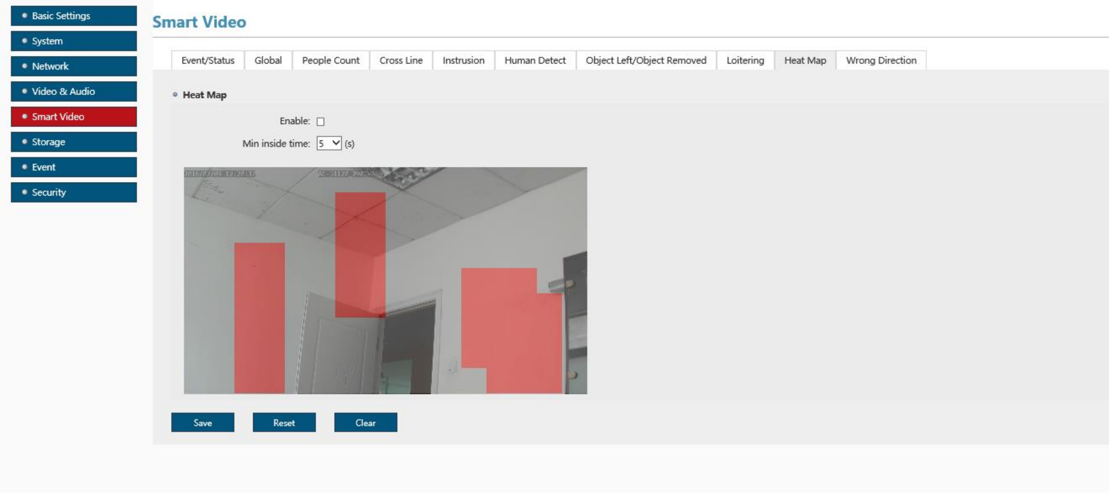
Smart Video

Loitering

Enable: □

Min loitering time: 5 (s)

Fig. 7: Loitering Detection



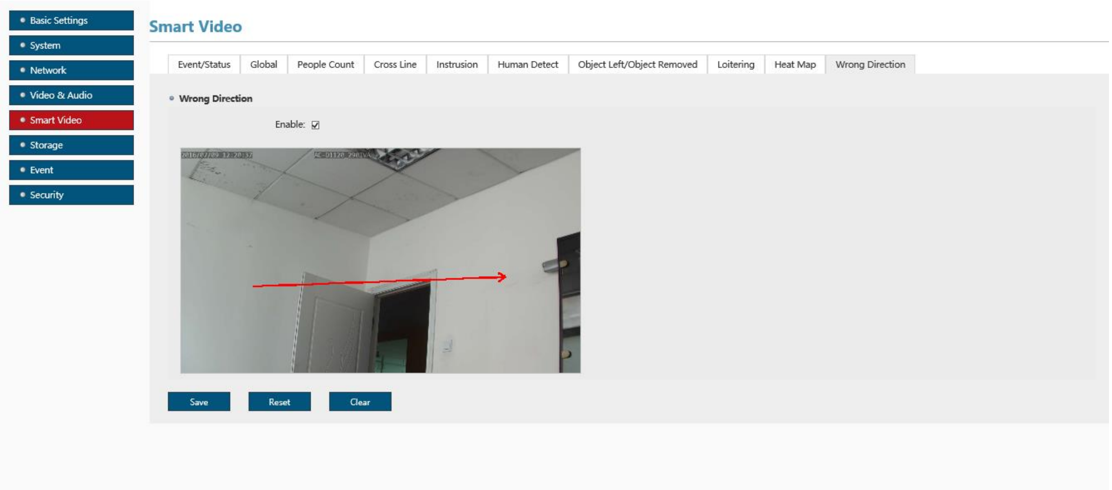
Smart Video

Heat Map

Enable: □

Min inside time: 5 (s)

Fig. 8: HeatMap Detection



Smart Video

Wrong Direction

Enable: ☑

Fig. 9: Wrong Direction Detection