Camera Trainer with FW 7.10 and CM 6.20
Technical Note

1: Train the camera

- Provide examples to teach camera.

(Car / no car)

2: Start application specific detection

- Camera detects parked cars.

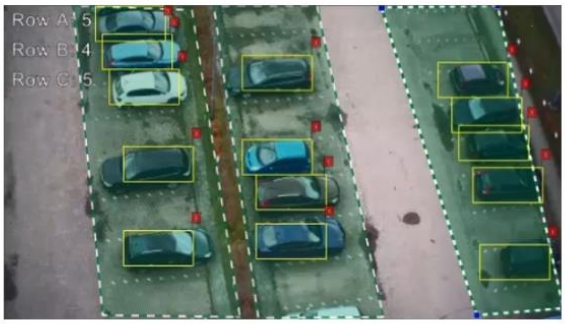


Table of contents

1 Introduction 3

1.1 Applications 3

1.2 Limitations 3

1.3 Supported cameras 4

2 Technical Background 5

2.1 Machine learning: Finding threshold between target object and world 5

2.2 Camera Trainer: Not the whole world but one single camera view 6

2.3 Camera Trainer: Based on Histogram of oriented Gaussians 6

2.4 Camera Trainer: Limitations on distance, pose and perspective 7

2.5 Will machine learning, especially deep learning, replace current video analytics? 8

3 Configuration 9

3.1 Activate license 9

3.2 Training a detector 9

3.3 Using recording for training 12

3.4 Configuring alarm or counting tasks 12

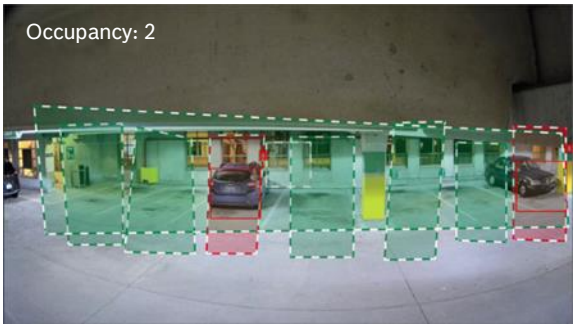
3.5 Alarm on trained object being absent 13

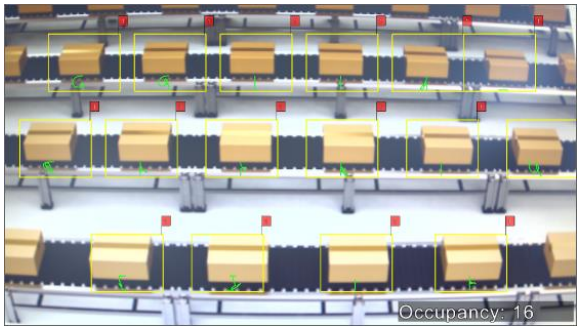
3.6 Backup & transfer to other cameras 14

3.7 Tips, Tricks & FAQs 14

1 Introduction

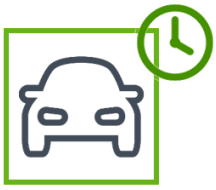
Based on examples of target objects and non-target objects, the Camera Trainer uses machine learning to allow the user to define objects of interest and generate detectors for them. In contrast to the moving objects detected in general by IVA 7.10, Camera Trainer allows detection of both moving and non-moving objects, separates and immediately classifies them. Training can be done both on live video as well as on recordings available through the respective camera. The resulting detectors can be down- and uploaded for distribution to other cameras.



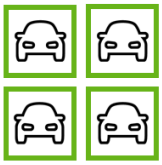


1.1 Applications

- Occupancy of parking spaces
- Presence of trucks at loading docks
- Vehicle counting
- Filter trains to only detect people on train tracks
- Box pile up in production lines
- Icicle detection
- Camera tamper detection via markers
- Object states: Open / closed / other position / ...
- ... and many more

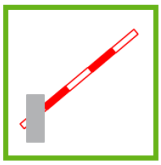


Find stationary objects



Separate objects clustered together

Detect user-defined objects



Recognize user-defined situations

1.2 Limitations

- Training is only possible via the Configuration Manager.
- Training samples are stored in the Configuration Manager. When needing to configure the Camera Trainer detectors from another computer, the samples need to be transferred by hand by saving the VCA configuration under VCA -> Main Operation on the computer with the samples, and loading the VCA configuration from the other computer. The Configuration Manager adds the samples to the VCA configuration and recovers them from there. (see Section 3.6)
- Target objects must have distinct edge patterns which are different to anything else in the scene (see Section 2.3)
- Objects that are too similar to each other in their edge patterns cannot be separated. This includes not being able to distinguish different car models, person identification, wearing helmets or safety vests (see Section 2.3)
- Target objects in the image must have a similar size in pixels, similar perspective from the camera to them, similar pose and similar background (see Section 2.4)
- The object color filter is not available nor used for Camera Trainer objects.
- No 3D evaluation including geolocation or speed for Camera Trainer objects.
- Not usable for intrusion detection
- Minimum FW version: 7.10
- Minimum Configuration Manager version: 6.20

1.3 Supported cameras

Camera Trainer can be enabled on all Bosch IP color cameras of range 7000-9000 that are based on the common product platforms 6, 7 and 7.3. This includes

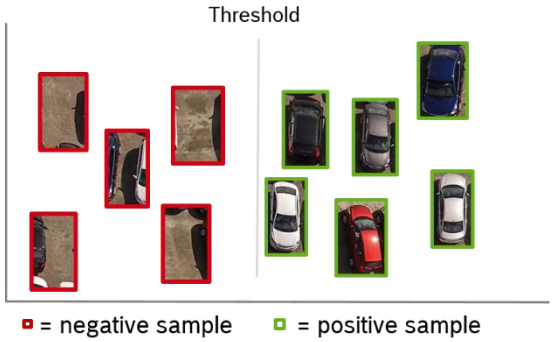
- DINION IP starlight 7000 HD
- DINION IP starlight 8000 MP
- DINION IP ultra 8000 MP
- FLEXIDOME IP panoramic 7000
- FLEXIDOME IP panoramic 7000 IC
- FLEXIDOME IP starlight 7000 HD
- FLEXIDOME IP starlight 8000i
- FLEXIDOME IP ultra 8000i
- MIC IP starlight 7000i
- MIC IP fusion 9000i (color channel only)

See the corresponding fact sheet for a full and updated list including CTNs.

2 Technical Background

2.1 Machine learning: Finding threshold between target object and world

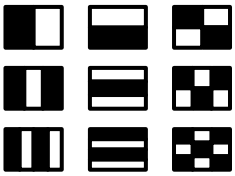
Machine learning for object detection is the process of a computer determining a good separation between positive (target) samples and negative (background) samples. In order to do that, a machine learning algorithm builds a model of the target object based on a variety of possible features, and of thresholds where these features do/don't describe the target object. This model building is also called the training phase or training process. Once the model is available, it's used to search for the target object in the images later on. This search in the image together with the model is called a detector.



Hand-crafted features typically describe edges and include:

- Haar features: Binary edge descriptors via black/white tiles
- Histogram of oriented Gaussians (HoG): Histogram of quantized edge direction and contrast

The resulting model will typically have around 2000 parameters. There are different methods of machine learning with hand-crafted features including support vector machines (SVNs), AdaBoost, and decision trees. Each of these methods has certain advantages, however, all of them result in similar performance levels. A detector based on these features can typically run in real-time in current network camera hardware.


Haar features

Histogram of oriented Gaussians (HOG)

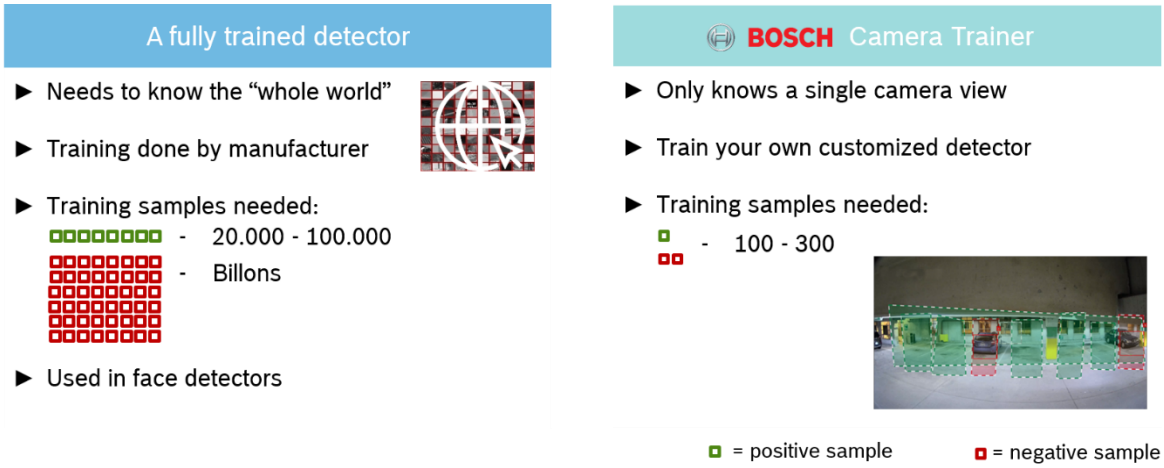
Neural networks are based on the visual cortex and are able to learn descriptive features on their own. They utilize a neural network structure of optimization parameters instead of using the handcrafted features. Typically, neural networks for image processing will also learn edge features and combine them first to parts of the object, and then the full target object itself. Deep neural networks for image processing use roughly ~20 million parameters, and can deliver a performance boost of up to 30%. On the other hand, deep neural networks are a brute force approach that requires hundreds of times more computational power than available in current network camera hardware. Using smaller networks with similar calculation power to hand crafted features is possible, but the performance boost will typically be much smaller.

Neural networks: Layers of neurons. Each neuron combines several inputs to a decision.

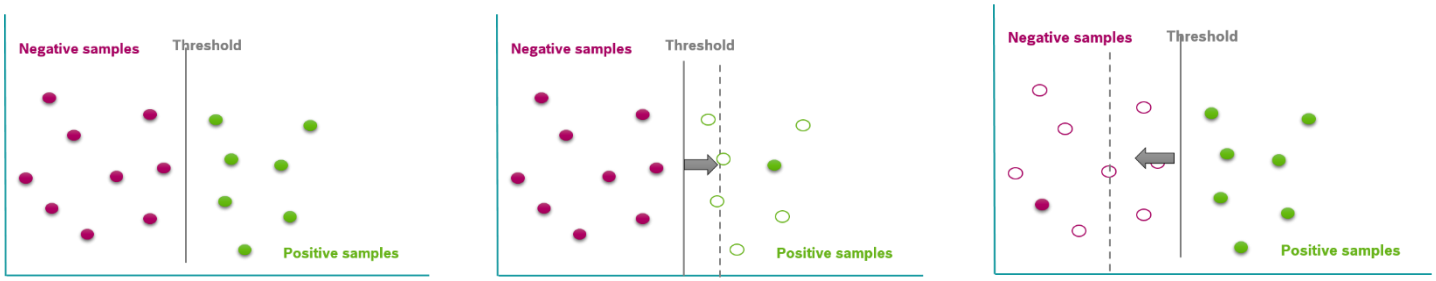
Besides the model and training method, the samples of target objects and the background are very important. For a task like face or person detection, the positive samples need to show all possible variations including perspective and pose, while the background samples have to represent the full world. Therefore, machine learning needs tens of thousands of examples of the target object and billions of examples of what the rest of the world looks like. This is a huge effort both in collecting and preparing the sample images. For automated machine learning, either the target objects have to be marked in the image or the images restricted to show only the target object. Furthermore, modelling the complexity of the full world is one of the reasons machine learning is computationally expensive.

2.2 Camera Trainer: Not the whole world but one single camera view

A single camera does not look at the whole world at once, but just at a single field of view. To describe both the target object and the background in a single field of view, far fewer samples are needed even when considering different seasons and weather conditions. Depending on the task, the number of samples required can be as few as some tens of samples, or as much as a few hundred samples in total. This opens up the possibility for everyone to train their own target object detector.



Both positive samples and negative samples are needed. Remember that the model includes the threshold where each feature describes the target object and where it does not. Let's look at a simple example here. In the left image, both positive and negative samples have been provided, and a perfect threshold has been found. In the middle image, only the solid green dot has been provided as positive sample. This will move the threshold to the right (dotted grey line), some positive samples will thus be misclassified as background, and missed by the resulting detector. In the image on the right, only the solid red dot has been supplied as negative or background example. This moves the threshold to the left (dotted grey line), some negative samples will be misclassified as being the target object, and false detections will occur.

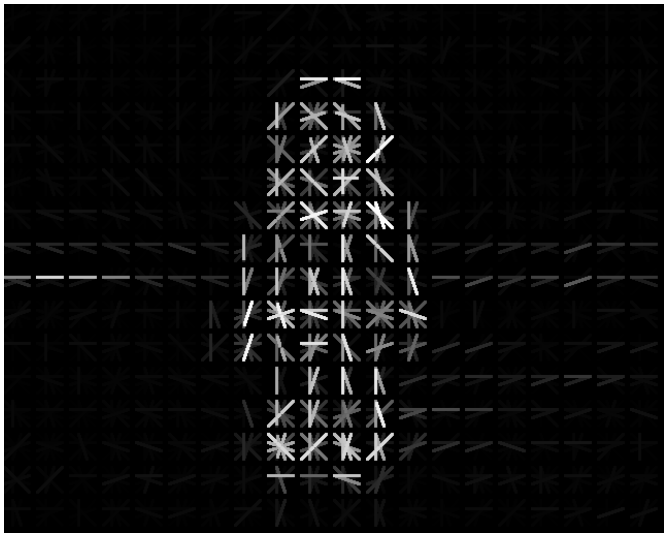


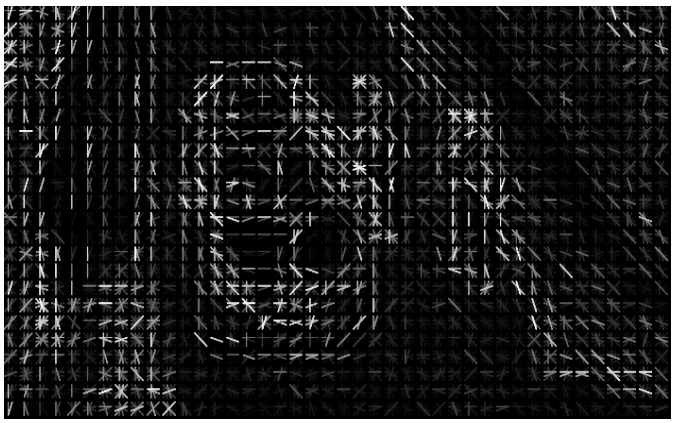
Thus, only by giving enough positive and negative samples will the threshold be set correctly. Note that in real examples it is often not possible to find such a perfect separation and some missed detections as well as false detections will occur, especially if something has not been seen before in the training.

As Camera Trainer is interactive, adding and removing samples is possible whenever needed, to adjust to changed conditions and to improve performance.

2.3 Camera Trainer: Based on Histogram of oriented Gaussians

Quite a few of the limitations of Camera Trainer are based on the fact that it uses Histogram of oriented Gaussians (HoG) as feature descriptors. For that, each sample is separated into many small patches. On each patch, the edges are determined and aggregated into a histogram. Find here two examples of images transformed into many small HoG patches:





Only objects with distinct edges can be detected, and the edge pattern needs to be sufficiently different from the background as well as all other objects in the field of view. Small details are easily lost. Thus, a person wearing a helmet or safety vest cannot be distinguished from one without. Similarly, a sports car would not be differentiable from a SUV.

In the end, the similarity between the target object model and any image patch will decide whether something is detected there or not. Where the threshold for this similarity lies (e.g where it separates target from background), has been learned along with the typical edge distribution and is part of the model.

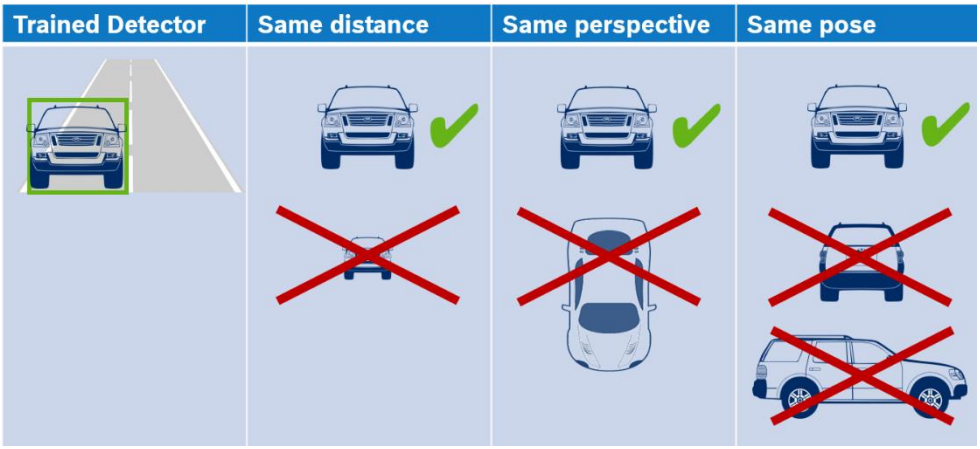
The larger the sample size in pixels, the more HoGs will be used. The more features and thresholds that are available, the more powerful the Camera Trainer detectors become to separate target from background and other objects. The minimal size for a Camera Trainer detector is 4x4 histogram patches, and only simple separations are possible with that. The detector sizes are also always restricted to a multitude of histogram patches.

2.4 Camera Trainer: Limitations on distance, pose and perspective

Camera Trainer detectors are limited to one size. Due to perspective, objects appear much smaller in the background than near the camera and cover far fewer pixels. This difference in size and pixel coverage within the image cannot be modeled with Camera Trainer. Thus a single detector is restricted concerning the detection distance, though further detectors can be trained to compensate for mid-range detections. Objects in the far distance cannot be detected at all, as they will not cover enough pixels for the Camera Trainer detector.

Camera Trainer detectors are also limited to a single pose and a single perspective. A car from the front looks completely different than a car from the side. A person walking looks quite different from a person crawling towards a fence, and a person from the front looks completely different than from a camera looking straight down on the person. The Camera Trainer detector is limited to a single edge pattern, and thus it cannot model these variances. When trying to do so with a single detector, the detector will become so general that a lot of other false detections will occur, rendering it useless. The best way to deal with these situations is to train several detectors.

Again, the assumption that a single camera only looks at a single field of view comes into play as this naturally restricts the perspective, and may also do so for the observable poses. Cars on the highway, for example, can only travel along the street and not perpendicular to it, and cars in a parking lot typically park correctly in the marked spaces.



2.5 Will machine learning, especially deep learning, replace current video analytics?

No. It will extend the possible applications.

Current intrusion detection has to cope with long detection distances, correspondingly few pixels on the target object, a wide variety of poses (walking, running, crawling, rolling) and needs to be able to deal with the unexpected. It can run on very low processing power. It is targeted towards moving objects, and will ignore all stationary ones.

Machine learning, on the other hand, needs high resolution and will thus only work in near range. It also needs high processing power, which is one of the main reasons adoption is not more widespread. Furthermore, it can only detect trained and expected objects. But with machine learning, objects can be properly classified, and close objects can be separated well. Non-moving objects can still be detected, even after a very long time without motion. Therefore machine learning in general is a good technology for applications like parking lot occupancy, traffic monitoring, as well as for counting people without false counts from shopping carts or baggage.

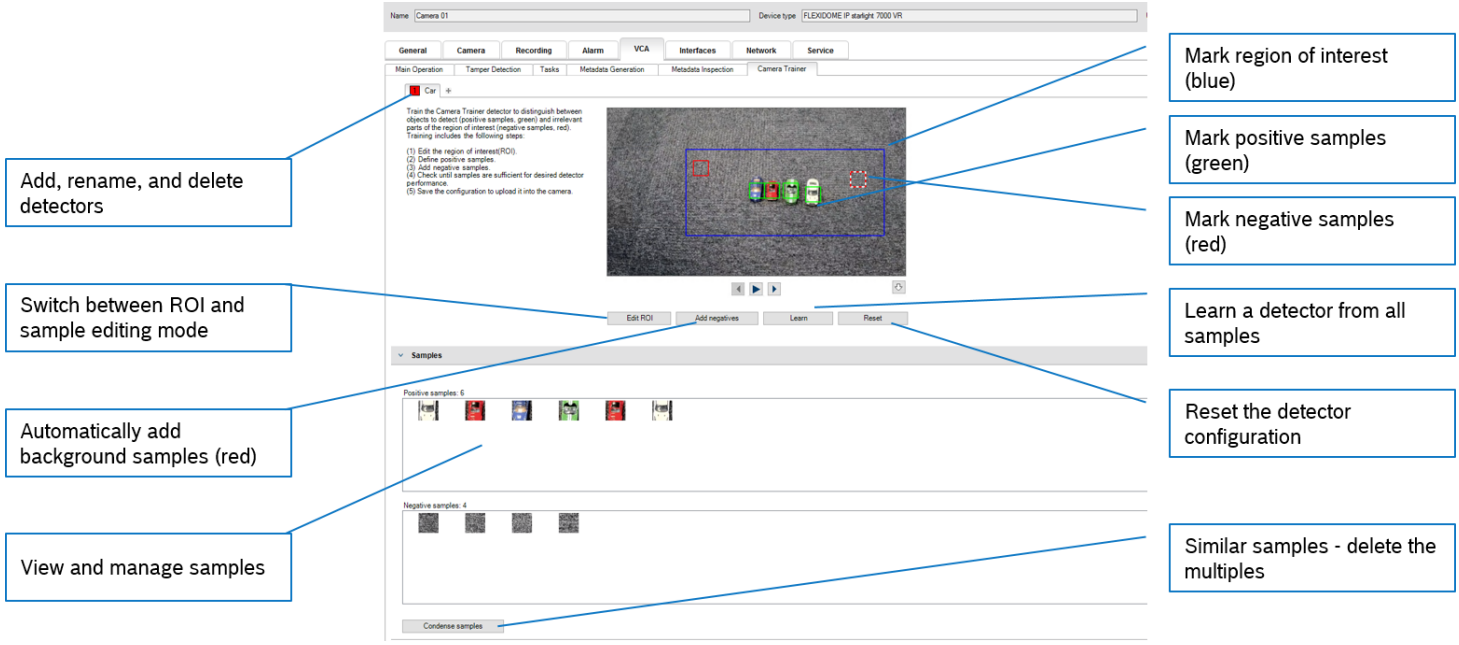
3 Configuration

3.1 Activate license

To activate the license in the Configuration Manager, select the camera, click the Service tab, then click the Licenses tab. In the activation key box enter the license key, then click Save. The activated licenses are displayed in the installed licenses list.

3.2 Training a detector

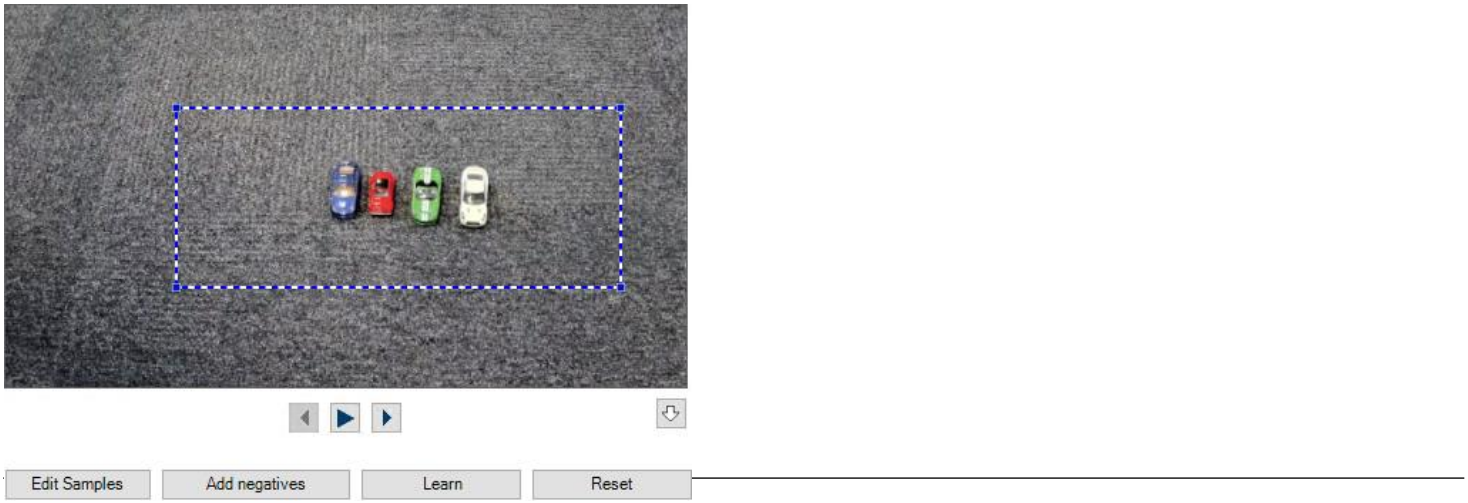
Overview



Region of interest (ROI)

For training a detector, the first step is to restrict the region of interest (ROI). Depending on the application, this can give a performance boost of up to 30% while also needing fewer samples. Part of this effect is caused by restricting the possible background, and part by the ever present perspective changes even within one field of view.

To define the ROI, first make sure that the toggle button for ROI and sample drawing shows “Edit Samples”. If “Edit ROI” is written there, then press this toggle button first. Then place a rectangle around the respective area in the camera image by clicking in the image, moving the mouse, and clicking a second time. The rectangle is displayed with blue-white boundary lines. To adjust the position: Place the pointer in the rectangle, press and hold the mouse button, then move the rectangle to the desired position. To adjust the size: Place the pointer over a boundary line or over one of the corners of the rectangle, press and hold the mouse button, then move the rectangle or corner to the desired position.



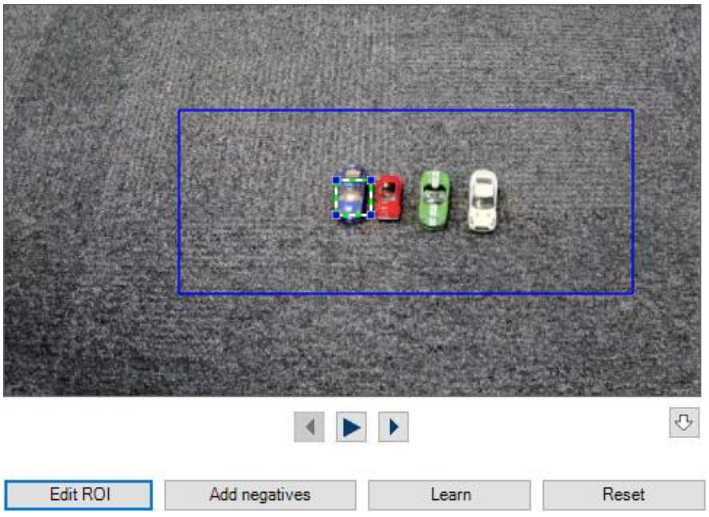
Target edge pattern & size
The next step is to determine which object and which part of the object the detector should focus on. Remember that edges are the main feature, and thus always make the detector a bit larger than the outer edges, so the edges are well-captured by the detector.

Now take for example a car from the front. One possible solution for a detector can be to train the full car front. Another option is to only train on the windshield area. When training on the full front, more variation between cars needs to be covered, and thus more samples may be needed. Furthermore, part of the car front will be occluded more easily. While a bit of occlusion will typically not be a problem, significant occlusion will prevent detection of the car. When training only on the windshield area, the detector will be smaller, needing fewer samples but also being less discriminative. Thus, a false detection on the vehicle hood would be possible. On the other hand, it is less likely that the windshield will be partly occluded. Both options can result in good performance.

Occlusion prevention and restriction is also often possible by careful selection of camera placement and field of view.

Draw the first sample
Click “Edit Samples” to toggle from ROI drawing mode to sample drawing mode. The ROI rectangle is now displayed in solid blue, and cannot be adjusted in this mode. Place a rectangle around the positive sample (object) in the camera image. To do this, click the camera image and adjust size and position of the automatically inserted rectangle. The rectangle around a sample is displayed with green-white boundary lines when selected, and solid green boundary lines otherwise. To adjust the position: Place the pointer in a rectangle, press and hold the mouse button, then move the rectangle to the desired position. To adjust the size: Place the pointer over a boundary line or over one of the corners of the rectangle, press and hold the mouse button, then move the rectangle or corner to the desired position.

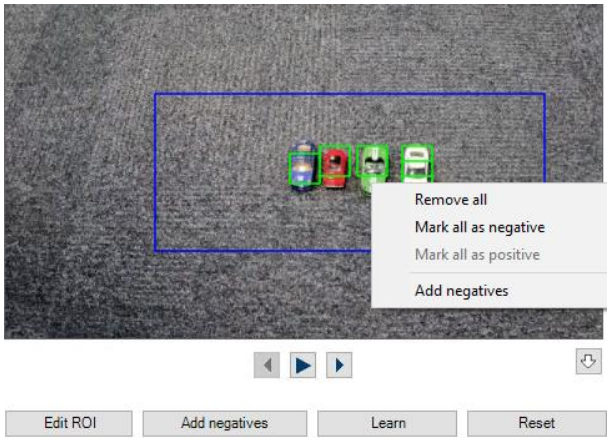
You can only resize the first sample. Therefore the first samples will also be the only one that has blue edge points. All other added samples use the same size and aspect ratio. If resizing is necessary after you have added a second sample, click Reset, then redefine the detector.

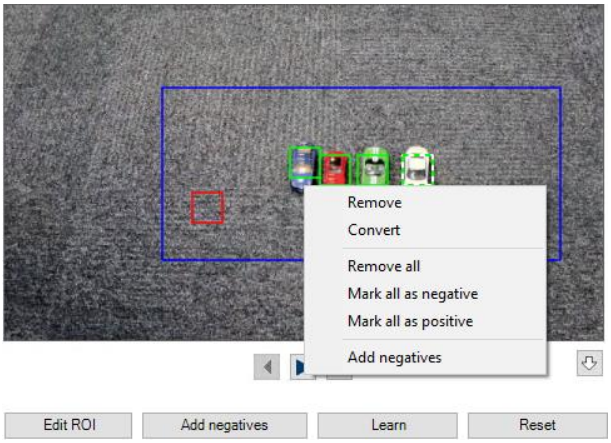


Mark other examples of the target object
To add additional samples (after the first), click the center of the desired area to place a positive sample automatically. Shift+click creates a negative sample. Due to the lower resolution of the Camera Trainer module, the displayed position and size of the region of interest and samples can differ to its position and size drawn using the mouse. The nearest possible position is set automatically by the system.

Double-click the sample to change a positive sample to a negative sample or back, or right-click on the sample and select “Convert” from the shortcut menu. Samples can also be removed with right-click on the sample and selection of “Remove”.

Note that the video menu differs depending on whether a sample has been selected or not.





Add background examples

Click Add negatives to automatically add negative samples. All negative samples are displayed as rectangles with red boundary lines. The Add negatives function can be performed several times. New samples will be added if there is sufficient space.

You can also manually place rectangles on negative samples by Shift+click or with a double click.

Learn

Click Learn. The samples are used to train the detector and will be added to the samples list. The drawn samples will vanish from the image, instead the current detector results will now be shown with green boundaries.

Iterate to obtain desired detection performance

Repeat the steps by adding more positive and negative samples. Ensure that the samples show a large variety.

Sample management section

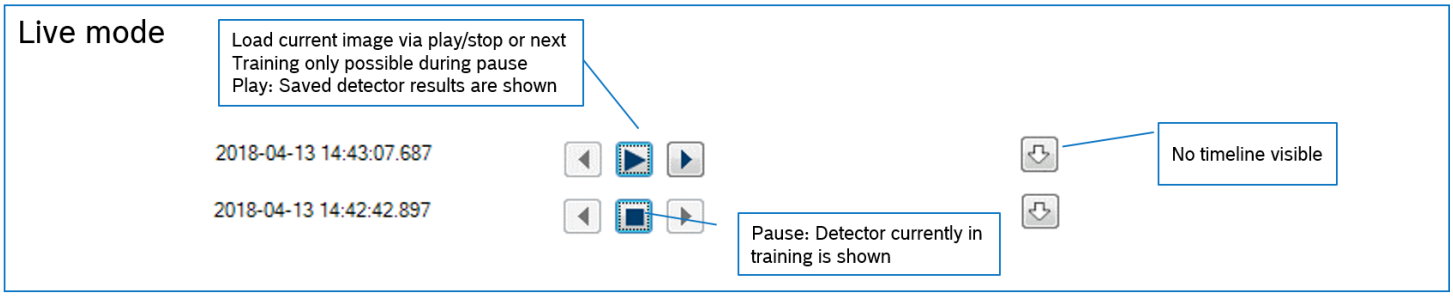
Click the Samples group to display and manage the positive and negative samples. To delete a sample: Click a sample, and then press the Delete key. To copy a sample: Drag the sample from the Positive samples area to the Negative samples area or vice versa.

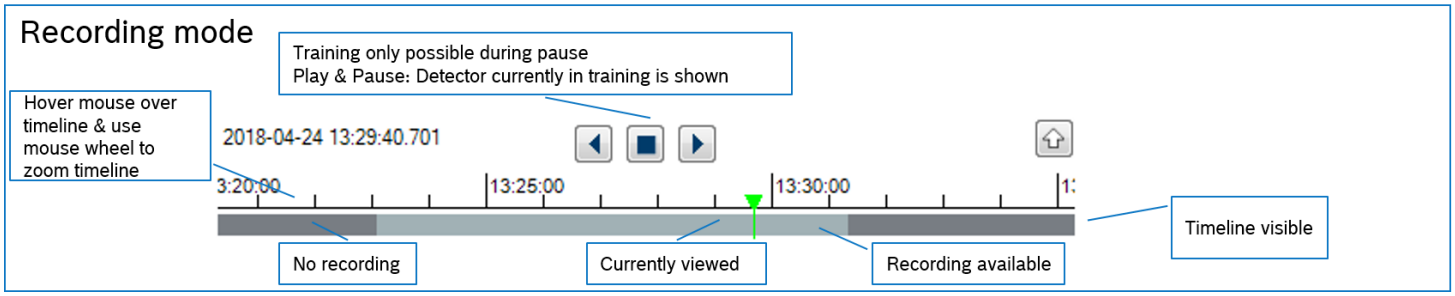
Optional: Click Condense to delete similar samples. You can use this function optionally if you have many samples and you want to delete the similar ones. This is not necessary if you remain within the limit of 1024 positive and 1024 negative samples.

Be careful when changing the Samples group, and remember to click Learn afterwards. Also, note that each time you click Learn, the samples in the camera image will be added to the samples group and all samples - stored in memory – will be used to train the detector. Thus when removing a sample from the Samples group, make sure it is not still marked in the image, as it would thus be re-added to the Samples group.

3.3 Using recording for training

In order to provide a wide variety of samples and background conditions, recording available through the camera can be used for training Camera Trainer detectors as well. Open the timeline via the downward arrow under the video. Note that while in live play mode, the saved detector results are shown via the general yellow/red metadata visualization, in the recording play mode it will be the current, possibly unsaved detector being shown there with green samples.



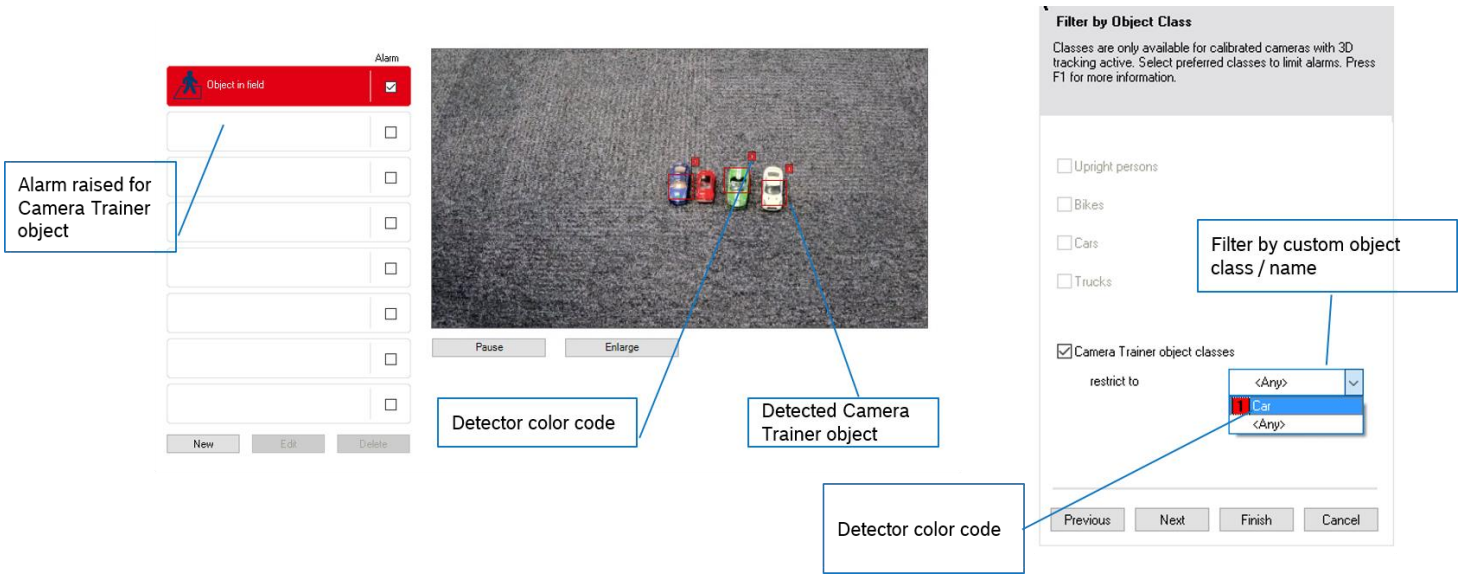


3.4 Configuring alarm or counting tasks

Camera Trainer objects are handled the same as other VCA objects. Find here an overview which tasks and object filters support them, and note that Camera Trainer objects have their own class that corresponds to their name.

Tasks supporting Camera Trainer objects	**Tasks not available**
Inside field	Idle object
Enter field	Removed object
Exit field	Crowd detection
Line crossing	Tampering
Loitering	Flow
Route	Counter flow
Counter	
Occupancy	
Condition change	
Similarity search	

Object filters supporting Camera Trainer objects	**Object filters not available**
Class	Speed
Direction	Color
Size	
Aspect ratio	



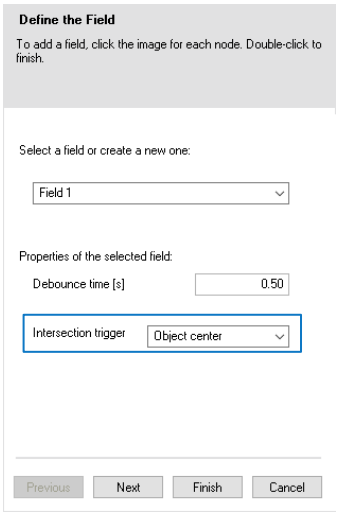
3.5 Alarm on trained object being absent

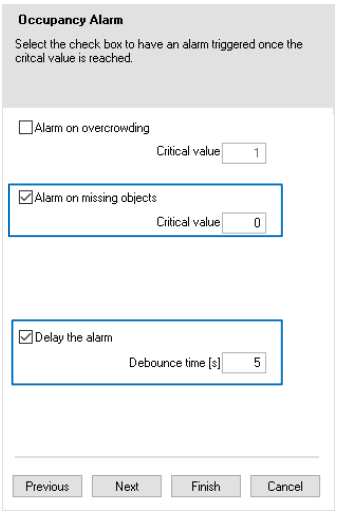
With Camera Trainer, you can not only alarm when an object is detected, but also when a trained object is not visible anymore. Be it that the object is occluded, has been moved away, or that a trained landmark in the image is no longer at its position due to camera tampering like rotating the camera or blinding it.

In order to alarm on that, choose an Occupancy Task. Set a small area around the target position, and set the intersection trigger to object center. Select “Alarm on missing objects” with critical value 0 if the alarm should only be send if no object is available there.

Determine how much time shall pass until the alarm is send. Note that when switching camera day/night modes, objects will be lost shortly, and to avoid alarms a debounce time of at least 5 seconds is recommended.

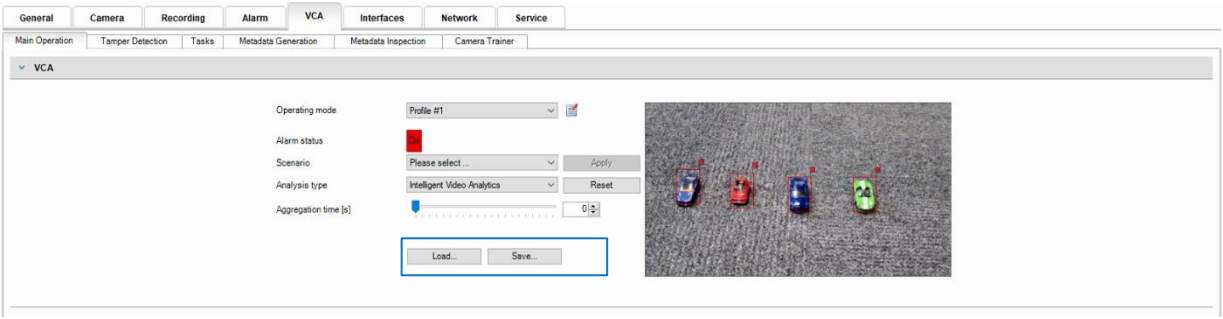
Note that the debounce time of the field can only delay on objects being present, not when objects are lost, which would cause an immediate alarm. Therefore use the debounce time of “Delay the alarm” instead, to alarm on lost objects only if they are not there for more than the selected debounce time.





3.6 Backup & transfer to other cameras

Via Configuration Manager, the video analytics configuration including Camera Trainer detector and samples can be saved and loaded. Select a camera, click the VCA tab, and on the Main Operation page find the Load and Save buttons.



Remember that, due to size limitations, the Camera Trainer training samples are not stored in the camera but stored instead in the Configuration Manager used to train the Camera Trainer detectors. To save the configuration click on the Main Operation tab, then click on Save. Select the location where you wish to store the configuration file and then click the Save button. This mechanism can also be used to transfer the samples to another computer and Configuration Manager, therefore be sure to always backup the video analytics configuration.

A second way to backup training samples is to copy or drag and drop them from the Camera Trainer samples section into a file folder and back. You can easily select all positive or all negative samples with Ctr-A. You can then copy them with Ctr-C and paste in the folder or into the Camera Trainer samples section with Ctr-V. Alternatively, you can select them with the mouse and drag and drop them.

Make sure to use two different folders for positive and negative samples, to keep them well separated. When dragging samples from a file folder into the Camera Trainer samples section, only samples of the correct detector size will be accepted.

3.7 Tips, Tricks & FAQs

Why do I get false detections in live view, but not on the stopped image?
In the live view, the detection is done inside the camera on the uncompressed images. These are the yellow or blue bounding boxes, depending on alarm state of the object. When working on the stopped image, in the training mode with green and red rectangles for positive and negative samples, the detection is done inside the Configuration Manager on compressed images. To get rid of the false detections in the live view, manually mark these areas as negative samples a few times and press learn. Typically, this will get rid of these false detections in live view.

Will the video stream quality affect the detection performance?
Yes and no. While the detection in the camera is independent of any encoding parameters, the training in the Configuration Manager is not. Make sure to use a very good quality image stream with very few compression artifacts for training. Otherwise, the training becomes harder, as the training may focus more on compression artifacts than on real content.

I cannot place the samples as exact as I want to, why?
Due to hardware restrictions, the detector will not be applied to every single pixel position, but regularly skip across pixel. For best performance, this is mirrored in the training process as well.

How many detectors can I train?
16 Camera Trainer detectors can be trained in total.
Please note that when all of them use the full image as ROI and the largest possible sample size, only 10 can run in real-time, and using the full 16 detectors will result in roughly half the usual frame rate for video analytics.

Bosch Sicherheitssysteme GmbH
Robert-Bosch-Ring 5
85630 Grasbrunn
Germany
www.boschsecurity.com
© Bosch Sicherheitssysteme GmbH, 2019