WHITE PAPER

AI in video analytics

Considerations for analytics based on machine learning and deep learning

March 2021

AXIS COMMUNICATIONS

Table of Contents

1 Summary

AI-based video analytics is one of the most discussed topics in the video surveillance industry. Some of the applications can substantially speed up data analysis and automate repetitive tasks. But AI solutions today cannot replace the human operator's experience and decision-making skills. The strength lies instead in a combination: taking advantage of AI solutions to improve and increase human efficiency.

The AI concept incorporates machine learning algorithms and deep learning algorithms. Both types automatically build a mathematical model, using substantial amounts of sample data (*training data)*, to gain the ability to calculate results without being specifically programmed for it. An AI algorithm is developed through an iterative process, in which a cycle of collecting training data, labeling training data, using the labeled data to train the algorithm, and testing the trained algorithm, is repeated until the desired quality level is reached. After this, the algorithm is ready to use in an analytics application which can be purchased and deployed on a surveillance site. At this point, all the training is done and the application will not learn anything new.

A typical task for AI-based video analytics is to visually detect humans and vehicles in a video stream and distinguish which is which. A *machine learning* algorithm has learned the combination of visual features that defines these objects. A *deep learning* algorithm is more refined and can - if trained for it - detect much more complex objects. But it also requires substantially larger efforts for development and training and much more computation resources when the finalized application is used. For well-specified surveillance needs, it should therefore be considered whether a dedicated, optimized machine learning application can be sufficient.

Algorithm development and increasing processing power of cameras have made it possible to run advanced AI-based video analytics directly on the camera (edge based) instead of having to perform the computations on a server (server based). This enables better realtime functionality because the applications have immediate access to uncompressed video material. With dedicated hardware accelerators, such as MLPU (machine learning processing unit) and DLPU (deep learning processing unit), in the cameras, edge-based analytics can be more power-efficiently implemented than with a CPU or GPU (graphics processing unit).

Before an AI-based video analytics application is installed, the manufacturer's recommendations based on known preconditions and limitations must be carefully studied and followed. Every surveillance installation is unique, and the application's performance should be evaluated at each site. If the quality is found to be lower than expected, investigations should be made on a holistic level, and not focus only on the analytics application itself. The performance of video analytics is dependent on many factors related to camera hardware, camera configuration, video quality, scene dynamics, and illumination. In many cases, knowing the impact of these factors and optimizing them accordingly makes it possible to increase video analytics performance in the installation.

As AI is increasingly applied in surveillance, the advantages of operational efficiency and new use cases must be balanced with a mindful discussion about when and where to apply the technology.

2 Introduction

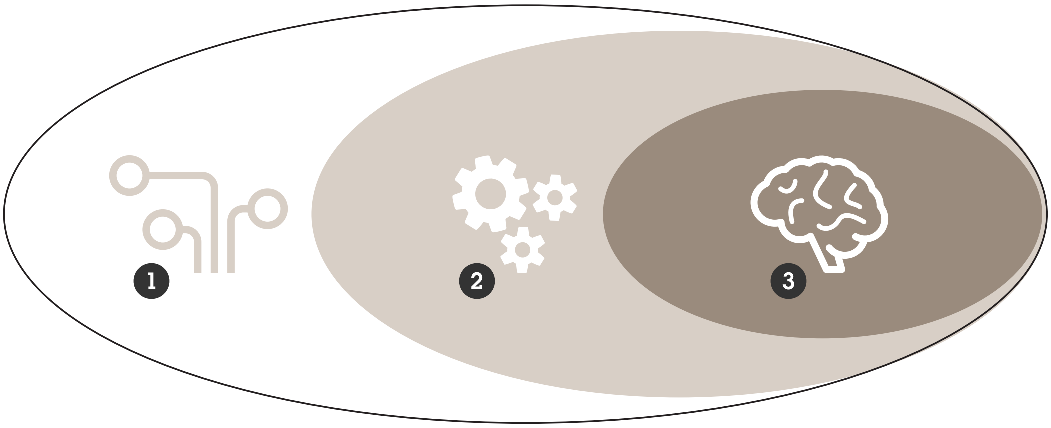
AI, artificial intelligence, has been developed and debated ever since the first computers were invented. While the most revolutionary incarnations are not yet here, AI-based technologies are widely used today for carrying out clearly defined tasks in applications such as voice recognition, search engines, and virtual assistants. AI is also increasingly employed in healthcare where it provides valuable resources in, for example, x-ray diagnostics and retina scan analysis.

AI-based video analytics is one of the most discussed topics in the video surveillance industry and expectations are high. There are applications on the market that use AI algorithms to successfully speed up data analysis and automate repetitive tasks. But in a wider surveillance context, AI today and in the near future should be viewed as just one element, among several others, in the process of building accurate solutions.

This white paper provides a technological background on machine learning and deep learning algorithms and how they can be developed and applied for video analytics. This includes a brief account of AI acceleration hardware and the pros and cons of running AI-based analytics on the edge compared to on a server. The paper also takes a look at how the preconditions for AI-based video analytics performance can be optimized, taking a wide scope of factors into account.

3 AI, machine learning, and deep learning

Artificial intelligence (AI) is a wide concept associated with machines that can solve complex tasks while demonstrating seemingly intelligent traits. Deep learning and machine learning are subsets of AI.



1 *Artificial intelligence*
2 *Machine learning*
3 *Deep learning*

3.1 Machine learning

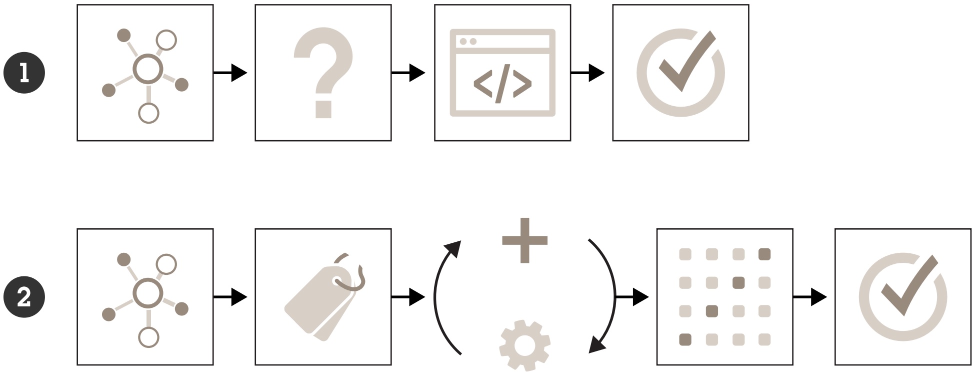
Machine learning is a subset within AI that uses statistical learning algorithms to build systems that have the ability to automatically learn and improve during training without being explicitly programmed.

In this section, we distinguish between traditional programming and machine learning in the context of *computer vision* – the discipline of making computers understand what is happening in a scene by analyzing images or videos.

Traditionally programmed computer vision is based on methods that calculate an image's *features*, for example, computer programs looking for pronounced edges and corner points. These features need to be manually defined by an algorithm developer who knows what is important in the image data. The developer then combines these features for the algorithm to conclude what is found in the scene.

Machine learning algorithms automatically build a mathematical model using substantial amounts of sample data – *training data* – to gain the ability to make decisions by calculating results without specifically being programmed to do so. The features are still hand-crafted but how to combine these features is learned by the algorithm itself through exposure to large amounts of labeled, or *annotated*, training data. In this paper, we refer to this technique of using hand-crafted features in learned combinations, as *classical machine learning*.

In other words, for a machine learning application we need to train the computer to get the program we want. Data is collected and then annotated by humans, sometimes assisted with pre-annotation by server computers. The result is fed into the system and this process goes on until the application has learned enough to detect what we wanted, for example, a specific type of vehicle. The trained model becomes the program. Note that when the program is finished the system does not learn anything new.



1 *Traditional programming:*
Data is collected. Program criteria are defined. The program is coded (by a human). Done.

2 *Machine learning:*
Data is collected. Data is labeled. The model undergoes an iterative training process. The finalized trained model becomes the program. Done.

The advantage of AI over traditional programming, when building a computer vision program, is the ability to process extensive data. A computer can go through thousands of images without losing focus, whereas a human programmer will get tired and unfocused after a while. That way, the AI can make the application substantially more accurate. However, the more complicated the application, the harder it is for the machine to produce the wanted result.

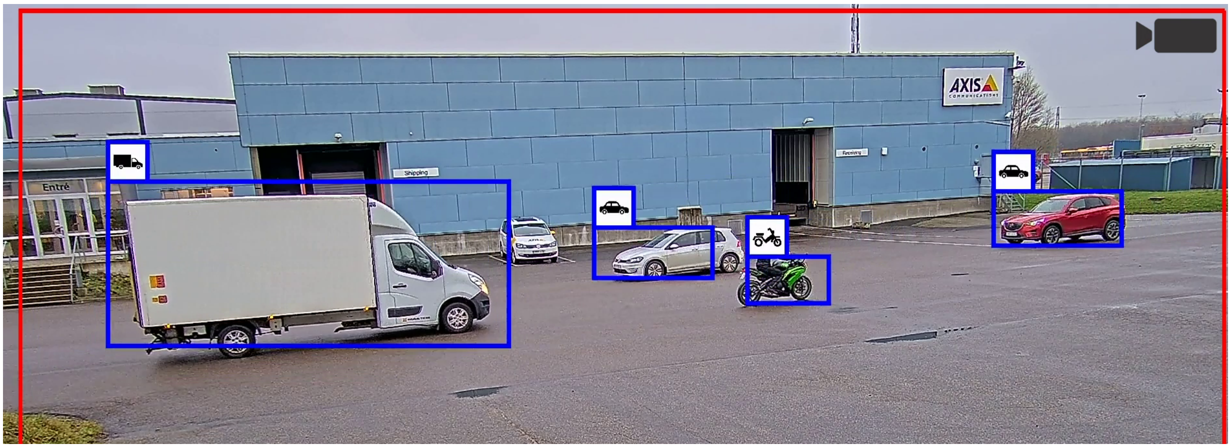
3.2 Deep learning

Deep learning is a refined version of machine learning in which both the feature extraction and how to combine these features, in deep structures of rules to produce an output, are learned in a data-driven manner. The algorithm can automatically define what features to look for in the training data. It can also learn very deep structures of chained combinations of features.

The core of the algorithms used in deep learning is inspired by how neurons work and how the brain uses these to form higher-level knowledge by combining the neuron outputs in a deep hierarchy, or a *network*, of chained rules. The brain is a system in which the combinations themselves are also formed by neurons, erasing the distinction between feature extraction and the combination of features, making them the same in some sense. These structures were simulated by researchers into something called *artificial neural networks*, which is the most widely used type of algorithm in deep learning. See the appendix of this document for a brief overview of neural networks.

Using deep learning algorithms, it is possible to build intricate visual detectors and automatically train them to detect very complex objects, resilient to scale, rotation, and other variations.

The reason behind this flexibility is that deep learning systems can learn from a much larger amount of data, and much more varied data, than classical machine learning systems. In most cases, they will significantly outperform hand-crafted computer vision algorithms. This makes deep learning especially suited for complex problems where the combination of features cannot easily be formed by human experts, such as image classification, language processing, and object detection.



Object detection based on deep learning can classify complex objects. In this example, the analytics application can not only detect vehicles, but also classify the type of vehicle.

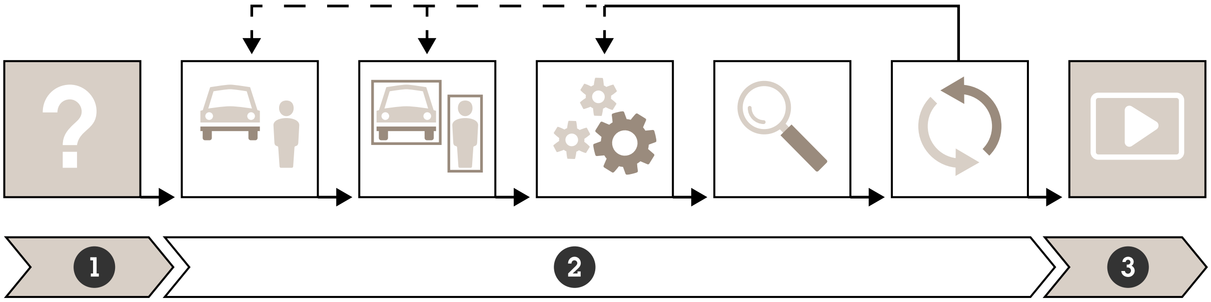
3.3 Classical machine learning vs. deep learning

While they are similar types of algorithms, a deep learning algorithm typically uses a much larger set of learned feature combinations than a classical machine learning algorithm does. This means that deep learning-based analytics can be more flexible and can - if trained to - learn to perform much more complex tasks.

For specific surveillance analytics, however, a dedicated, optimized classical machine learning algorithm can be sufficient. In a well specified scope, it can provide similar results as a deep learning algorithm while requiring less mathematical operations and can therefore be more cost-efficient and less power consuming to use. It furthermore requires much less training data and this greatly reduces the development effort.

4 The stages of machine learning

The development of a machine learning algorithm follows a series of steps and iterations, roughly visualized below, before a finalized analytics application can be deployed. At the heart of an analytics application is one or more algorithms, for example an object detector. In the case of deep learning based applications the core of the algorithm is the deep learning model.



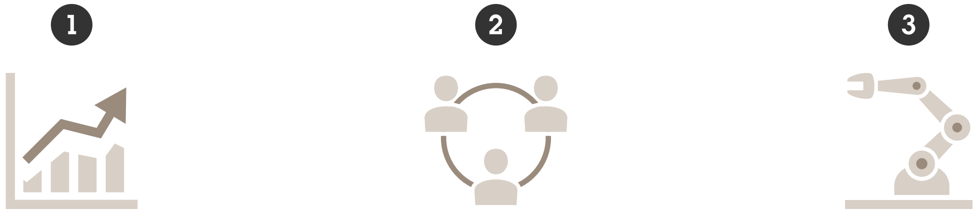
1 *Preparation: Defining the purpose of the application.*
2 *Training: Collecting training data. Annotating the data. Training the model. Testing the model. If quality is not as expected, the previous steps are done again in an iterative improvement cycle.*
3 *Deployment: Installing and using the finished application.*

4.1 Data collection and data annotation

To develop an AI-based analytics application you need to collect large amounts of data. In video surveillance, this typically consists of images and video clips of humans and vehicles or other objects of interest. In order to make the data recognizable for a machine or computer a data annotation process is necessary, where the relevant objects are categorized and labeled. Data annotation is mainly a manual and labor-intense task. The prepared data needs to cover a large-enough variety of samples that are relevant for the context where the analytics application will be used.

4.2 Training

Training, or learning, is when the model is fed annotated data and a training framework is used to iteratively modify and improve the model until the desired quality is reached. In other words, the model is optimized to solve the defined task. Training can be done according to one of three main methods.



1 *Supervised learning: the model learns to make accurate predictions*
2 *Unsupervised learning: The model learns to identify clusters*
3 *Reinforcement learning: The model learns from mistakes*

4.2.1 Supervised learning

Supervised learning is the most used method in machine learning today. It can be described as learning by examples. The training data is clearly annotated, meaning that the input data is already paired with the desired output result.

Supervised learning generally requires a very large amount of annotated data and the performance of the trained algorithm is directly dependent on the quality of the training data. The most important quality aspect is to use a dataset that represents all potential input data from a real deployment situation. For object detectors, the developer must make sure to train the algorithm with a wide variety of images, with different objects instances, orientations, scales, light situations, backgrounds, and distractions. Only if the training data is representative for the planned use case, the final analytics application will be able to make accurate predictions also when processing new data, unseen during the training phase.

4.2.2 Unsupervised learning

Unsupervised learning uses algorithms to analyze and group unlabeled datasets. This is not a common training method in the surveillance industry, because the model requires a lot of calibration and testing while the quality can still be unpredictable.

The datasets must be relevant for the analytics application but do not have to be clearly labeled or marked. The manual annotation work is eliminated, but the number of images or videos needed for the training must be greatly increased, by several orders of magnitude. During the training phase, the to-be-trained model is identifying, supported by the training framework, common features in the datasets. This enables it to, during the deployment phase, group data according to patterns while also allowing it to detect anomalies which do not fit into any of the learned groups.

4.2.3 Reinforcement learning

Reinforcement learning is used in, for example, robotics, industrial automation, and business strategy planning, but due to the need for large amounts of feedback, the method has limited use in surveillance today. Reinforcement learning is about taking suitable action to maximize the potential *reward* in a specific situation, a reward that gets larger when the model makes the right choices. The algorithm does not use data/label pairs for training, but is instead optimized by testing its decisions through interaction with the environment while measuring the reward. The goal of the algorithm is to learn a policy for actions that will help maximize the reward.

4.3 Testing

Once the model is trained, it needs to be thoroughly tested. This step typically contains an automated part complemented with extensive testing in real-life deployment situations.

In the automated part, the application is benchmarked with new datasets, unseen by the model during its training. If these benchmarks are not where they are expected to be, the process starts over again: new training data is collected, annotations are made or refined and the model is retrained.

After reaching the wanted quality level, a field test starts. In this test, the application is exposed to real world scenarios. The amount and variation depend on the scope of the application. The narrower the scope, the less variations need to be tested. The broader the scope, the more tests are needed.

Results are again compared and evaluated. This step can then again cause the process to start over. Another potential outcome could be to define preconditions, explaining a known scenario in which the application is not or only partly recommended to be used.

4.4 Deployment

The deployment phase is also called inference or prediction phase. *Inference* or *prediction* is the process of executing a trained machine learning model. The algorithm uses what it learned during the training phase to produce its desired output. In the surveillance analytics context, the inference phase is the application running on a surveillance system monitoring real life scenes.

To achieve real-time performance when executing a machine learning based algorithm on audio or video input data, specific hardware acceleration is generally required.

5 Edge-based analytics

High-performance video analytics used to be server based because they required more power, and cooling, than a camera could offer. But algorithm development and increasing processing power of edge devices in recent years have made it possible to run advanced AI-based video analytics on the edge.

There are obvious advantages of edge based analytics applications: they have access to uncompressed video material with very low latency, enabling realtime applications while avoiding the additional cost and complexity of moving data into the cloud for computations. Edge based analytics also come with lower hardware and deployment costs since less server resources are needed in the surveillance system.

Some applications may benefit from using a combination of edge based and server based processing, with preprocessing on the camera and further processing on the server. Such a hybrid system can facilitate cost-efficient scaling of analytics applications by working on several camera streams.

6 Hardware acceleration

While you can often run a specific analytics application on several types of platforms, using dedicated hardware acceleration achieves a much higher performance when power is limited. Hardware accelerators enable power-efficient implementation of analytics applications. They can be complemented by server and cloud compute resources when suitable.

- **GPU (graphics processing unit).** GPUs were mainly developed for graphics processing applications but are also used for accelerating AI on server and cloud platforms. While sometimes also used in embedded systems (edge), GPUs are not optimal, from a power efficiency standpoint, for machine learning inference tasks.

- **MLPU (machine learning processing unit).** An MLPU can accelerate inference of specific classical machine learning algorithms for solving computer vision tasks with very high power efficiency. It is designed for real-time object detection of a limited number of simultaneous object types, for example, humans and vehicles.

- **DLPU (deep learning processing unit).** Cameras with a built-in DLPU can accelerate general deep learning algorithm inference with high power efficiency, allowing for a more granular object classification.

7 AI is still in its early development

It is tempting to make a comparison between the potential of an AI solution and what a human can achieve. While human video surveillance operators can only be fully alert for a short period of time, a computer can keep processing large amounts of data extremely quickly without ever getting tired. But it would be a fundamental misunderstanding to assume that AI solutions would replace the human operator. The real strength lies in a realistic combination: taking advantage of AI solutions to improve and increase the efficiency of a human operator.

Machine learning or deep learning solutions are often described as having the capability to automatically learn or improve through experience. But AI systems available today do *not* automatically learn new skills after deployment and will *not* remember specific events that have occurred. To improve the system's performance, it needs to be retrained with better and more accurate data during supervised learning sessions. Unsupervised learning typically requires a lot of data to generate clusters and is therefore not used in video surveillance applications. It is instead used today mainly for analyzing large datasets to find anomalies, for example in financial transactions. Most approaches that are promoted as "self-learning" within video surveillance are based on a statistical data analysis and not on actually retraining the deep learning models.

Human experience still beats many AI-based analytics applications for surveillance purposes. Especially those which are supposed to perform very general tasks and where contextual understanding is critical. A machine learning based application might successfully detect a "running person" if specifically trained for it but unlike a human who can put the data into context, the application has no understanding of why the person is running – to catch the bus or flee from the nearby running police officer? Despite promises from companies applying AI in their analytics applications for surveillance, the application cannot yet understand what it sees on video with remotely the same insight as a human can.

For the same reason, AI-based analytics applications can also trigger false alarms or miss alarms. This could typically happen in a complex environment with a lot of movement. But it could also be about, for example, a person carrying a large object — effectively obstructing the human characteristics to the application, making a correct classification less likely.

AI-based analytics today should be used in an assisting way, for example, to roughly determine how relevant an incident is before alerting a human operator to decide about the response. This way, AI is used to reach scalability and the human operator is there to assess potential incidents.

8 Considerations for optimal analytics performance

To navigate the quality expectations of an AI-based analytics application, it is recommended to carefully study and understand the known preconditions and limitations, typically listed in the application's documentation.

Every surveillance installation is unique and the application's performance should be evaluated at each site. If the quality is not at the expected or anticipated level, it is strongly recommended to not only focus the investigation on the application itself. All investigations should be made on a holistic level because the performance of an analytics application depends on so many factors, most of which can be optimized if we are aware of their impact. These factors include, for example, camera hardware, video quality, scene dynamics, illumination level, as well as camera configuration, position, and direction.

8.1 Image usability

Image quality is often said to depend on high resolution and high light sensitivity of the camera. While the importance of these factors cannot be questioned, there are certainly others that are just as influential for the actual *usability* of an image or a video. For example, the best quality video stream from the most expensive surveillance camera can be useless if the scene is not sufficiently lit at night, if the camera has been redirected, or if the system connection is broken.

The placement of the camera should be carefully considered before deployment. For video analytics to perform as expected, the camera needs to be positioned to enable a clear view, without obstacles, of the intended scene.

Image usability may also depend on the use case. Video that looks good to a human eye may not have the optimal quality for the performance of a video analytics application. In fact, many image processing methods that are commonly used to enhance video appearance for human viewing are not recommended when using video analytics. This may include, for example, applied noise reduction methods, wide dynamic range methods, or auto exposure algorithms.

Video cameras today often come with integrated IR illumination which enables them to work in complete darkness. This is positive as it may enable cameras to be placed on difficult-light sites and reduce the need for installing additional illumination. However, if heavy rain or snowfall are expected on a site, it is highly recommended not to rely on light coming from the camera or from a location very close to the camera. Too much light may be directly reflected back to the camera, against raindrops and snowflakes, making the analytics unable to perform. With ambient light, on the other hand, there is a better chance that the analytics will deliver some results even in difficult weather.

8.2 Detection distance

It is difficult to determine a maximum detection distance of an AI-based analytics application — an exact datasheet value in meters or feet can never be the whole truth. Image quality, scene characteristics, weather conditions, and object properties such as color and brightness have a significant impact on the detection distance. It is evident, for example, that a bright object against a dark background during a sunny day can be visually detected at much longer distances than a dark object on a rainy day.

The detection distance also depends on the speed of the objects to be detected. To achieve accurate results, a video analytics application needs to "see" the object during a sufficiently long period of time. How long that period needs to be depends on the processing performance (framerate) of the platform: the lower the processing performance, the longer the object needs to be visible in order to be detected. If the camera's shutter time is not well matched with the object speed, motion blur appearing in the image may also lower the detection accuracy.

Fast objects may be more easily missed if they are passing by closer to the camera. A running person located far from the camera, for example, might be well detected, while a person running very close to the camera at the same speed may be in and out of the field of view so quickly that no alarm is triggered.

In analytics based on movement detection, objects moving directly towards the camera, or away from it, present another challenge. Detection will be especially difficult for slow-moving objects, which will only cause very small changes in the image compared to movement across the scene.

A higher resolution camera typically does not provide a longer detection distance. The processing capabilities needed for executing a machine learning algorithm are proportional to the size of the input data. This means that the processing power required to analyze the full resolution of a 4K camera is at least four times higher than for a 1080p camera. It is very common to run AI-based applications on a lower resolution than the camera or stream can offer due to limitations in the camera's processing capability.

8.3 Alarms and recording setup

Because of the various levels of filters they apply, object analytics generate very few false alarms. But object analytics perform as they should only when their listed preconditions are all met. In other cases, they might instead miss important events.

If it is not absolutely certain that all conditions will be met at all times, it is therefore recommended to take a conservative approach and set up the system so that a specific object classification is not the only alarm trigger. This will cause more false alarms but also reduce the risk of missing something important. When alarms or triggers go directly to an alarm monitoring center, each false alarm becomes very costly. There is an obvious need for a reliable object classification to filter out unwanted alarms. But the recording solution still can and should be setup to rely not only on the object classification. In the case of a missed real alarm, this setup allows you to assess, from the recording, the reason for missing the alarm and then to improve the overall installation and configuration.

If the object classification is done on the server during an incident search, it is recommended to configure the system to continuous recording and not filter the initial recording at all. Continuous recording consumes a lot of storage but this is, to some extent, compensated for by modern compression algorithms like Zipstream.

8.4 Maintenance

A surveillance installation should be regularly maintained. Physical inspections, and not only viewing the video through the VMS interface, is recommended in order to discover and remove anything that might disturb or block the field of view. This is important also in standard, recording-only installations, but is even more critical when using analytics.

In the context of basic video motion detection, a typical obstacle such as a spider's web that sways in the wind could increase the number of alarms, resulting in a higher storage consumption than necessary. With object analytics, the web would basically create an exclude zone in the detection area. Its threads would obscure objects and greatly reduce the chance of detection and classification.

Spider webs might disturb a surveillance camera's field of view.

Dirt on the front glass or bubble of the camera is unlikely to cause problems during daytime. But in low-light conditions, light that hits a dirty bubble from the side, for example from the headlights of a car, can cause unexpected reflections that may decrease detection accuracy.

Scene-related maintenance is equally important as camera maintenance. During the lifetime of a camera, a lot can happen in the scene it is monitoring. A simple before-and-after image comparison will reveal potential problems. What did the scene look like when the camera was deployed and what does it look like today? Is there a need to adjust the detection zone? Should the camera's field of view be adjusted, or should the camera be moved to a different location?

9 Privacy and personal integrity

Working with security and surveillance requires balancing individual rights to privacy and personal integrity with the ambition to increase safety by preventing crimes or enabling forensic investigations. In the specific installation and use case, this requires careful ethical consideration as well as understanding and applying local legislation. It also places requirements on the solution to, for example, ensure cybersecurity and prevent unintentional access to video material. At the same time, edge-based analytics and the generation of metadata for statistical purposes may increase privacy protection if only anonymized data is transmitted for later processing.

With the increasing application of automated analytics in surveillance systems, some new aspects must be taken into consideration. Because the analytics applications come with a risk of false detections it is important that the decision process involves an experienced operator or user. This is often referred to as keeping a "human in the loop". Moreover, it is important to recognize that the human decision may be affected by how the alarm is generated and presented. Without proper training and awareness of the functionality of the analytics solution, the wrong conclusions could be drawn.

Additional concern can be caused by the way that deep learning algorithms are developed, and for some use cases this requires a cautious approach when applying the technology. The quality of these algorithms is fundamentally linked to the datasets, that is, the videos and images, used for training the algorithm. Tests have shown that if that material is not carefully selected, some AI systems may exhibit both ethnic and gender bias in the detections. This has prompted an open discussion and given rise to both legislative limitations and activities to ensure that such aspects are addressed during the development of the systems.

As AI is increasingly being applied in surveillance, it is important to balance the advantages of operational efficiency and new potential use cases with a mindful discussion about where and when to apply the technology.

10 Appendix

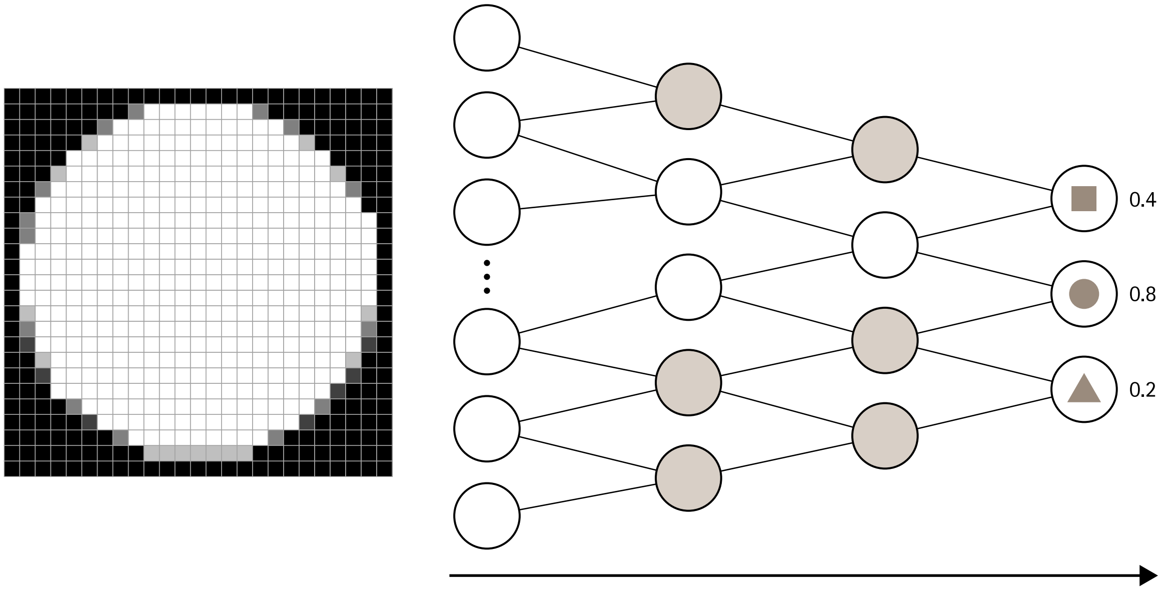
This appendix provides background information about artificial neural networks which form the base of deep learning.

10.1 Neural networks

Neural networks are a family of algorithms that are used to recognize relationships in datasets through a process that is somewhat similar to how the human brain works. A neural network consists of a hierarchy of multiple layers of so-called nodes or neurons which are interconnected, and information is being passed along the connections, from the input layer, through the network, to the output layer.

The assumption for neural networks to work is that an input data sample can be reduced to a finite set of features, creating a good representation of the input data. These features can then be combined and will help to classify the input data, for example, describing the contents of an image.

The illustration below shows an example where a neural network is used to identify which class the input image belongs to. Each pixel in the image is represented by one input node. All input nodes are coupled to the nodes in the first layer. These produce output values which are passed along as input values to the second layer, and so on. In each layer, weighting functions, bias values, and activation functions are also involved in the process.



Example of an input image (left) and a neural network (right). When the output layer is reached, the network has concluded probabilities for each possible category (square, circle, or triangle). The category with the highest probability value is the most likely shape of the input image.

This process is called *forward propagation*. In case of a mismatch of the result of the forward propagation the network parameters are slightly modified through *back propagation*. During this iterative training process the performance of the network is gradually improved.

After deployment, a neural network, in general, has no memory from previous forward passes. This means that it does not improve over time and that it can only detect the types of objects, or solve the types of tasks, it has been trained for.

10.2 Convolutional neural networks (CNN)

Convolutional neural networks (CNN) are a subtype of artificial neural networks that have proven to be especially suited for computer vision tasks and they are at the core of the rapid progress of deep learning. In the case of computer vision, the network is trained to automatically look for distinctive image features, similar to edges, corners, and color differences, in effect identifying object shapes across an image.

The main operation for accomplishing this is the mathematical operation called *convolution.* This is a very efficient operation since the output of each individual node is dependent only on a limited surrounding in the input data, which was produced by the previous layer, rather than using the entire input data volume. In other words: in a CNN, each node is not connected to every node in the previous layer but only to a small subset. The convolutions are complemented by other operations that reduce the size of the data while retaining the most useful information. As in a standard artificial neural network, the data becomes more and more abstract the deeper it travels into the network.

During the training phase, the CNN learns the best way to apply the layers. That is, how the convolutions should combine the features from the previous layer to have the output of the network agree as much as possible with the annotations of the training data. During inference, the trained convolutional neural network then sequentially applies the layers of convolutions that were the result of the training.

T10163986/EN/M3.2/2103

About Axis Communications

Axis enables a smarter and safer world by creating solutions for improving security and business performance. As a network technology company and industry leader, Axis offers solutions in video surveillance, access control, intercom, and audio systems. They are enhanced by intelligent analytics applications and supported by high-quality training.

Axis has around 4,000 dedicated employees in over 50 countries and collaborates with technology and system integration partners worldwide to deliver customer solutions. Axis was founded in 1984, and the headquarters are in Lund, Sweden

AXIS® COMMUNICATIONS