aps advanced protection systems

a game changer in drone detection and mitigation

ctrl+sky

about us

Advanced Protection Systems is a technology company that has developed and commercialised a unique system for identifying and neutralising drones: **Ctrl+Sky.**

The system is based entirely on the original patented solutions developed by APS team - outstanding scientists and engineers. The company has headquarters in Poland and the United States, however, the network of distributors includes 32 countries around the world.

Advanced Protection Systems with Ctrl+Sky, is one of the key players in the global drone detection and identification market, according to the Markets and Markets ranking.

the challenge

Over the last few years, there have been regular media reports regarding troublesome and even dangerous incidents involving drones which are able to spy, steal confidential information and violate our privacy. In 2016 alone there were as many as 1,200 incidents involving drones within the European Union that came dangerously close to commercial planes.

By 2025, more than 400 thousand professional drones will fly in the European Union’s airspace.

This is the reason why the demand for anti-drone systems is growing rapidly. However, their market availability is still relatively small. It is possible to purchase extremely expensive, dedicated, military-grade solutions or simpler yet not always effective systems aimed at commercial market.

The challenge for Advanced Protection Systems is to provide a product that can ensure the highest standard of security at a price-point that is accessible for commercial and private clients alike.

competitive advantages

Ctrl+Sky has many competitive advantages. From a fully reconfigurable system, to a unique set of sensors, finishing with an excellent price to quality ratio. The combination of practical and scientific approaches as well as the focus on maximising efficiency makes Ctrl+Sky a global game changer in its category.

Ctrl+Sky is the only drone detection system in the world featuring:

- A modular and fully reconfigurable radar sensor
- MIMO radar technology for improved performance
- Radar tracking based on MHT (multi hypothesis tracking) algorithm
- An acoustic sensor with direction-finding capability

clients

It is estimated that the potential target market for drone detection and neutralisation systems is about 300,000 entities and that number is growing rapidly from year to year. The area of application is very wide:

- **protection of private property**
- **security at public events**
- **government building security**
- **stadium security**
- **airport security**
- **protection of strategic infrastructure**
- **prison protection**
- **u-space monitoring**

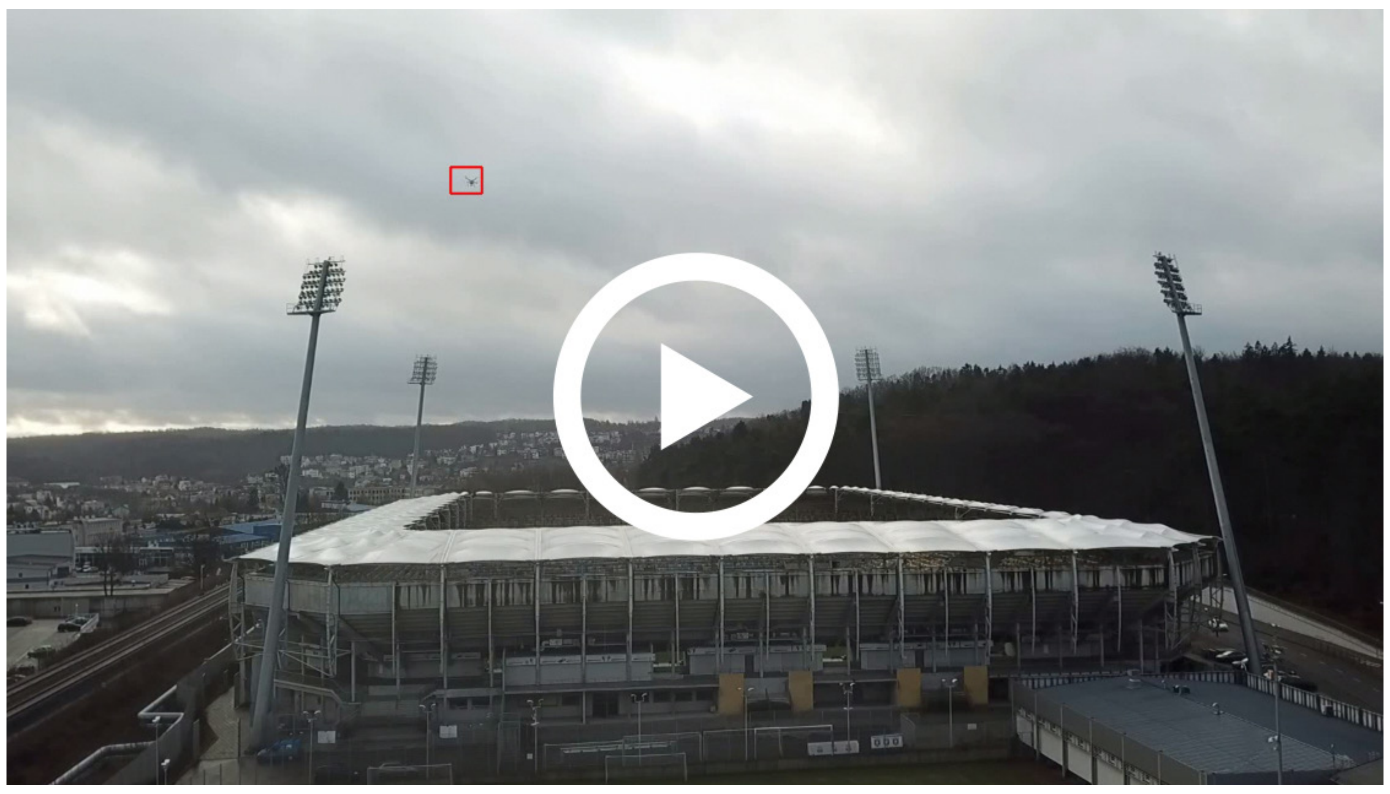
Stationary

Mobile

Ctrl+Sky: multi-sensor drone detection, classification and tracking

Ctrl+Sky System is designed to deliver real-time, AI-enabled drone detection information from multiple sensors. To achieve this system is divided and all signal processing is distributed. The system consists of multiple nodes that are designed to communicate with each other. Like all distributed solutions this one has a single node bounding all the other ones together. Every channel between nodes has a very specific, very carefully prepared communication protocol.

Ctrl+Sky System Architecture



The main module is the REST Server Application with Web Based Front End Application. Server is the core of the system and its responsibility is to gather all information from all other nodes. It is highly scalable and designed to communicate with any number of client modules. Server applications are designed to be deployed as micro services to allow even better scalability. It is critical for the system to be able to process as much data as possible, not only to present it on the frontend application, but also to perform multiple input and parallel data fusion. Behind REST Service all data is processed through the Fusion Service which is responsible for multi sensor target classification and tracking. With more data there is higher probability to detect any target. Detection from one node provides additional true data for all other sensors, which is later on processed into AI model.

Before any communication each module is required to authorize into the system. Every node is capable of pushing new sensor data into the API. Each frame contains compressed and processed data from sensor. Behind each module there are always at least two applications. Endpoint application is always communicating with the signal providing hardware. Whether it is a radar box, microphone array, or any other device, endpoint application is always responsible for reading and data processing. For each sensor there are number of algorithms that are being used to process all the data. If detection occurs, the output of the application will be all the information about the target. Depending on the signal source, it can be wifi signal, electromagnetic signal (radar), noise (microphone), RF signal (spectrum tool), or image (camera). The detection is than sent to the API Client, where it is again processed and sent to server. For different applications there are many occasions when REST communication is not enough to provide real time data to the user. In these situations, there is always socket link communication between the node and the server.
Every node is responsible for its own processing inside. Each sensor has different type of computational needs, so different sensors use different processing units. The most powerful board required is **Jetson TX2**, which is used for image processing, classification and target tracking.

Some sensors are aware of each other and are able to communicate. Camera is one of the modules that is situated on the PTZ mechanism, which allows it to Pan, Tilt and Zoom on an object. In most cases cameras are mounted on top of the mast, while radars are mounted on the side. This solution allows us to rotate the camera to any direction required. If (for example) detection occurs on the South West direction (azimuth 225°) camera can rotate on that direction. If distance and/or elevation of the target is known than camera can more precisely aim by tilting and zooming on a given object. The whole process is fully automated. When cameras are available on the site than their responsibility is to track radar targets and provide more information from processed image to the whole system.

The last module of our application is the Frontend Web application. It is user friendly control panel to the whole system. The main idea is to provide as much information about the targets in the least overwhelming way. On the control panel there are several shapes drawn on the map.

Artificial Intelligence in Ctrl+Sky

To handle image processing in our system we use Nvidia Jetson TX2, which gives us real-time AI performance. Each module has 256 CUDA cores, which are all used for deep learning, computer vision, GPU computing making it ideal for our needs. Jetson runs Linux Ubuntu with dedicated JetPack 3.2. Our software runs OpenCV 3.4 with GPU accelerated vision processing, which has a great performance with new CUDA 9.0 Toolkit. We use cuDNN (v. 7.0.5) for even faster image classification. Classification is done by Keras - high-level neural networks API. Keras is capable of running on top of TensorFlow library (numerical computation library) which is even faster with new high-performance deep learning inference optimizer and runtime - Nvidia TensorRT. Basically, we are using TensorRT to deliver real-time processing and streaming to the user.

The whole system uses three types of neural network algorithms. Microphone array uses AI to classify targets noise. Image from the camera is processed and uses artificial intelligence to classify objects and depending on the objects surrounding it uses two tracking algorithms. We use KCF algorithm for general purpose tracking, TLD tracker under occlusion (tracked objects are being covered by other objects - crossing objects), and in some cases we use GOTURN tracker. Entire AI processing in the vision sensor (camera) is performed using NVIDIA Jetson TX2. This allows us to leverage the powerful supercomputer capabilities of Jetson TX2, at a relatively low cost and low power consumption – important factors in our commercial application.

As it was said before – on detection triggered by any sensor there is a signal generated and sent to the camera. It gives information about the approximate position of the object. We say object, because it may not be drone. To be certain about the detection it's best to have more than one sensor confirming detection. The coordinates generated by the radar are used to control the camera. Knowing the approximate position of the object we point the camera to have optimal view on it. For better control, we use Onvif protocol, which is supported by most of the cameras. Our system uses AXIS Q8685-E PTZ camera. It provides a thirty-fold zoom and panoramic image and allows quick but smooth control.

The image is obtained by the GStreamer application. One of the benefits of using Gstreamer is it's architecture. It allows using any number of processing steps between acquiring the video and using it in our application. It is designed as a pipeline application, which means that we build our processing line from elements, that are passing the image from one to another as a stream.

After image is obtained it is processed by the OpenCV algorithms. First step is to apply mog to the image, and then we find contours of all objects. We sort the contours to first have acceptable size. If the object is to small, there is very little chance that it will be classified as drone, so it is excluded from the search. It may reject some drones that are very far away and may appear as dots on the image, but very little objects have a very small chance to be classified, and with many small objects there is a lot of processing power required to process them. So, rejecting some of the small object is a way to optimize, or rather minimize number of objects that have a very small chance to be classified anyway.

This solution gives us information about all objects that are on the image at all time. To make it better we don't always use all the image for classification, but only one frame every few seconds. Rather than processing whole image, we only find moving objects. We do this by finding differences between two frames, and for this we use background subtraction. After that, we have parts of image that are sent to the processing.

Detection algorithm works in two states; detecting and tracking. After confirmed detection, algorithm switches to the tracking state. At this state algorithm has one or more targets to track. Depending on the number of moving objects and their paths we use different tracking algorithms. If moving targets intersect, we switch to TLD tracker, which quickly recovers after the track is lost. Second tracker used is GOTURN tracker, which is using regression networks. This tracker is used whenever there are more than two targets located at a very close range and their area intersect and/or one is covering the other one. Great advantage of using this tracker is that it allows us to reduce number of tracked objects by merging image areas into one track. This also helps join many detections triggered by cloud movement and then later reject them. For general purpose we use KLD, which is faster and more reliable.

For classification we use Keras Neural Network. The model consists of densely-connected Neural Network layers with relu activation function, categorical cross entropy as loss function and SGD optimizer. The model is created from the background of the site. Depending on the site the background will be different, so for each site we use special application to gather images of the site, and later on there is an automated process of learning the model. For each learning process we use as much drone images as possible, to have the model perfectly balanced.

paweł recław
lead software engineer

Pawel is a programmer proficient in many languages (including C, C ++, Java, C #). He has worked on the development of distributed applications, created networks and administered IT equipment and systems.

katarzyna studzińska
al signal processing engineer

Kasia has received the BSc degree in Robotics from Gdansk University of Technology. Currently she's pursuing MSc degree in Computer Science. She is specialising in deep neural networks, applied mainly to image recognition and classification. Kasi previously has worked at Insaldo on mobile applications as an Android Developer. In APS Kasi is developing AI-based solutions for drone detection and classification.

radosław piesiewicz
ceo & co-founder

Doctor of Communication Sciences at the Technical University of Braunschweig (Germany). During his career, he became head of broadband and wireless networks at Create-Net in Italy. In 2009 he moved to Wrocław, where he managed large innovation projects, taking up the post of Director of the Center for Research and Development of ICT in Wroclaw's EIT+ Center. At APS he is responsible for coordinating operational processes, business development and sales.

maciej klemm
co-founder

Senior Lecturer at the Department of Electrical and Electronics Engineering at the University of Bristol. In 2003 he joined the Electronics Laboratory of ETH Zurich, Switzerland, where after three years he obtained a doctorate in Electronic Science. In 2006 he joined the University of Bristol in the United Kingdom. In 2007 he was awarded the Best British Research Engineer Award. At APS he is responsible for coordinating the engineering team and the development of new products.

for more information visit **www.apsystems.tech**