WHITEPAPER

Vision and AI for Public Safety Applications

Introduction and Design Considerations

Arcturus
We are Edge.

Table of Contents

Introduction 1

Image Acquisition Methods and Camera Selection 1

Algorithmic Processing 1

Motion and Intrusion Detection 2

Anomalous Event Detection 2

Inference and Object Detection 3

Object Tracking 4

Facial Recognition 4

Arcturus Analytics 5

1. Motion / Intrusion 5
2. Boundary Crossing / Zone Incursion 5
3. Abandoned Package 5
4. Loitering and Motionless 6
5. Traffic and Crowd Analysis 6
6. Face Verification 6
7. Removed Object 6
8. Semantic Characterization 7
9. Tracking and Re-Identification 7

Interacting with the System 7

Summary 7

How to Engage with Arcturus 7

Additional Material 8

Introduction

As our cities get smarter, the combination of connectivity and intelligence will propel the advancement of the ways we interact with each other and our surroundings. No place is this clearer than in public safety. Our urban centres are being challenged to meet the new demands of physical distancing, crowd and capacity management, in addition to the traditional physical security needed to move people and commerce safely. Vision and AI systems will play a vital role in this paradigm. It will enable intelligent transportation systems to detect and adapt to ridership; enhance building access control systems to deliver capacity management and support innovative new healthcare applications that will bridge society's social needs with the health and welfare of our population.

This whitepaper will introduce fundamental vision processing methods and principles as they apply to public safety and security. It will address classic vision processing algorithms, AI models, detection methods and processing requirements along with common pitfalls not considered when developing vision and AI solutions at edge.

Image Acquisition Methods and Camera Selection

When light enters a camera lens, it is filtered into its red, green and blue (RGB) components, each of which hit an image sensor separately. The digital signal is processed by an Image Signal Processor (ISP) responsible for image enhancement, removing defective pixels, de-bayering the RGB mosaic, colour correction, gamma, filtering noise, white balance, exposure and black level adjustment. Once processing is complete, the signal is then serialized and transmitted to a host processor using a camera serial interface such as MIPI CSI (Camera Serial Interface).

Not all vision applications acquire an image directly from a sensor. In some cases, it is desirable to maintain existing camera equipment, use low-cost commodity camera hardware, or extend the distance between the camera and the resources performing the vision processing. In this case acquiring the image source from a network stream can be useful. Efficient transmission of video across a network requires encoding, typically using H.264 or H.265, VP8 or VP9. Each format gains efficiency by relying on inter-frame dependency-based compression. Simply put, each video frame is divided into segments and subsequent frames contain only the delta for the segments that change in the image. Periodically an I-frame is sent that contains the complete image data, thus rectifying any image corruption due to packet loss or error. This is an over-simplification as complex and predictive methods are used to determine this compression along with sub-streaming and other tricks to maintain quality of experience.

A third method of image acquisition is hybrid or accelerated cameras. These are most commonly found in consumer webcams or in specialized industrial vision applications. These cameras are front-end systems that contain a sensor, ISP and a processor capable of transmitting the camera data over a USB interface. Industrial applications may prefer to have the RAW (e.g.: YUY2) image while consumer webcams may encode data into mJPEG or H.264 to off-load this processing from the main PC CPU. For broadest compatibility, USB devices that follow the UVC (USB Video Class) generic video driver are natively supported under V4L2 (Video for Linux2) framework. Using this type of hybrid camera can be a convenient method to attach a camera to an existing system that may not have encoding/decoding hardware or may require specialized capabilities such as LWIR (Long-Wave Infrared).

Algorithmic Processing

Algorithmic processing is the classic computer vision approach. We generally think that AI inference obsoletes this approach – but vision algorithms serve a significant complementary purpose. Algorithms are generally less computationally expensive, meaning

that they can run on every frame or even every pixel. In addition, algorithms are generally good candidates for hardware acceleration using DSPs, floating-point and SIMD units such as NEON®, included in Arm® cores.

The first stage of algorithmic processing is the application of low-level primitives for pixel manipulations and feature extraction. This type of processing is typically performed on each pixel within the image. The objective is to look for areas of interest such as a group of pixels with similar traits – referred to as blob detection. Feature extractions increase in complexity to include edge detection (Canny or Sobel), corner detection using Harris or FAST and algorithms such as Hough transformations that bridge the gap between feature extraction and object detection by adding geometric shape recognition. SIFT and SURF can be used to recognize objects based on relative key point comparisons, while the HOG algorithm applies the theory that more complex objects can be described better by using gradients. This makes HOG a particularly powerful tool for detecting shapes with regular contours –such as human faces. As a face detector, HOG can pass the co-ordinates of the detected face to a more powerful inference network to create the facial landmark feature embedding. Thus, the network generating the embedding is run only as required.

Motion and Intrusion Detection

One of the basic primitives in video analytics is the ability to detect motion and change. This requires a method to understand frame-over-frame differences, such as Background Subtraction (BGS). This technique isolates the foreground image through a process of subtracting a background image. The output then provides key motion detection and location information, useful for higher-level application processing.

Simple BGS can be implemented by maintaining a median value of each pixel, updated frame over frame, differencing the pixels and dropping the ones that fall below a threshold limit (effectively a form of low-pass filter). While this method is simple and fast, it is not very robust.

More modern approaches use Gaussian Mixture Models (GMM) where a mixture of Gaussians form the underlying distribution of the pixel colour. With each subsequent frame, the mixture is updated, if the distance of the new pixels exceeds a predetermined limit in the RGB spectrum, then they are determined to be foreground. This approach is the most commonly used with the MOG2 implementation, publicly available in OpenCV. This approach is referred to as a parametric approach as it makes statistical assumptions about the distribution being analyzed. This method is effective, but has limitations when backgrounds becomes less visible than the foreground or if the background experiences high variance.

Arcturus utilizes a non-parametric background subtraction implementation that improves on the pixel-to-pixel comparison, by considering neighbouring pixels, thus a spatiotemporal approach. This level of precision comes with a performance cost, which is off-set by using a more efficient update and classify approach and light-weight, consensus-based post-processing. The result is faster performance with a higher tolerance to shadow intensities, small motion, noise, undulating objects and varying light conditions.

Regardless of the implementation, using algorithmic methods alone to detect motion/intrusion may suffer from high false positive rates. To overcome this, object detection can be applied to help filter unwanted events by classifying the object –for example a bird flying through the field of view can be selectively ignored as a motion/intrusion event.

Anomalous Event Detection

Anomalous event detection has a broad range of applications from detecting unusual vibration in equipment to finding arrhythmia in heart rate data. In public safety, anomalous event detection bridges the

gap between what we know is normal and what is not. In other words, detecting what is abnormal, requires that we first define what is normal, then the outliers can be identified using the following methods;

- A Covariance Matrix models the optical flow of movement in a scene and works best to detect outliers where the trajectory of the anomaly differs from that of the common optical flow, such as an object crossing a path.
- Motion History Intensity is similar to a co-variance matrix but represents the optical flow by temporal accumulation of motion in the image. This offers methods to establish regions where different levels of motion are expected.
- Interaction Energy Potentials use tracklets to determine the velocity of objects, thus a sudden change in direction or speed could be considered an anomaly, such as a reaction to a sudden stimulus.
- A Bag of Words (BoW) approach divides the images into patches and instead of using a histogram to describe each patch, it uses words. Analysis is done by looking at the distribution of words and word clusters in the image, in a similar way that textures are analyzed.

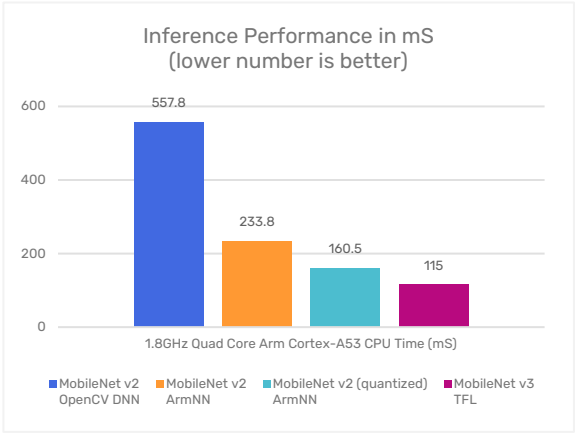
In the previous section, we described how a pipeline of algorithms and object detection could mitigate false positive motion detection events by classifying and selectively ignoring a bird. Using this approach has some interesting unintended consequences. Object detection, by its nature, is invariant to scale and therefore does not care what size the bird in the image should be. This means that any object that looks sufficiently bird-like will fool the object detector – even a person in a bird costume. In this case we can choose to use anomalous event detection or other higher-level logic to identify this type of outlier.

Inference and Object Detection

Convolutional Neural Networks (CNNs) represent the single most significant advancement in vision processing. There are many different types of networks used for specialized tasks, but object detection and classification is the most common and general purpose.

Object detection relies on a network that has been pretrained on a labelled dataset containing many object images using different scale, angle, resolution, background etc. These datasets such as COCO (Common Object in Context) or ImageNet are publicly available. This training takes place outside of the edge device. The network weights obtained during the training process are then applied to the network running at the edge.

A number of state-of-the-art, pre-trained object detection networks are available including Yolo or Google's MobileNet. MobileNet's architecture is implemented to reduce the number of floating-point operations, making it a good choice for edge applications. MobileNet with SSD (Single Shot Detector) forms an efficient combination of detection and classification. SSD uses a regression-based approach to object detection starting with a predefined grid of multiple boxes. This approach differs from traditional object detectors that propose many regions and then go back to analyze each of them. Instead, MobileNet_SSD improves efficiency by being able to perform detection and classification in one forward pass of the network. Model quantization further reduces the floating-point operations and when combined with optimized edge runtime frameworks such as ArmNN and TensorFlow Lite, it becomes possible to achieve real-time object detection using general purpose processors, without any specialized hardware acceleration. The following chart illustrates this performance across several generations of networks and runtimes using an NXP® i.MX 8M Mini (4 x Arm Cortex-A53 @ 1.8GHz) CPU.



Notes

- 6 images, each resized to 300 x 300
- OpenCV and ArmNN, both using 4 threads
- Model is ssd_mobilenet_v2; OpenCV loads Tensorflow .pb/.pbtxt while ArmNN uses .tflite (quantized and non-quantized)
- Pretrained on MS-COCO from Tensorflow Model Zoo
- Model is ssd_mobilenet_v3_small, non-quantized using .tflite

Object Tracking

Tracking forms a frame-over-frame relationship between objects. This relationship allows velocity to be determined and predictions made about the future. Consider tracking a person walking down a corridor; while it is possible to re-detect them as an object in every frame, this provides only limited information about their localization. Tracking establishes a unique identity which can be maintain, even if the object is temporarily occluded, i.e. a person walking behind a pillar. Similarly, in scenes that contain multiple moving objects, it is possible to track each one distinctly without confusing them with each other. Since motion-model tracking makes predictions about where an object is going to be, it inherently does not need to be run on every frame. This makes it less computationally intensive than approaches that rely on redetection.

For applications that require the concurrent tracking of many objects in real-time, a highly efficient method is needed. Typically, this involves using a Kalman Filter to predict the future location of the tracked object based on its velocity and probability. This method is combined with the Hungarian algorithm to match the object in the subsequent frames. This combination of motion model tracking and identity assignment is referred to as Simple Online Realtime Tracking (SORT). Deep SORT augments this by using a deep association metric – a visual appearance embedding generated by an embedding network. This approach benefits by being able to use the Euclidean distance between visual appearance embeddings as a method of increasing the confidence when making the re-identification assignment match.

Facial Recognition

Facial recognition is fast, robust and has broad application in tracking, identification, reidentification, and authentication; however, it is also at the forefront of privacy concerns with countries such as Belgium outlawing it wholesale for private use.

Face recognition requires a pipeline that detects faces, compensates for angle and rotation, detects facial landmarks and compares these features to a known set. State-of-the-art methods implement a HOG (described earlier) based face detector as a method to quickly identify multiple faces from within a video frame. This method is coupled with an image pyramid technique to improve facial detection, even if faces are small. The output from the face detector is sent to a shape predictor that uses 5 facial landmarks to align the face to a standard pose. This aligned image is then processed by an embedding network that generates a 128D feature vector of the facial landmarks. Euclidean distance can then be used to compare these landmarks against a database of known embeddings. The closer the distance, the more likely the faces are a match.

While 2D facial recognition is not as robust as 3D, it is cheaper, faster, requires no specialized hardware and works well under comparatively non-ideal conditions. This means that 3D face recognition is suitable as a

strong method to authenticate a cooperative user, while 2D is useful to perform 1-to-1 comparisons to quickly identify a subject. With the addition of liveness detection using blinks, micro-movements and/or changes in gradients, improved protection against spoofing can be achieved using 2D.

Arcturus Analytics

Through the combination of methods discussed in this paper, Arcturus delivers a comprehensive set of analytics for public safety and security. These analytics are built on top of a highly extensible vision pipeline framework that supports discrete nodes for video source, inference, embeddings, analytics, reidentification and video sink. The pipeline architecture employs cloud native design methodologies allowing nodes to be deployed and interconnected across different physical resources. This approach allows AI processing to span from fully on-premises to cloud seamlessly, irrespective of resources. This mitigates the need to over-build hardware at the edge without constraining the scalability of the AI application. Arcturus supports edge optimized runtimes including TensorRT, ArmNN and TensorFlow Lite. Arcturus analytics include:

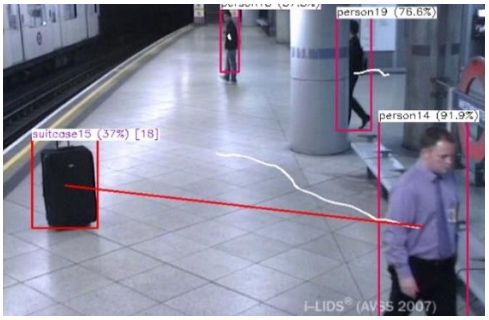
1. Motion / Intrusion

- Detect appearance during a restricted time
- Detect vehicles where only people should be
- Detect people where only vehicles should be
- Detect tailgating (two entrances for one pass)

2. Boundary Crossing / Zone Incursion

- Define zones or boundaries
- Detect and track people or vehicles
- Detect object crossing zone or boundary
- Real-time tally of zone occupancy
- Real-time tally of line cross entrances and exits
- Estimate occupancy/capacity

3. Abandoned Package

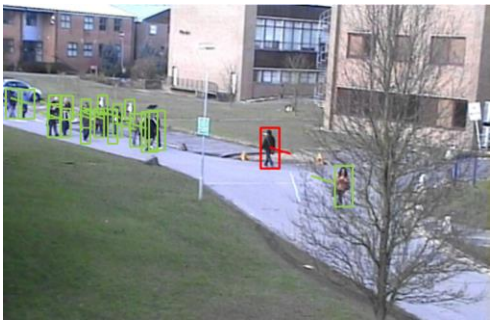


- Detect packages, backpack, bags or suitcase
- Detect people entering with or without package
- Detect people exiting with or without package
- Detect association of package with person to form ownership
- Measure proximity of owner to package
- Detect package abandonment

4. Loitering and Motionless

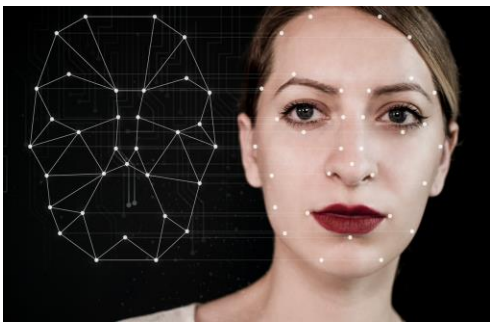
- Detect and track people
- Determine time each person in field of view
- Detect if person becomes motionless
- Detect if tracking pattern is consistent with pacing or circling
- Reidentify people that have left the field of view and returned

5. Traffic and Crowd Analysis



- Detect direction of travel of people or vehicles
- Define zones or boundaries
- Detect flow rate across a zone or boundary
- Detect counterflow / wrong-way access
- Detect sudden traffic increase or decrease
- Approximate capacity
- Provide real-time intelligence metrics and bottleneck detection

6. Face Verification



- 2D face detection
- HOG based fast facial landmarks
- Facial standard pose alignment
- 128D feature vector embedding distance
- Euclidian distance measured against known image database
- Suitable for two-factor authentication and general-purpose identification

7. Removed Object

- Secure zone and/or secure object
- Intrusion into secure zone
- Direction of travel of secure object from secure zone into insecure zone
- Suitable for package theft
- Suitable for high-value displays

8. Semantic Characterization

- Light-weight pose estimation
- Body model segmentation
- 18 classes of attributes from hats to shoes
- Natural language interface
- Suitable for active watch lists and archival searches

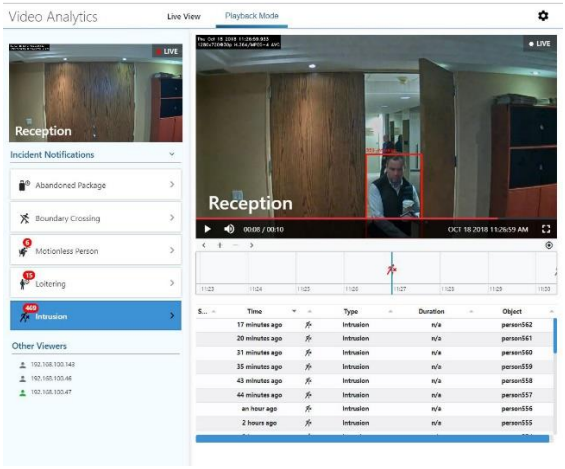
9. Tracking and Re-Identification

- Target localization and trajectory prediction
- Visual appearance embedding
- Deep association metric for multi-camera environments
- Image used as probe
- Suitable for active watch lists and archival searches

Interacting with the System

Arcturus provides a UI/UX that has been developed using a Design Thinking process, in consultation with transit authority, police and security professionals. The interface presents real-time event notifications and uses intuitive workflows to switch between notifications, live video and event playback seamlessly. A timeline and clustering makes searching events across time domains simple.

The system uses a time series database (TSDB) backend, optimized for fast, high-availability storage using standard API queries. This simplifies the customization of visualization and dashboards through the use of off-the-shelf tooling. Live streaming is handled using webRTC and RTSP for playback with scrubbing controls.



Summary

In this paper we have discussed methods of video acquisition and image pre-processing as well as the application of algorithms, object detection and the benefits of a pipeline that combines both. We have addressed the importance of object tracking and edge implementations, along with an overview of 2D and 3D facial recognition. These building blocks make for a powerful toolkit from which edge-based public safety and security applications are developed.

Arcturus is continuously improving and expanding its capabilities and is at the fore-front of research in areas including reidentification, semantic characterization and behaviour detection.

How to Engage with Arcturus

Arcturus offers a range of engagements, depending on the project needs and type of collaboration.

- Low barrier-to-entry engagement packages provide engineer-level access and define

short-term objectives. These programs are ideal for projects that need to demonstrate viability without heavy investment.

- Full-stack system solutions are available for projects that require turn-key development with access to deep expertise and a comprehensive portfolio of IP.
- A developer program is available for company's that want to build internal AI capacity but need a head start with expert tooling and support.

Whether you choose to do it yourself, or work directly with us, Arcturus accelerates the delivery of world class AI solutions at the edge. Contact us directly for more information dsteele@arcturusnetworks.com.

Additional Material

- Bring Edge AI and Vision Analytics product landing page

Arcturus Networks Inc.

701 Evans Ave. – Suite 300
Toronto, ON
M9C 1A3
CANADA

Toll Free North America: 1.866.733.8647
Tel: +1 416.621.0125

https://ArcturusNetworks.com

arcsales@arcturusnetworks.com

Arcturus
We are Edge.