Abstract

Objective:

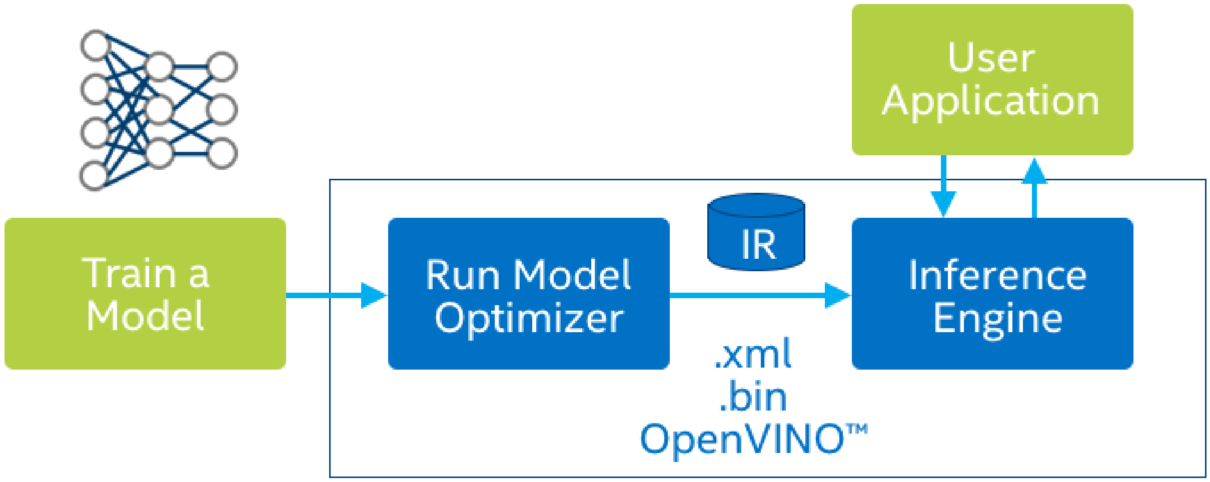
Robberies, burglaries and thefts continue to plague the length and breadth of this country. As per the National Crime Records Bureau (NCRB), 2,44,119 cases of robbery, theft, burglary, dacoity, among others, took place in residential premises in 2017. This was a jump of over 10% from 2016 when the number of such cases stood at 2,20,854. The financial loss due to these thefts and burglaries are staggering. In 2017, value of property stolen from residential premises was in excess of Rs. 2065 crores, a 40% jump from Rs. 1,475 crores stolen the previous year. Indeed, home safety is a serious concern for the residents as well as the police and the authorities. These figures do not take into account the rest of thefts that take place every year inside industrial complexes.

With the advent of CCTV's for surveillance we can now use the video feed to survey and protect our resources. But a CCTV is just a tool and requires constant monitoring by a human element. With large industrial complexes the number of cameras far exceed the capabilities of the human element to effectively process all the video feed he receives. Thus we propose a smart Intruder detection system capable of using computer vision and AI capabilities to detect and track the intruder if any.

Method:

We use a Person/Vehicle/Bike detector is based on SSD detection architecture, RMNet backbone, and learnable image downscale block (like person-vehicle-bike-detection-crossroad-0066, but with extra pooling). The model is intended for security surveillance applications and works in a variety of scenes and weather/lighting conditions.

The model was initially trained and developed with Caffe framework. We then use Intel's OpenVino Toolkit for deploying the model at the edge and performing inference on it. The OpenVINO™ toolkit is a comprehensive toolkit that you can use to develop and deploy vision-oriented solutions on Intel® platforms. Vision-oriented means the solutions use images or videos to perform specific tasks. A few of the solutions use cases include autonomous navigation, digital surveillance cameras, robotics, and mixed-reality headsets. The basic workflow of OpenVino is as follows:



Our model has the following inputs and outputs:

Inputs

1. name: "input" , shape: [1x3x1024x1024] - An input image in the format [BxCxHxW], where:
   - B - batch size
   - C - number of channels
   - H - image height
   - W - image width

   The expected color order is BGR.

Outputs

1. The net outputs a blob with shape: [1, 1, N, 7], where N is the number of detected bounding boxes. For each detection, the description has the format: [`image_id`, `label`, `conf`, `x_min`, `y_min`, `x_max`, `y_max`], where:
   - `image_id` - ID of the image in the batch
   - `label` - predicted class ID
   - `conf` - confidence for the predicted class
   - (`x_min`, `y_min`) - coordinates of the top left bounding box corner
   - (`x_max`, `y_max`) - coordinates of the bottom right bounding box corner

We use the OpenCV library for video input and then process the video frame by frame for inference and showing the output.