Journal of Signal and Information Processing, 2019, 10, 73-95
http://www.scirp.org/journal/jsip
ISSN Online: 2159-4481
ISSN Print: 2159-4465

Target Tracking and Classification Using Compressive Measurements of MWIR and LWIR Coded Aperture Cameras

Chiman Kwan[1*], Bryan Chou[1], Jonathan Yang[2], Akshay Rangamani[3], Trac Tran[3], Jack Zhang[4], Ralph Etienne-Cummings[3]

[1]Applied Research LLC, Rockville, Maryland, USA
[2]Google, Inc., Mountain View, California, USA
[3]Department of Electrical and Computer Engineering, the Johns Hopkins University, Baltimore, USA
[4]Department of Electrical Engineering, Massachusetts Institute of Technology, Cambridge, Massachusetts, USA
Email: *chiman.kwan@signalpro.net

How to cite this paper: Kwan, C., Chou, B., Yang, J., Rangamani, A., Tran, T., Zhang, J. and Etienne-Cummings, R. (2019) Target Tracking and Classification Using Compressive Measurements of MWIR and LWIR Coded Aperture Cameras. *Journal of Signal and Information Processing*, **10**, 73-95.
https://doi.org/10.4236/jsip.2019.103006

Received: July 4, 2019
Accepted: August 5, 2019
Published: August 8, 2019

Abstract

Pixel-wise Code Exposure (PCE) camera is one type of compressive sensing camera that has low power consumption and high compression ratio. Moreover, a PCE camera can control individual pixel exposure time that can enable high dynamic range. Conventional approaches of using PCE camera involve a time consuming and lossy process to reconstruct the original frames and then use those frames for target tracking and classification. In this paper, we present a deep learning approach that directly performs target tracking and classification in the compressive measurement domain without any frame reconstruction. Our approach has two parts: tracking and classification. The tracking has been done using YOLO (You Only Look Once) and the classification is achieved using Residual Network (ResNet). Extensive experiments using mid-wave infrared (MWIR) and long-wave infrared (LWIR) videos demonstrated the efficacy of our proposed approach.

Keywords

Target Tracking, Classification, Compressive Sensing, MWIR, LWIR, YOLO, ResNet, Infrared Videos

1. Introduction

There are many applications such as traffic monitoring, surveillance, and security monitoring that use optical and infrared videos [1]-[5]. Object features in optical and infrared videos can be clearly seen as compared to radar-based

trackers [6] [7].

Compressive measurements [8] [9] [10] [11] can save data storage and transmission costs. They are normally collected by multiplying the original vectorized image with a Gaussian random matrix. Each measurement contains a scalar value and the measurement is repeated M times where M is much fewer than N (the number of pixels). To track a target using compressive measurements, it is normally done by reconstructing the image scene and then conventional trackers are then applied.

Tracking and classification of targets in compressive measurement domain is difficult because target location, size, and shape information are destroyed by the Gaussian measurement matrix. Conventional approaches do not work well without image reconstruction.

Recently, a new compressive sensing device known as Pixel-wise Code Exposure (PCE) camera was proposed [12]. A hardware prototype was developed and performance was proven. In [12], the original frames were reconstructed using L_0 [13] [14] [15] or L_1 [16] sparsity-based algorithms. One problem with the reconstruction-based approach is that it is extremely time consuming to reconstruct the original frames and hence this may prohibit real-time applications. Moreover, information may be lost in the reconstruction process [17]. For target tracking and classification applications, it will be ideal if one can carry out target tracking and classification directly in the compressive measurement domain. Although there are some tracking papers [18] in the literature that appear to be using compressive measurements, they are actually still using the original video frames for tracking.

In our earlier paper [19], we presented a deep learning approach that directly incorporates the PCE measurements. In that work, we focused only on short-wave infrared (SWIR) videos. It is well-known that there are several key differences between SWIR, MWIR, and LWIR videos. First, SWIR cameras require external illuminations whereas MWIR and LWIR do not need external illumination sources because MWIR and LWIR are sensitive to heat radiation from objects. Second, the image characteristics are very different. Target shadows can affect the target detection performance in SWIR videos. However, there are no shadows in MWIR and LWIR videos. Third, atmospheric obscurants cause much less scattering in the MWIR and LWIR bands than in the SWIR band. Consequently, MWIR and LWIR cameras are tolerant of smoke, dust and fog.

Because of the different characteristics in SWIR, MWIR, and LWIR videos, it is necessary to study the performance of the previously proposed deep learning approach [19] to MWIR and LWIR videos. In this paper, we propose a target tracking and classification approach in compressive measurement domain for MWIR and LWIR images. First, a YOLO detector [20] is used for target tracking. This is called tracking by detection. The training of YOLO tracker is very simple, which requires image frames with known target locations. Although YOLO can also perform classification, the performance is not good as we have a

very limited number of video frames for training. As a result, in the second step of target classification, we decided to use ResNet [21] for classification. We chose ResNet because it allows us to perform customized training by augmenting the data from the limited video frames. Our proposed approach was demonstrated using MWIR and LWIR videos with about 3000 frames in each video. The tracking and classification results are reasonable. This is a big improvement over conventional trackers [22] [23], which do not work well in the compressive measurement domain.

This paper is organized as follows. In Section 2, we describe some background materials, including the PCE camera, YOLO, ResNet, video data, and performance metrics. In Section 3, we summarize the tracking and classification results using MWIR and LWIR videos. Finally, we conclude our paper with some remarks for future research.

2. Background and Technical Approach

2.1. PCE Imaging and Coded Aperture

In this paper, we employ a sensing scheme based on PCE or also known as Coded Aperture (CA) video frames as described in [12]. **Figure 1** illustrates the differences between a conventional video sensing scheme and PCE, where random spatial pixel activation is combined with fixed temporal exposure duration. First, conventional cameras capture frames at certain frame rates such as 30 frames per second. In contrast, PCE camera captures a compressed frame called motion coded image over a fixed period of time (Tv). For example, a user can compress 30 conventional frames into a single motion coded frame. This will yield significant data compression ratio. Second, the PCE camera allows a user to use different exposure times for different pixel locations. For low lighting regions, more exposure times can be used and for strong light areas, short exposure can be exerted. This will allow high dynamic range. Moreover, power can also be saved via low sampling rate in the data acquisition process. As shown in **Figure 1**, one conventional approach to using the motion coded images is to apply sparse reconstruction to reconstruct the original frames and this process may be very time consuming.

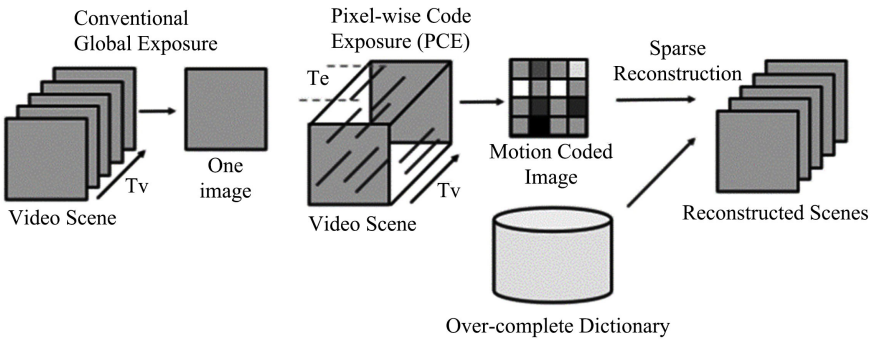


Figure 1. Conventional camera vs. Pixel-wise Coded Exposure (PCE) Compressed Image/Video Sensor [12].

Suppose the video scene is contained in a data cube $\boldsymbol{X} \in \boldsymbol{R}^{M \times N \times T}$ where $M \times N$ is the image size and T is the number of frames. A sensing data cube is defined by $\boldsymbol{S} \in \boldsymbol{R}^{M \times N \times T}$ which contains the exposure times for pixel located at (m, n, t). The value of $\boldsymbol{S}(m,n,t)$ is 1 for frames $t \in [t_{\text{start}}, t_{\text{end}}]$ and 0 otherwise. [t_{start}, t_{end}] denotes the start and end frame numbers for a particular pixel.

The measured coded aperture image $\boldsymbol{Y} \in \boldsymbol{R}^{M \times N}$ is obtained by

$$\boldsymbol{Y}(m,n) = \sum_{t=1}^{T} \boldsymbol{S}(m,n,t) \cdot \boldsymbol{X}(m,n,t) \tag{1}$$

The original video scene $\boldsymbol{X} \in \boldsymbol{R}^{M \times N \times T}$ can be reconstructed via sparsity methods (L_1 or L_0). Details can be found in [12].

Instead of doing sparse reconstruction on PCE images or frames, our scheme directly acts on the PCE or Coded Aperture Images, which contain raw sensing measurements without the need for any reconstruction effort. Utilizing raw measurements has several challenges. First, moving targets may be smeared if the exposure times are long. Second, there are also missing pixels in the raw measurements because not all pixels are activated during the data collection process. Third, there are much fewer frames in the raw video because many original frames are compressed into a single coded frame. Consequently, training data may be scarce.

In this study, we have focused our effort into simulating the measurements that should be produced by the PCE-based compressive sensing (CS) sensor. We then proceed to show that detecting, tracking, and even classifying moving objects of interest in the scene is feasible. We carried out multiple experiments with three diverse sensing models: PCE/CA Full, PCE/CA 50%, and PCE/CA 25%. PCE full refers to the compression of 30 frames to 1 with no missing pixels. PCE 50 is the case where we compress 30 frames to 1 and at the same time, only 50% of pixels are activated for a length of 4/30 seconds. PCE 25 is similar to PCE 50 except that only 25% of the pixels are activated for 4/30 seconds.

Table 1 below summarizes the comparison between the three sensing models. Details can be found in [19].

2.2. YOLO

Strictly speaking, YOLO is a detector rather than a tracker. Here, tracking is done via detection. That is, we apply YOLO to detect multiple targets and the target locations are extracted in every frame. Collecting the location information from the various frames will then create target trajectories.

Table 1. Comparison in data compression ratio and power saving ratio between three sensing models.

	PCE Full/CA Full	PCE 50%/CA 50%	PCE 25%/CA 25%
Data Saving Ratio	30:1	60:1	120:1
Power Saving Ratio	1:1	15:1	30:1

YOLO tracker [20] is fast and has similar performance as Faster R-CNN [24]. We picked YOLO because it is easy to install and is also compatible with our hardware, which seems to have a hard time to install and run Faster R-CNN. The training of YOLO is quite simple. Images with ground truth target locations are needed.

YOLO has 24 convolutional layers followed by 2 fully connected layers. Details can be found in [20]. The input images are resized to 448 × 448. It has some built-in capability to deal with different target sizes and illuminations. However, it is found that histogram matching is essential in order to make the tracker more robust to illumination changes.

YOLO also comes with a classification module. However, based on our evaluations, the classification accuracy using YOLO is not as good as ResNet in Section 3. This is perhaps due to a lack of training data.

2.3. ResNet Classifier

The ResNet-18 model is an 18-layer convolutional neural network (CNN) that has the advantage of avoiding performance saturation and/or degradation when training deeper layers, which is a common problem among other CNN architectures. The ResNet-18 model avoids the performance saturation by implementing an identity shortcut connection, which skips one or more layers and learns the residual mapping of the layer rather than the original mapping.

Training of ResNet requires target patches. The targets are cropped from training videos. Mirror images are then created. We then perform data augmentation using scaling (larger and smaller), rotation (every 45 degrees), and illumination (brighter and dimmer) to create more training data. For each cropped target, we are able to create a data set with 64 more images.

2.4. Data

We have mid-wave infrared (MWIR) and long-wave infrared (LWIR) videos from our sponsor. There are two videos from each imager: Video 4 and Video 5. Vehicles in Video 4 start from a parking lot and then travel to a remote location. Video 5 is just the opposite. Each frame contains up to three vehicles (Ram, Silverado, and Frontier), which are shown below in **Figure 2**.

It is challenging for target tracking and classification using the above videos for several reasons. First, the target orientation changes from the top view to side views. Second, the target size varies a lot in different frames. Third, the illumination is also different. Fourth, the vehicles look very similar to one another, as can be seen in **Figure 2**.

Here, we also briefly mention the image characteristics of SWIR, MWIR, and LWIR. From **Figure 3** [25], one can see the bands are different. SWIR lies in the range of 0.9 to 1.7 microns; MWIR is in the range of 3 to 5 microns; LWIR is within the range of 8 to 14 microns. Because of those different wavelength ranges, the image characteristics are very different, as can be seen in **Figure 4** and **Figure 5**. The daytime and nighttime behaviors are also different.

Figure 2. Pictures of Ram, Frontier, and Silverado.

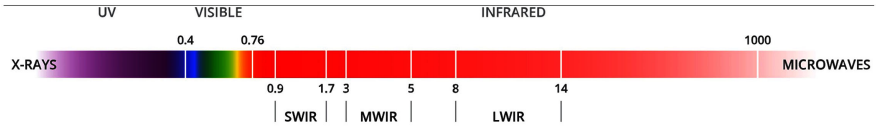
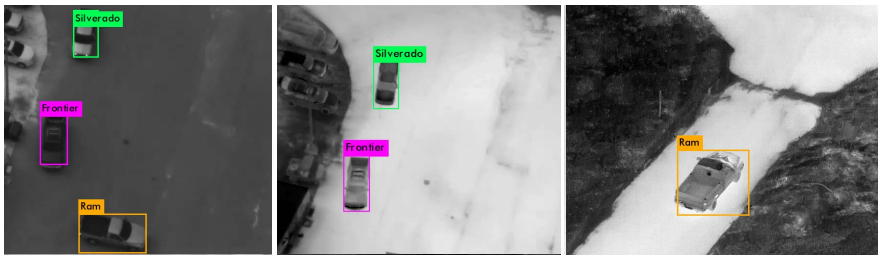


Figure 3. Spectrum of SWIR, MWIR, and LWIR [25].



SWIR frame; MWIR frame; LWIR frame

Figure 4. Frames from SWIR, MWIR, and LWIR videos. For MWIR and LWIR videos, the engine parts of the vehicles are brighter due to heat radiation. The road pixels are also bright due to heat from the impervious surface.

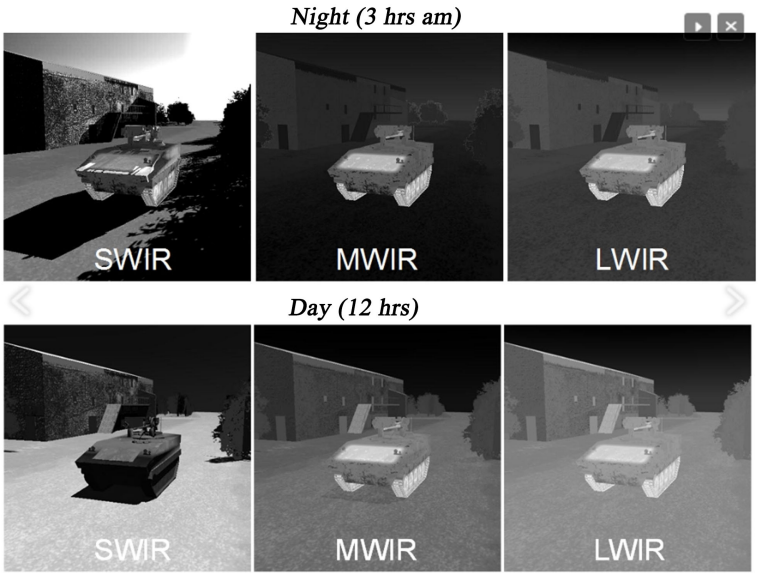


Figure 5. Different image characteristics of SWIR, MWIR, and LWIR images in night and day [26]. Objects in SWIR videos have shadows whereas MWIR and LWIR do not have shadows. The engine parts are also brighter than other parts.

2.5. Performance Metrics

We used the following metrics for evaluating the YOLO tracker performance:

- Center Location Error (CLE): It is the error between the center of the bounding box and the ground-truth bounding box.
- Distance Precision (DP): It is the percentage of frames where the centroids of detected bounding boxes are within 20 pixels of the centroid of ground-truth bounding boxes.
- EinGT: It is the percentage of the frames where the centroids of the detected bounding boxes are inside the ground-truth bounding boxes.
- Number of frames with detection: This is the total number of frames that have detection.

For classification, we used confusion matrix and classification accuracy as performance metrics.

3. Tracking and Classification Results Using MWIR Videos

In a companion paper [19], we have applied the YOLO + ResNet framework to some SWIR videos directly in compressive measurement domain. Since image characteristics are very different for SWIR, MWIR, LWIR, it is necessary to carry out a new study to investigate the deep learning-based framework in [19]. Here, we focus on the case of tracking and classification using a combination of YOLO and ResNet for MWIR and LWIR videos. There are three cases.

We have two MWIR videos. Each one has close to 3000 frames. One video (Video 4) starts with vehicles (Ram, Frontier, and Silverado) leaving a parking lot and moves on to a remote location. Another video (Video 5) is just the opposite. In addition to the aforementioned challenges, the two videos are difficult for tracking and classification because the cameras also move in order to follow the targets.

3.1. Tracking Results

Conventional tracker results

We first present some tracking results using a conventional tracker known as STAPLE [22]. STAPLE requires the target location to be known in the first frame. After that, STAPLE learns the target model online and tracks the target. However, even in PCE full cases as shown in **Figure 6** for MWIR videos and in **Figure 7** for LWIR videos. STAPLE was not able to track any targets in subsequent frames. This shows the difficulty of target tracking using PCE cameras.

MWIR: Train using Video 4 and Test using Video 5

We used YOLO tracker here. Video 4 was used for training and Video 5 for testing. Four performance metrics were used in the studies. **Tables 2-4** show the tracking results for PCE full, PCE 50, and PCE 25, respectively. In **Table 2** (PCE full case), one can see that the percentages of correct detection are very high. The CLE is around 5 pixels and the DP and EinGT values are all close to 100%. In **Table 3** (PCE 50 case), we observe that the percentages of correct detection start to drop. The CLE values become higher as compared to PCE full. The DP and EinGT values are still good. For the PCE 25 case (**Table 4**), the percentages of

frames with detection are even lower as compared to the other two cases. The CLE values are getting bigger. The general trend is that when the compression ratio increases, the performance drops accordingly. This can be corroborated in the snapshots shown in Figures 8-10 where more incorrect labels can be seen in the high compression cases. It should be noted that labels came from the YOLO tracker, which has inferior performance than ResNet. We will see more classification results in the later sections.

Table 2. Tracking metrics for PCE full. Train using Video 4 and test using Video 5.

	CLE	DP	EinGT	Number of frames with detection
Ram	5.17	1	1	85/89
Frontier	4.23	1	1	85/89
Silverado	4.58	1	0.96	70/89

Table 3. Tracking metrics for PCE 50. Train using Video 4 and test using Video 5.

	CLE	DP	EinGT	Number of frames with detection
Ram	7.58	1	0.99	76/89
Frontier	6.26	1	1	79/89
Silverado	6.75	1	0.95	62/89

Table 4. Tracking metrics for PCE 25. Train using Video 4 and test using Video 5.

	CLE	DP	EinGT	Number of frames with detection
Ram	8.89	1	1	58/89
Frontier	7.27	1	1	63/89
Silverado	8.31	1	0.95	40/89

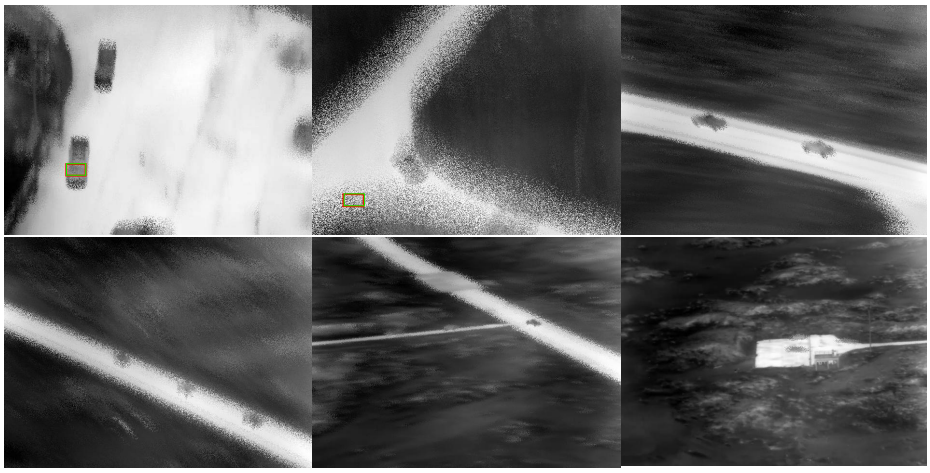


Figure 6. STAPLE tracking results for the MWIR PCE full case. Frames: 10, 30, 50, 70, 90, 110 are shown here. STAPLE cannot track any targets in subsequent frames.

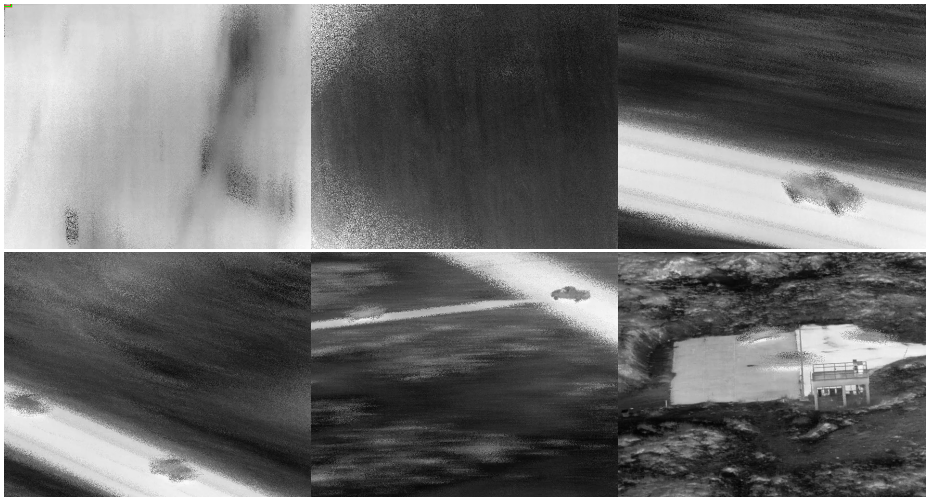


Figure 7. STAPLE tracking results for the LWIR PCE full case. Frames: 10, 30, 50, 70, 90, 110 are shown here. STAPLE cannot track any targets in subsequent frames.

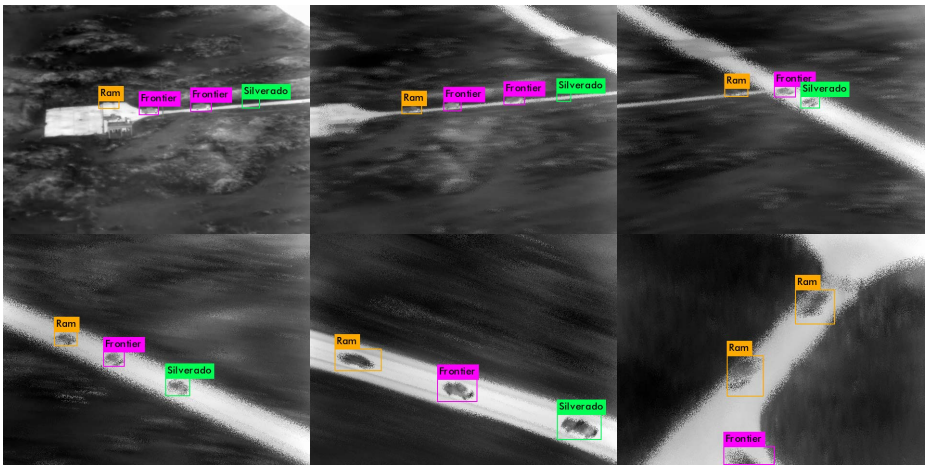


Figure 8. Tracking results for frames 1, 15, 29, 43, 57, and 71. PCE full case. Train using Video 4 and test using Video 5.

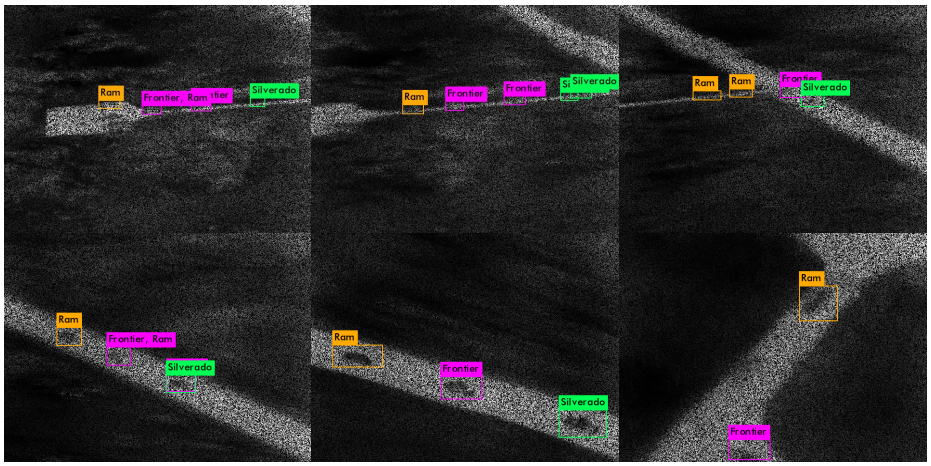


Figure 9. Tracking results for frames 1, 15, 29, 43, 57, and 71. PCE 50 case. Train using Video 4 and test using Video 5.

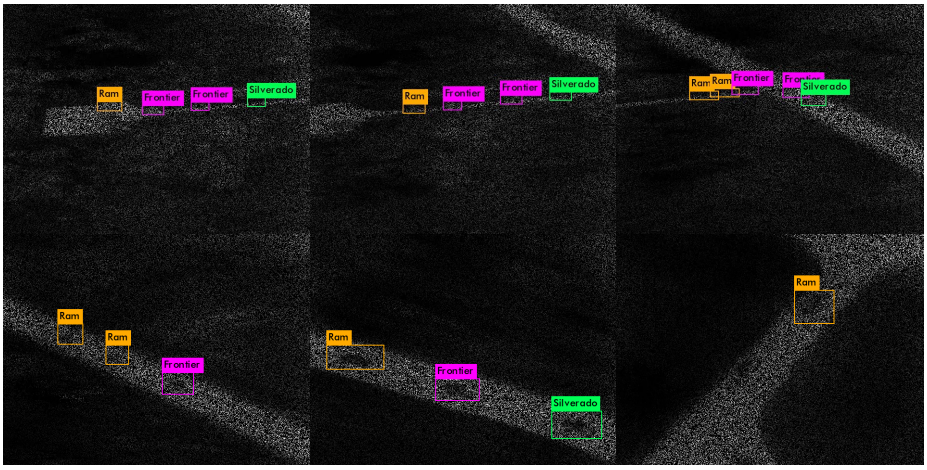


Figure 10. Tracking results for frames 1, 15, 29, 43, 57, and 71. PCE 25 case. Train using Video 4 and test using Video 5.

MWIR: Train using Video 5 and Test using Video 4

This is the reverse case where Video 5 was used for training and Video 4 for testing. Tables 5-7 show the tracking results for PCE full, PCE 50, and PCE 25, respectively. The trend is that when the compression ratio increases, the performance drops accordingly. This can be confirmed in the snapshots shown in Figures 11-13 where we can see that some targets do not have bounding boxes around them in the high compression cases. We also see that more incorrect labels in high compression cases.

3.2. Classification Results

Here, we applied two classifiers: YOLO and ResNet. It should be noted that classification is performed only when there are good detection results from the YOLO tracker. For some frames in the PCE 50 and PCE 25, there may not be positive detection results and for those frames, we do not generate any classification results.

MWIR: Training Using Video 4 and Testing Using Video 5

Here, Video 4 was used for training and Video 5 for testing. Tables 8-10 show the classification results using YOLO and ResNet for PCE full, PCE 50, and PCE 25, respectively. In each table, the left side shows the confusion matrix and the last column shows the classification accuracy. In all cases, the first observation is that the ResNet performance is better than that of YOLO. For instance, the averaged classification accuracy in ResNet is 0.5 and the averaged classification accuracy for YOLO is only 0.31 in the PCE full case. The second observation is that the classification performance deteriorates with high missing rates. Due to aggressive compression (>30:1), the ResNet classification rates are also low in the PCE 25 case. Third, we also notice that Frontier has higher classification accuracy than Ram and Silverado. This is probably because RAM and Silverado may have similar appearance, as can be seen from the confusion matrices in those tables.

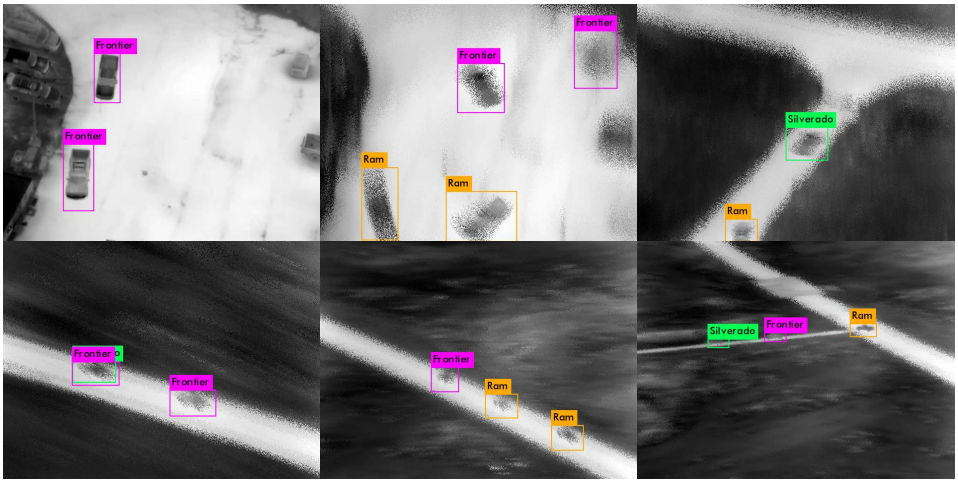


Figure 11. Tracking results for frames 1, 19, 37, 55, 73, and 91. PCE full case. Train using Video 5 and test using Video 4.

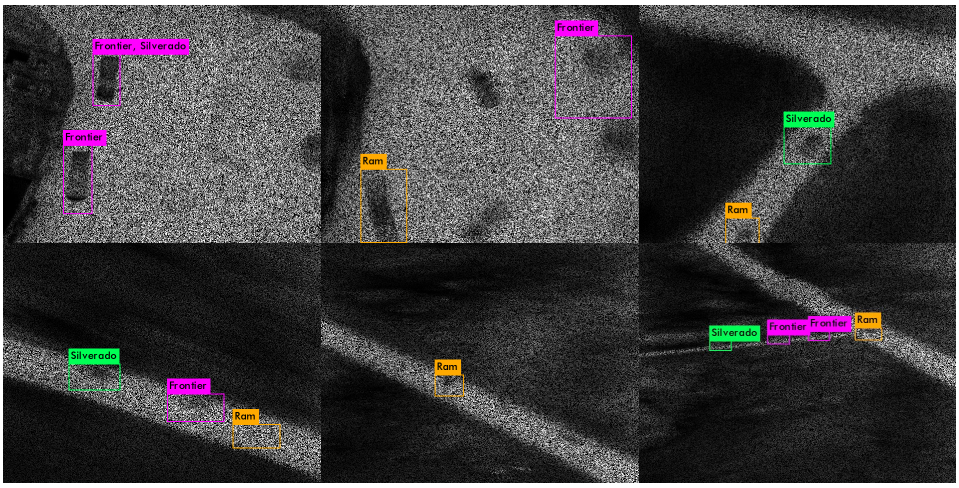


Figure 12. Tracking results for frames 1, 19, 37, 55, 73, and 91. PCE 50 case. Train using Video 5 and test using Video 4.

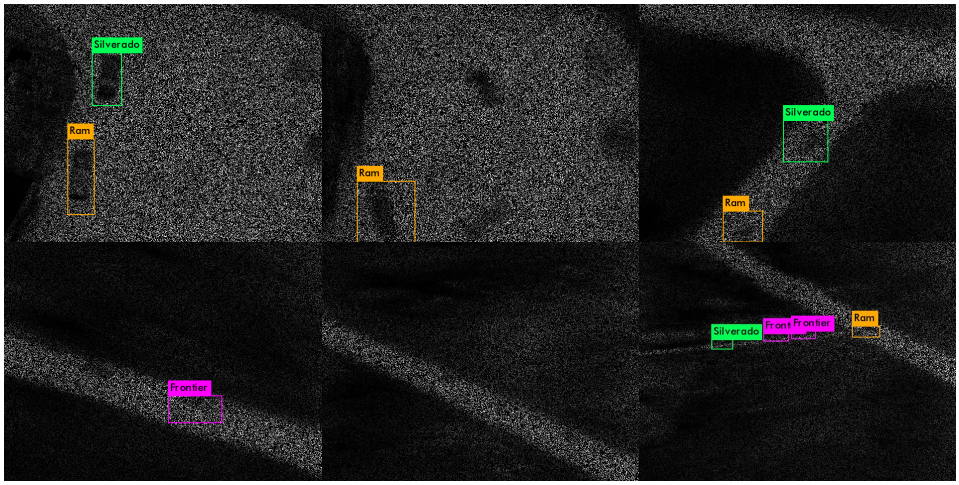


Figure 13. Tracking results for frames 1, 19, 37, 55, 73, and 91. PCE 25 case. Train using Video 5 and test using Video 4.

Table 5. Tracking metrics for PCE full. Train using Video 5 and test using Video 4.

	CLE	DP	EinGT	Number of frames with detection
Ram	6.31	1	0.97	93/110
Frontier	6.53	1	0.97	107/110
Silverado	6.19	1	1	66/110

Table 6. Tracking metrics for PCE 50. Train using Video 5 and test using Video 4.

	CLE	DP	EinGT	Number of frames with detection
Ram	7.72	1	0.97	86/110
Frontier	8.08	1	0.98	91/110
Silverado	8.5	1	1	50/110

Table 7. Tracking metrics for PCE 25. Train using Video 5 and test using Video 4.

	CLE	DP	EinGT	Number of frames with detection
Ram	9.27	1	0.98	64/110
Frontier	8.43	1	0.95	58/110
Silverado	7.75	1	1	24/110

Table 8. Classification results for PCE full case. Video 4 for training and Video 5 for testing. (a)YOLO classifier outputs; (b) ResNet classifier outputs.

(a)

		Actual			
		Ram	Frontier	Silverado	Classification Accuracy
Predicted	Ram	12	32	41	0.1412
	Frontier	15	65	2	0.7927
	Silverado	63	1	1	0.0154

(b)

		Actual			
		Ram	Frontier	Silverado	Classification Accuracy
Predicted	Ram	51	3	31	0.6000
	Frontier	30	45	10	0.5294
	Silverado	41	1	28	0.4000

Table 9. Classification results for PCE 50 case. Video 4 for training and Video 5 for testing. (a) YOLO classifier outputs; (b) ResNet classifier outputs.

(a)

		Actual			
		Ram	Frontier	Silverado	Classification Accuracy
Predicted	Ram	12	41	22	0.1600
	Frontier	22	52	1	0.6933
	Silverado	57	2	0	0.0000

(b)

		Actual			
		Ram	Frontier	Silverado	Classification Accuracy
Predicted	Ram	36	12	28	0.4737
	Frontier	34	42	3	0.5316
	Silverado	38	4	20	0.3226

Table 10. Classification results for PCE 25 case. Video 4 for training and Video 5 for testing. (a) YOLO classifier outputs; (b) ResNet classifier outputs.

(a)

		Actual			
		Ram	Frontier	Silverado	Classification Accuracy
Predicted	Ram	6	35	16	0.1053
	Frontier	24	35	2	0.5738
	Silverado	37	3	0	0.0000

(b)

		Actual			
		Ram	Frontier	Silverado	Classification Accuracy
Predicted	Ram	22	30	6	0.3793
	Frontier	16	46	1	0.7302
	Silverado	15	10	15	0.3750

MWIR: Training Using Video 5 and Testing Using Video 4

Here, Video 5 was used for training and Video 4 for testing. The observations in Tables 11-13 are similar to the earlier case. That is, ResNet is better than YOLO and classification performance drops with high compression rates.

4. Tracking and Classification Results Using LWIR Videos

Here, we summarize the studies for LWIR videos.

4.1. Tracking Results

LWIR: Train using Video 4 and Test using Video 5

From Table 14 (PCE full) case, the CLE, DP, and EinGT metrics all look normal. The numbers of frames with detection are lower than those of MWIR. Frontier has higher detections than Ram and Silverado. For PCE 50 (Table 15) and PCE 25 cases (Table 16), we observe that the YOLO tracker has more missed detections when the compression ratio increases. DP and EinGT scores are all high. CLE scores increase as compression increases. Moreover, from Figures 14-16, we can see that there are some missed bounding boxes as well as incorrect labels around the vehicles.

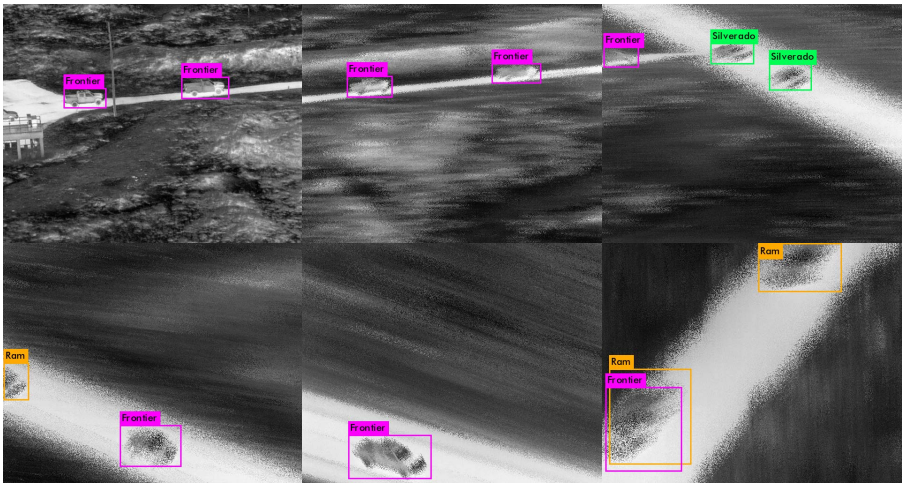


Figure 14. Tracking results for frames 1, 15, 29, 43, 57, and 71. PCE full case. Train using Video 4 and test using Video 5.

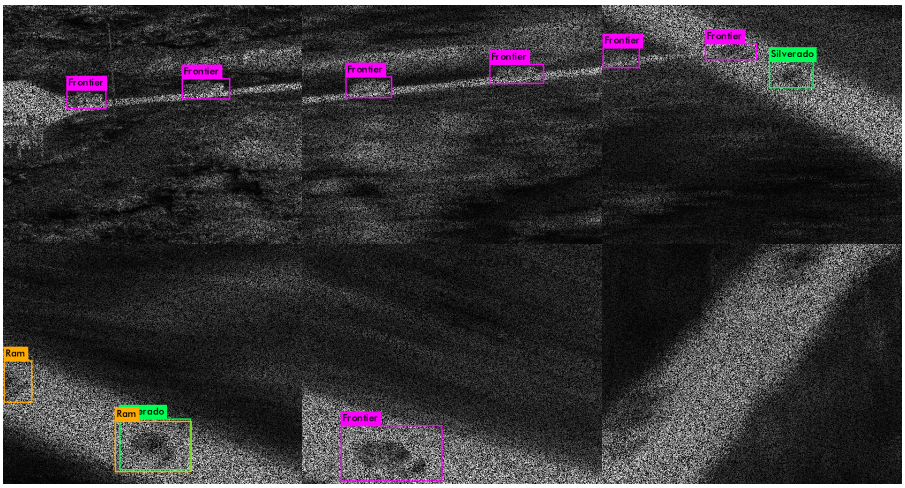


Figure 15. Tracking results for frames 1, 15, 29, 43, 57, and 71. PCE 50 case. Train using Video 4 and test using Video 5.

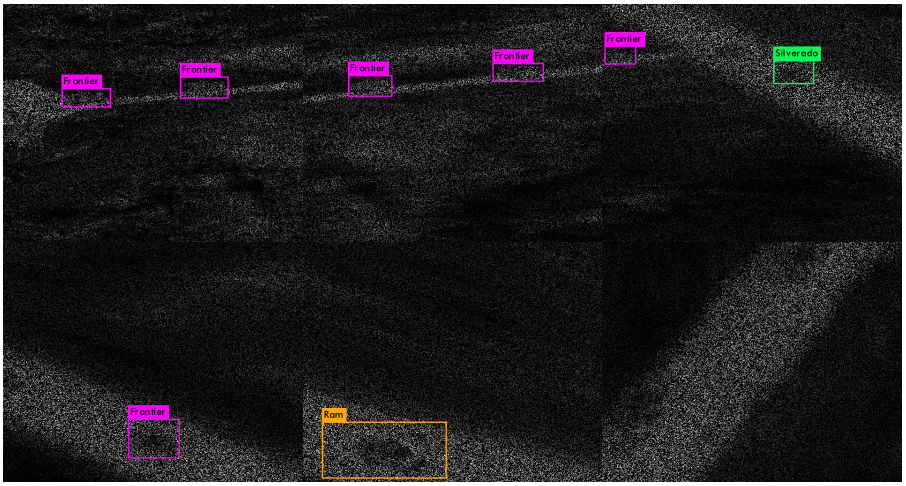


Figure 16. Tracking results for frames 1, 15, 29, 43, 57, and 71. PCE 25 case. Train using Video 4 and test using Video 5.

Table 11. Classification results for PCE Full case. Video 5 for training and Video 4 for testing. (a) YOLO classifier outputs; (b) ResNet classifier outputs.

(a)

		Actual			
		Ram	Frontier	Silverado	Classification Accuracy
Predicted	Ram	17	38	38	0.1828
	Frontier	44	61	0	0.5810
	Silverado	51	13	1	0.0154

(b)

		Actual			
		Ram	Frontier	Silverado	Classification Accuracy
Predicted	Ram	31	30	32	0.3333
	Frontier	6	94	7	0.8785
	Silverado	16	23	27	0.4091

Table 12. Classification results for PCE 50 case. Video 5 for training and Video 4 for testing. (a) YOLO classifier outputs; (b) ResNet classifier outputs.

(a)

		Actual			
		Ram	Frontier	Silverado	Classification Accuracy
Predicted	Ram	12	30	43	0.1412
	Frontier	44	43	2	0.4831
	Silverado	33	12	5	0.1000

(b)

		Actual			
		Ram	Frontier	Silverado	Classification Accuracy
Predicted	Ram	16	63	7	0.1860
	Frontier	15	73	3	0.8022
	Silverado	18	23	9	0.1800

Table 13. Classification results for PCE 25 case. Video 5 for training and Video 4 for testing. (a) YOLO classifier outputs; (b) ResNet classifier outputs.

(a)

		Actual			
		Ram	Frontier	Silverado	Classification Accuracy
Predicted	Ram	11	8	45	0.1719
	Frontier	27	29	2	0.5000
	Silverado	10	2	12	0.5000

(b)

		Actual			
		Ram	Frontier	Silverado	Classification Accuracy
Predicted	Ram	11	48	5	0.1719
	Frontier	17	41	0	0.7069
	Silverado	11	12	1	0.0417

Table 14. Tracking metrics for PCE full. Train using Video 4 and test using Video 5.

	CLE	DP	EinGT	Number of frames with detection
Ram	4.33	1	1	54/89
Frontier	6.64	1	1	71/89
Silverado	5.09	1	0.96	25/89

Table 15. Tracking metrics for PCE 50. Train using Video 4 and test using Video 5.

	CLE	DP	EinGT	Number of frames with detection
Ram	7.47	1	1	50/89
Frontier	9.05	1	1	53/89
Silverado	5.75	1	1	16/89

Table 16. Tracking metrics for PCE 25. Train using Video 4 and test using Video 5.

	CLE	DP	EinGT	Number of frames with detection
Ram	7.58	1	1	31/89
Frontier	7.99	1	1	30/89
Silverado	5.2	1	1	11/89

LWIR: Train using Video 5 and Test using Video 4

Figures 17-19 and Tables 17-19 summarize the LWIR study where Video 5 was used for training and Video 4 for testing. Similar to earlier section, we have decent performance when the compression is low. Some partial targets can be tracked. In general, the percentages of frames with detection are lower as compared to the case of using Video 4 for training and Video 5 for testing. Moreover, the overall performance of tracking of LWIR videos is inferior to that of MWIR videos.

4.2. Classification Results

LWIR: Training Using Video 4 and Testing Using Video 5

We performed a comparative study between ResNet and the built-in YOLO classifiers. Tables 20-22 summarize the classification results for PCE full, PCE 50, and PCE 25, respectively. In each table, the left side shows the confusion ma-

trix and the last column shows the classification accuracies. We observe that ResNet has much higher classification accuracy than YOLO. The results here are similar to those in the MWIR cases. That is, ResNet classification results are much better than those of YOLO.

LWIR: Training Using Video 5 and Testing Using Video 4

We have similar observations as the previous LWIR case. In the PCE full case (Table 23), the ResNet results are very good for Ram and Frontier, but not for Silverado. When compression ratio is beyond 30 to 1, the classification rates drop significantly as can be seen in Table 24 and Table 25 in which Ram and Silverado have very poor classification rates. We think that, in the coded aperture camera case, we should not use high compression, as the targets will be smeared too much. Moreover, MWIR may be preferred over LWIR in target tracking and classification.

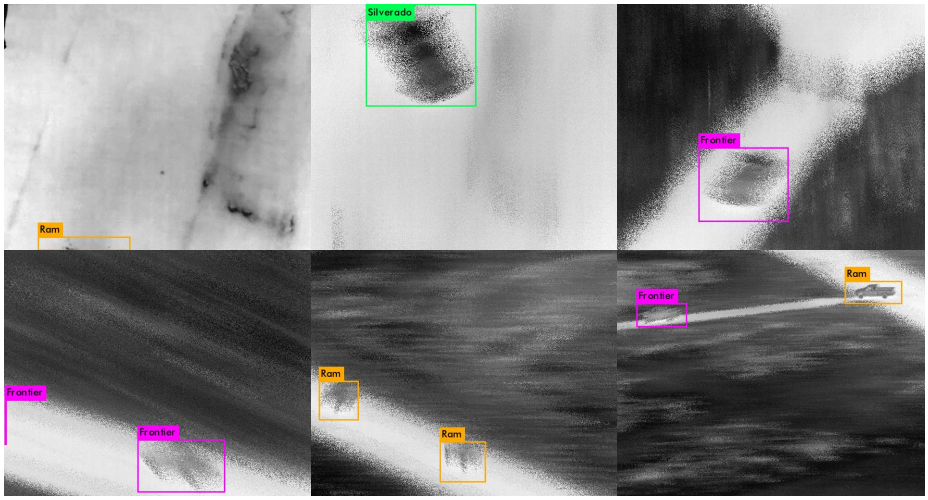


Figure 17. Tracking results for frames 1, 19, 37, 55, 73, and 91. PCE full case. Train using Video 5 and test using Video 4.

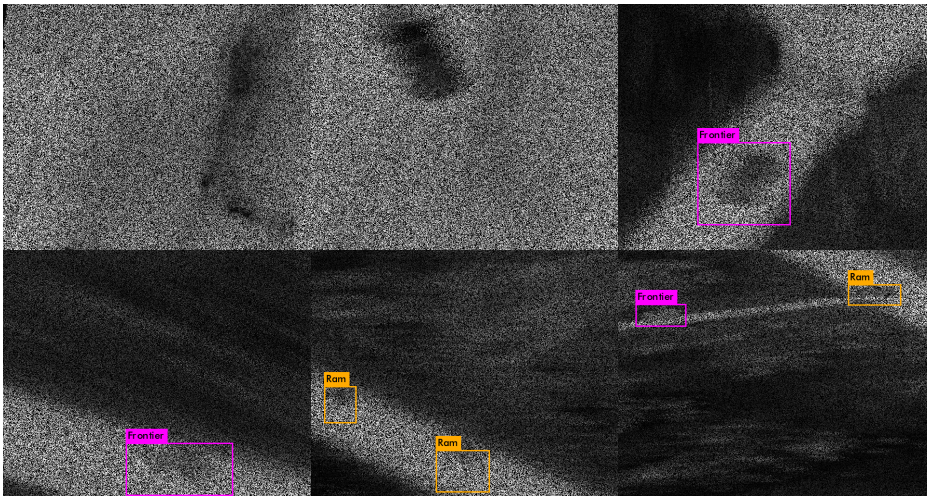


Figure 18. Tracking results for frames 1, 19, 37, 55, 73, and 91. PCE 50 case. Train using Video 5 and test using Video 4.

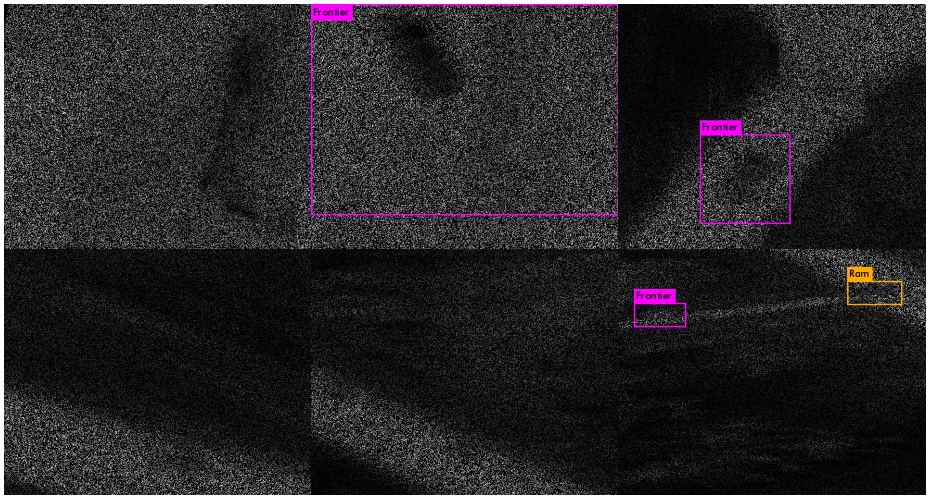


Figure 19. Tracking results for frames 1, 19, 37, 55, 73, and 91. PCE 25 case. Train using Video 5 and test using Video 4.

Table 17. Tracking metrics for PCE full. Train using Video 5 and test using Video 4.

	CLE	DP	EinGT	Number of frames with detection
Ram	9.73	1	0.88	34/110
Frontier	8.17	1	0.98	55/110
Silverado	8.55	1	1	23/110

Table 18. Tracking metrics for PCE 50. Train using Video 5 and test using Video 4.

	CLE	DP	EinGT	Number of frames with detection
Ram	13.44	1	1	8/110
Frontier	6.55	1	0.96	25/110
Silverado	9.5	1	1	10/110

Table 19. Tracking metrics for PCE 25. Train using Video 5 and test using Video 4.

	CLE	DP	EinGT	Number of frames with detection
Ram	15.32	1	1	4/110
Frontier	7.96	1	0.95	19/110
Silverado	9.92	1	1	8/110

Table 20. Classification results for PCE full case. Video 4 for training and Video 5 for testing. (a) YOLO classifier outputs; (b) ResNet classifier outputs.

(a)

		Actual			
		Ram	Frontier	Silverado	Classification Accuracy
Predicted	Ram	4	35	14	0.0755
	Frontier	23	45	1	0.6522
	Silverado	11	14	0	0.0000

(b)

		Actual			
		Ram	Frontier	Silverado	Classification Accuracy
Predicted	Ram	36	1	17	0.6667
	Frontier	35	32	4	0.4507
	Silverado	13	0	12	0.4800

Table 21. Classification results for PCE 50 case. Video 4 for training and Video 5 for testing. (a) YOLO classifier outputs; (b) ResNet classifier outputs.

(a)

		Actual			
		Ram	Frontier	Silverado	Classification Accuracy
Predicted	Ram	4	31	13	0.0833
	Frontier	15	38	0	0.7170
	Silverado	3	13	0	0.0000

(b)

		Actual			
		Ram	Frontier	Silverado	Classification Accuracy
Predicted	Ram	25	20	5	0.5000
	Frontier	19	30	4	0.5660
	Silverado	3	12	1	0.0625

Table 22. Classification results for PCE 25 case. Video 4 for training and Video 5 for testing. (a) YOLO classifier outputs; (b) ResNet classifier outputs.

(a)

		Actual			
		Ram	Frontier	Silverado	Classification Accuracy
Predicted	Ram	3	22	3	0.1071
	Frontier	5	25	0	0.8333
	Silverado	0	11	0	0.0000

(b)

		Actual			
		Ram	Frontier	Silverado	Classification Accuracy
Predicted	Ram	11	19	1	0.3548
	Frontier	6	24	0	0.8000
	Silverado	0	11	0	0.0000

Table 23. Classification results for PCE Full case. Video 5 for training and Video 4 for testing. (a) YOLO classifier outputs; (b) ResNet classifier outputs.

(a)

		Actual			
		Ram	Frontier	Silverado	Classification Accuracy
Predicted	Ram	7	22	3	0.2188
	Frontier	17	36	0	0.6792
	Silverado	21	0	2	0.0870

(b)

		Actual			
		Ram	Frontier	Silverado	Classification Accuracy
Predicted	Ram	30	2	2	0.8824
	Frontier	13	35	7	0.6364
	Silverado	19	0	4	0.1739

Table 24. Classification results for PCE 50 case. Video 5 for training and Video 4 for testing. (a) YOLO classifier outputs; (b) ResNet classifier outputs.

(a)

		Actual			
		Ram	Frontier	Silverado	Classification Accuracy
Predicted	Ram	3	5	0	0.3750
	Frontier	1	24	0	0.9600
	Silverado	7	0	3	0.300

(b)

		Actual			
		Ram	Frontier	Silverado	Classification Accuracy
Predicted	Ram	0	8	8	0.0000
	Frontier	10	14	1	0.5600
	Silverado	7	1	2	0.2000

Table 25. Classification results for PCE 25 case. Video 5 for training and Video 4 for testing. (a) YOLO classifier outputs; (b) ResNet classifier outputs.

(a)

		Actual			
		Ram	Frontier	Silverado	Classification Accuracy
Predicted	Ram	0	4	0	0.0000
	Frontier	0	19	0	1.0000
	Silverado	4	1	3	0.3750

(b)

		Actual			
		Ram	Frontier	Silverado	Classification Accuracy
Predicted	Ram	0	4	0	0.0000
	Frontier	3	16	0	0.8421
	Silverado	3	5	0	0.0000

5. Conclusions

In this paper, we present a high-performance approach to target tracking and classification directly in the compressive sensing domain for MWIR and LWIR videos. Skipping the time consuming reconstruction step will allow us to perform real-time target tracking and classification. The proposed approach is based on a combination of two deep learning schemes: YOLO for tracking and ResNet for classification. The proposed approach is suitable for applications where limited training data are available. Experiments using MWIR and LWIR videos clearly demonstrated the performance of the proposed approach. One key observation is that the MWIR has better tracking and classification performance than that of LWIR. Another observation is that the ResNet has much better performance than the built-in classification in YOLO.

One potential direction is to integrate our proposed approach with real hardware to perform real-time target tracking and classification directly in the compressive sensing domain.

Acknowledgements

This research was supported by the US Air Force under contract FA8651-17-C-0017. The views, opinions and/or findings expressed are those of the authors and should not be interpreted as representing the official views or policies of the Department of Defense or the U.S. Government.

Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

References

[1] Li, X., Kwan, C., Mei, G. and Li, B. (2006) A Generic Approach to Object Matching and Tracking. 3*rd International Conference Image Analysis and Recognition*, *Lecture Notes in Computer Science*, Póvoa de Varzim, 18-20 September 2006, 839-849. https://doi.org/10.1007/11867586_76

[2] Zhou, J. and Kwan, C. (2018) Tracking of Multiple Pixel Targets Using Multiple Cameras. 15*th International Symposium on Neural Networks*, Minsk, 25-28 June 2018, 484-493. https://doi.org/10.1007/978-3-319-92537-0_56

[3] Zhou, J. and Kwan, C. (2018) Anomaly Detection in Low Quality Traffic Monitoring Videos Using Optical Flow. *Pattern Recognition and Tracking*, Vol. 10649.

[4] Kwan, C., Zhou, J., Wang, Z. and Li, B. (2018) Efficient Anomaly Detection Algorithms for Summarizing Low Quality Videos. *Pattern Recognition and Tracking*, Vol. 10649. https://doi.org/10.1117/12.2303764

[5] Kwan, C., Yin, J. and Zhou, J. (2018) The Development of a Video Browsing and Video Summary Review Tool. *Pattern Recognition and Tracking*, Vol. 10649. https://doi.org/10.1117/12.2303654

[6] Zhao, Z., Chen, H., Chen, G., Kwan, C. and Li, X.R. (2006) IMM-LMMSE Filtering Algorithm for Ballistic Target Tracking with Unknown Ballistic Coefficient. *Proceedings SPIE*, Vol. 6236, *Signal and Data Processing of Small Targets*. https://doi.org/10.1117/12.665760

[7] Zhao, Z., Chen, H., Chen, G., Kwan, C. and Li, X.R. (2006) Comparison of Several Ballistic Target Tracking Filters. *Proceedings of American Control Conference*, Minneapolis, 14-16 June 2006, 2197-2202.

[8] Candes, E.J. and Wakin, M.B. (2008) An Introduction to Compressive Sampling. *IEEE Signal Processing Magazine*, **25**, 21-30. https://doi.org/10.1109/MSP.2007.914731

[9] Kwan, C., Chou, B. and Kwan, L.M. (2018) A Comparative Study of Conventional and Deep Learning Target Tracking Algorithms for Low Quality Videos. 15*th International Symposium on Neural Networks*, Minsk, 25-28 June 2018, 521-531. https://doi.org/10.1007/978-3-319-92537-0_60

[10] Kwan, C., Chou, B., Yang, J. and Tran, T. (2019) Target Tracking and Classification Directly in Compressive Measurement for Low Quality Videos. *Pattern Recognition and Tracking*, Baltimore, 16-18 April 2019. https://doi.org/10.1117/12.2518496

[11] Kwan, C., Chou, B., Echavarren, A., Budavari, B., Li, J. and Tran, T. (2018) Compressive Vehicle Tracking Using Deep Learning. *IEEE Ubiquitous Computing, Electronics & Mobile Communication Conference*, New York, 8-10 November 2018.

[12] Zhang, J., Xiong, T., Tran, T., Chin, S. and Etienne-Cummings, R. (2016) Compact All-CMOS Spatio-Temporal Compressive Sensing Video Camera with Pixel-Wise Coded Exposure. *Optics Express*, **24**, 9013-9024. https://doi.org/10.1364/OE.24.009013

[13] Tropp, J.A. (2004) Greed Is Good: Algorithmic Results for Sparse Approximation. *IEEE Transactions on Information Theory*, **50**, 2231-2242. https://doi.org/10.1109/TIT.2004.834793

[14] Dao, M., Kwan, C., Koperski, K. and Marchisio, G. (2017) A Joint Sparsity Approach to Tunnel Activity Monitoring Using High Resolution Satellite Images. *IEEE Ubiquitous Computing, Electronics & Mobile Communication Conference*, New York, 19-21 October 2017, 322-328. https://doi.org/10.1109/UEMCON.2017.8249061

[15] Zhou, J., Ayhan, B., Kwan, C. and Tran, T. (2018) ATR Performance Improvement Using Images with Corrupted or Missing Pixels. *Pattern Recognition and Tracking*, Vol. 10649, 106490E.

[16] Yang, J. and Zhang, Y. (2011) Alternating Direction Algorithms for 1-Problems in Compressive Sensing. *SIAM Journal on Scientific Computing*, **33**, 250-278. https://doi.org/10.1137/090777761

[17] Applied Research LLC, Phase 1 Final Report, August 2016.

[18] Yang, M.H., Zhang, K. and Zhang, L. (2012) Real-Time Compressive Tracking. *European Conference on Computer Vision*, Florence, 7-13 October 2012, 864-877.

[19] Kwan, C., Chou, B., Yang, J., Rangamani, A., Tran, T., Zhang, J. and Etienne-

Cummings, R. (2019) Target Tracking and Classification Directly Using Compressive Sensing Camera for SWIR Videos. *Journal of Signal, Image, and Video Processing*, 1-9. https://doi.org/10.1007/s11760-019-01506-4

[20] Redmon, J. and Farhadi, A. (2018) YOLOv3: An Incremental Improvement.

[21] He, K., Zhang, X., Ren, S. and Sun, J. (2016) Deep Residual Learning for Image Recognition. *Conference on Computer Vision and Pattern Recognition*, Las Vegas, 27-30 June 2016, 770-778. https://doi.org/10.1109/CVPR.2016.90

[22] Bertinetto, L., Valmadre, J., Golodetz, S., Miksik, O. and Torr, P. (2016) Staple: Complementary Learners for Real-Time Tracking. *Conference on Computer Vision and Pattern Recognition*, Las Vegas, 27-30 June 2016, 1401-1409. https://doi.org/10.1109/CVPR.2016.156

[23] Stauffer, C. and Grimson, W.E.L. (1999) Adaptive Background Mixture Models for Real-Time Tracking, Computer Vision and Pattern Recognition. *IEEE Computer Society Conference*, Vol. 2, Ft. Collins, 23-25 June 1999, 2246.

[24] Ren S., He, K., Girshick, R. and Sun, J. (2015) Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *Advances in Neural Information Processing Systems*, Montreal, 7-12 December 2015, 91-99.

[25] https://www.opto-e.com/resources/infrared-theory

[26] http://www.oktal-se.fr/website/index.php/solutions-system-design