

Received December 9, 2021, accepted December 23, 2021, date of publication December 27, 2021, date of current version January 6, 2022.

Digital Object Identifier 10.1109/ACCESS.2021.3138980

HM-Net: A Regression Network for Object Center Detection and Tracking on Wide Area Motion Imagery

HAKKI MOTORCU¹, (Student Member, IEEE), HASAN F. ATES², (Senior Member, IEEE),
H. FATIH UGURDAG¹, (Senior Member, IEEE), AND BAHADIR K. GUNTURK²

¹Electrical and Electronics Engineering Department, Ozyegin University, 34794 Istanbul, Turkey

²School of Engineering and Natural Sciences, Istanbul Medipol University, 34810 Istanbul, Turkey

Corresponding author: Hasan F. Ates (hfates@medipol.edu.tr)

This work was supported in part by TUBITAK under Grant 118E891.

ABSTRACT Wide Area Motion Imagery (WAMI) yields high resolution images with a large number of extremely small objects. Target objects have large spatial displacements throughout consecutive frames. This nature of WAMI images makes object tracking and detection challenging. In this paper, we present our deep neural network-based combined object detection and tracking model, namely, Heat Map Network (HM-Net). HM-Net is significantly faster than state-of-the-art frame differencing and background subtraction-based methods, without compromising detection and tracking performances. HM-Net follows object center-based joint detection and tracking paradigm. Simple heat map-based predictions support unlimited number of simultaneous detections. The proposed method uses two consecutive frames and the object detection heat map obtained from the previous frame as input, which helps HM-Net monitor spatio-temporal changes between frames and keep track of previously predicted objects. Although reuse of prior object detection heat map acts as a vital feedback-based memory element, it can lead to unintended surge of false positive detections. To increase robustness of the method against false positives and to eliminate low confidence detections, HM-Net employs novel feedback filters and advanced data augmentations. HM-Net outperforms state-of-the-art WAMI moving object detection and tracking methods on WPAFB dataset with its 96.2% F1 and 94.4% mAP detection scores, while achieving 61.8% mAP tracking score on the same dataset. This performance corresponds to an improvement of 2.1% for F1, 6.1% for mAP scores on detection, and 9.5% for mAP score on tracking over state-of-the-art.

INDEX TERMS Deep neural networks, object detection, tracking, wide area motion imagery.

I. INTRODUCTION

Object detection and tracking are broad and active subfields of computer vision. Recent improvements in deep neural networks and hardware capabilities have attracted more attention to these fields. Object detection and tracking applications on Wide Area Motion Imagery (WAMI) have had its share of the attention, and today it still holds its place as a challenging computer vision task. WAMI deals with monitoring a large area, which is several kilometers in diameter, with an airborne sensor consisting of multiple cameras. Captured frames from cameras are stitched to obtain a high-resolution image of the target area. Generally,

The associate editor coordinating the review of this manuscript and approving it for publication was Sudipta Roy.

a registration algorithm is applied to consecutive images to compensate for the camera motion. Object detection and tracking with WAMI have numerous applications in both civilian and military domains [1]–[4]. Traffic monitoring, driver behavior analysis, disaster response, and wildlife protection are just some examples in civilian domain. Military domain applications include intelligence-gathering, reconnaissance, border security, and surveillance.

Unmanned Aerial Vehicles (UAV) and WAMI sensors became more accessible in recent years. As a result of that trend, demand for robust detection and tracking algorithm for wide-area surveillance is constantly increasing.

Object detection in wide-area images has unique challenges compared to regular object detection [1], [6], [7]. Although wide-area images have higher resolution, since the

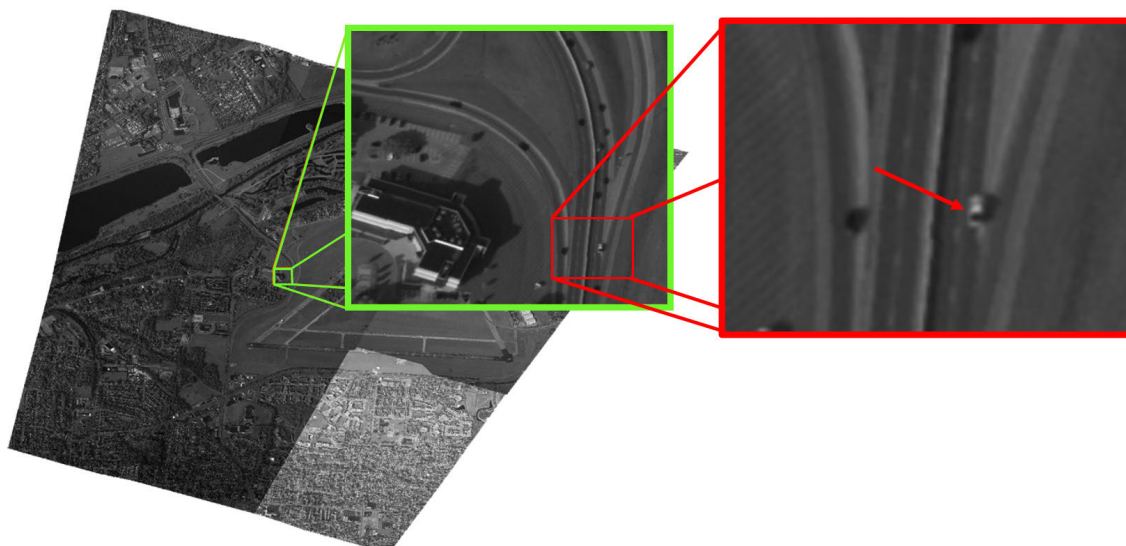


FIGURE 1. A frame from a WAMI footage [5] is magnified to show a vehicle location.

area represented by a single image is large, each target object is tiny (see Fig. 1) compared to normal object detection targets [1], [7]. Smaller target objects result in fewer features to exploit by the detection algorithm. Usually, WAMI sensors collect grayscale images, and that one channel input yields less information compared to three-channel RGB input. WAMI sensor can capture several thousand objects within a single frame due to high camera point of view, unlike regular video feeds, which are used in regular object detection. Also, ratio of combined pixel area of the objects to total image area is respectively smaller in WAMI. Ground truth information typically contains just center coordinates of objects; other necessary properties for supervised learning like object size and orientation are not provided. From object tracking perspective WAMI introduces large object displacements between consecutive frames and nearly none of the moving objects intersect with their appearances on previous frames. These large displacement vectors are due to slow frame update rate and the high speed of moving vehicles. Overall, a WAMI object tracker should detect and track high number of small objects which carry less appearance information and move very large distances between consecutive frames.

In order to address these existing challenges, we propose a deep neural network-based combined object detector and tracker architecture. Our method represents object locations as a heat map, where each object is represented as a 2D Gaussian placed on its center coordinate. By using center-based object detection we eliminate the need for bounding boxes, which are typically not provided by the WAMI datasets. Our method uses temporal information alongside appearance information to get better detection and tracking results. We use the previous frame alongside the current frame as input to our algorithm. Also, we use a feedback loop that feeds the filtered previous heat map as input. Our method uses

deep neural network backbone HM-Net (Heat Map Network), which is specially built and trained for this task. HM-Net employs multiple separate encoders for each input to help model focus on each input independently and then fuses separately extracted features to reduce memory footprint and computational complexity of the model. HM-Net introduces shortcuts between layers and several activation adder blocks to increase overall model connectivity. One-stage nature of the model yields remarkable throughput rate compared to most state-of-the-art algorithms.

Our model training and feedback procedures are designed to address unique challenges of the WAMI object detection and tracking tasks. For more efficient feedback process and to increase detection performance, elimination of false positives is essential. False positive centers tend to accumulate on heat maps; when the network uses the previous predicted heat map as input, these false detection clusters have tendency to propagate and grow after several feedback cycles. To eliminate false positive propagation during inference, we apply several novel training techniques. In model training we adopt new data augmentation strategies to simulate false positive clusters and false confidence amplifications on feedback heat maps. These measures are taken to make our model learn the challenging real-life situations on feedback process. Also we use class based selective filtering and rendering algorithm to eliminate potential false positives or low confidence detections during inference. As a result of these novel features we have achieved state-of-the-art performance on both WAMI-based object detection and tracking, and our method is several times faster than recent WAMI detection and tracking algorithms. Our key contributions are:

- One-stage HM-Net deep neural network architecture for joint object detection and tracking. HM-Net does not

need background subtraction or any other preprocessing, is relatively faster than state-of-the-art WAMI detection algorithms and detects both moving and stationary targets with high accuracy.

- Introduction of Selective Gaussian Reconstruction (SGR) filter. SGR filter is applied during heat map feedback process. SGR aims to reduce false positive propagation error caused by the feedback loop by using two different mechanisms: 1) Class based confidence thresholding to eliminate low confidence detections 2) Confidence amplification for a high confidence detection to avoid potential false positives in its vicinity on the predicted heat map.
- Novel data augmentation techniques for training: Randomized Confidence Reduction (RCR) and Randomized Center Propagation (RCP) are designed for the training process to eliminate vanishing confidence and false positive propagation that is observed during inference.

In Section II state-of-the-art approaches are analyzed for their strengths and weaknesses. Section III explains our combined detection and tracking algorithm in detail. Tracking and detection performance of our approach is compared against state-of-the-art models in Section IV. Finally, conclusions and future research are listed in Section V.

II. RELATED WORK

WAMI-based object detection can be classified as spatial, temporal, and spatio-temporal methods according to which information is mostly utilized for detection. Since a complete object tracker has to detect objects for tracking, discussion will be mostly towards the state-of-the-art detection models and their related tracking applications in this section. Regular object detection and tracking tasks are mainly follow objects, which are closer to the are that the camera focuses. Those objects occupy large portions of the image and have more spatial features to exploit during detection and tracking. State-of-the-art detection algorithms [8]–[16] heavily rely on the appearance information of those objects. Since, on WAMI, objects are smaller, less detailed, and occupy less total area on the image, regular detection algorithms are likely to fail in detect target objects.

State-of-the-art single object and Multi-Object Tracking (MOT) algorithms [17]–[21] are mostly built on tracking-by-detection paradigm or joint detection and tracking paradigm. Hence, these tracking algorithms inherit problems related to their object detection backbones. In tracking-by-detection approach, most trackers [17], [18], [20] utilize intersections of the detections between consecutive frames for tracking. Although object re-identification (ID transfer) with intersections yields decent results in high proximity (regular) target objects, which have less displacement throughout time, on WAMI nearly none of the moving objects have any intersection throughout consecutive appearances due to the low frame rate. An alternative paradigm to tracking-by-detection is joint detection and tracking, which is used

by [19], [21], [22]. In joint detection and tracking, detection model supports tracking algorithm with extra information such as future or previous locations prediction of detected objects to ease target identity transfer. Joint algorithms are more resilient to distortions on inputs and give better results in tracking targets with extreme displacements that yield no appearance overlap.

State-of-the-art WAMI object detectors take advantage of the temporal changes in a video sequence. When consecutive images are registered (each pixel corresponds to the same location on each image) moving objects will change pixel values around its location in an ideal scenario. Background subtraction and consecutive frame differencing are main approaches to detect temporal differences, but have their challenges. Background subtraction-based algorithms [6], [23]–[27] use registered video frames to build a background model. From each frame, the algorithm subtracts the modeled background to detect changes and hence finds objects. Background subtraction-based detection can be used solely on moving objects and cannot detect stationary objects. Even slow objects can cause detection problems. Parallax effect, stitching artifacts, illumination variations (see Fig. 2) between frames, and minor frame registration problems will lead to false detections. Construction of the background model also takes several frames and increases algorithmic complexity and memory utilization. Frame differencing-based methods [28]–[32] also rely on registered images to detect motion. Unlike background subtraction, frame differencing-based methods rely on fewer consecutive images. These methods use pixel-wise intensity variations between several consecutive frames (generally 2-3). Frame differencing methods suffer from similar problems with background subtraction. Both algorithms need perfect stitching, stable global illumination, and ideal frame registration for successful detections. Any distortion will lead to a significant number of false detections.

Recent approaches for object detection on WAMI (see Table 1) mostly use motion and appearance information together. This information fusion is generally done with Convolutional Neural Networks (CNN). As discussed above motion-based approaches give false detections due to stitching artifacts, registration errors and sudden pixel illumination changes. To filter false detections [33] utilizes a CNN post-processing module. Although CNN-based filtering decreases number of false detections, addition of a deep neural network on top of an already computationally expensive background subtraction algorithm decreases overall speed. As discussed above, regular object detection algorithms rely on just spatial information and do not perform well on WAMI. Modified versions of deep object detectors [1], [7], [34]–[36] use temporal information to increase overall detection success in both regular detection and WAMI scenarios. These models propose alternative ways to eliminate costly preprocessing methods, i.e. background subtraction and frame differencing. [34] and [36] use 3D CNN and stacked video frames for utilizing temporal information. [35] proposes a CNN

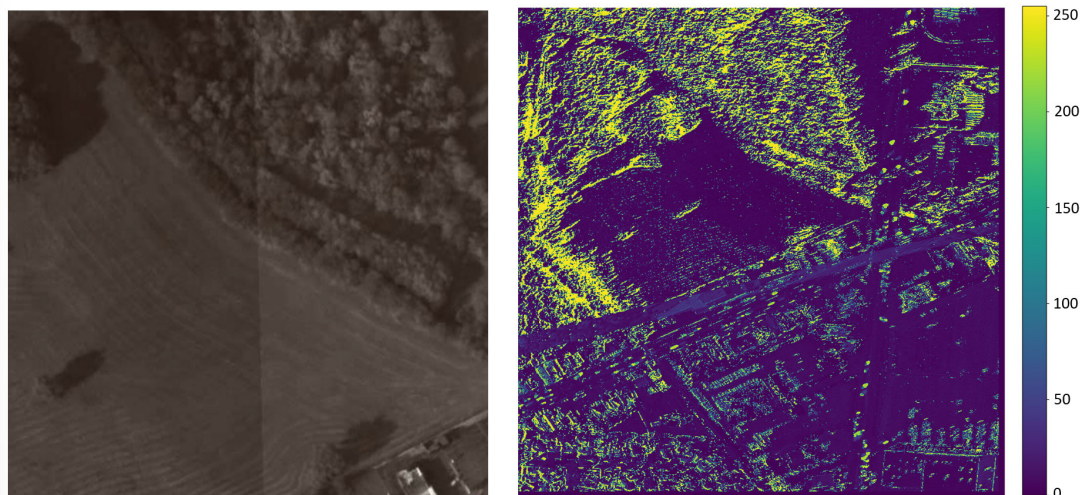


FIGURE 2. A stitching artifact example, which occurs due to different illumination levels of stitched images on the left. Differences between two consecutive registered frames are shown on the right. Error map shows unintended shifts during registration.

TABLE 1. Summary of WAMI detection and tracking methods in literature.

Methods	Detector	Use of Temporal Info.	Tracker	Limitations
Saleemi <i>et al.</i> [30]	Adaptive consecutive frame differencing	2-frame differencing	Multiple Hypothesis Tracker (MHT) and Hungarian matching algorithm	Overall speed of the algorithm is low. Absence of an explicit 1–1 matching constraint. Can not detect stationary vehicles.
Xiao <i>et al.</i> [31]	Frame differencing and tracking-based target refinement	3-frame differencing	The Graph Matching Framework combined with Vehicle Behavior Modeling	Overall speed of the algorithm is low. Can not detect stationary vehicles. Sensitive to distortions in input images.
Keck <i>et al.</i> [32]	Frame differencing and thresholding	3-frame differencing	MHT with a Kalman filter position/velocity motion model	Can not detect stationary vehicles. Sensitive to distortions in input images. Struggles in highly populated areas.
Reilly <i>et al.</i> [27]	Background subtraction	The median background model is build for every 10 frame cycle	Bipartite graph which uses Hungarian matching algorithm	Method cannot detect stationary vehicles. Sensitive to input distortions.
Pflugfelder <i>et al.</i> [7]	Modified version of FoveaNet and thresholding	Up to 5 prior frames are used by CNN model for vehicle detection	-	Method cannot detect stationary vehicles. Simple thresholding based detection struggles on occluded and close vehicles.
ClusterNet [1]	Two stage CNN architecture (FoveaNet and ClusterNet)	Up to 5 prior frames are used by CNN model for vehicle detection	-	Two stage nature of the model increases computational complexity and reduces speed.
Zhou <i>et al.</i> [33]	Background subtraction combined with CNN	Background modeling (5 frame model)	GM-PHD filtering	Speed of the method is vulnerable to number of proposed detections. Expensive background subtraction reduces model speed. Method can not detect stationary vehicles.
CenterTrack [22]	Modified version of CenterNet	Two consecutive frames used as input to CNN	Joint detection and tracking with greedy matching supported by one stage CNN predictions	Model is originally designed to work in regular object detection/tracking tasks.

that uses separate stacks carrying video frames and optical flow information. [37] proposes object tracking and trajectory calculation boost to object detection by including long

short-term memory sub-network and optical flow guided tracking algorithms. ClusterNet [1] uses a two-stage CNN that utilizes FoveaNet [38]. CNN-based detection approach

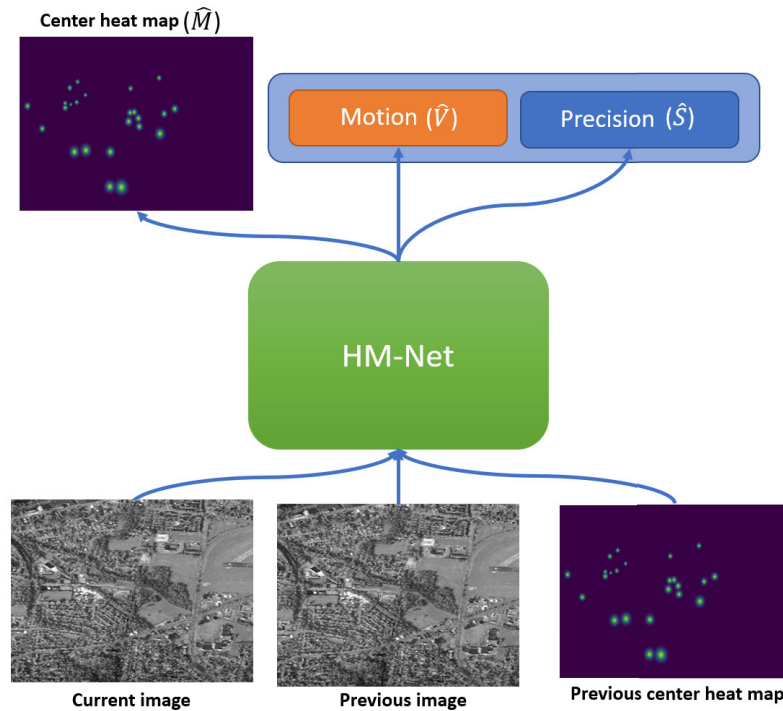


FIGURE 3. Input/Output (I/O) diagram of HM-Net.

enables ClusterNet [1] to detect stationary targets and moving objects simultaneously.

III. REGRESSION BASED DETECTION AND TRACKING

In this paper, we propose a deep neural network-based joint detection and tracking algorithm for WAMI. Inspired by center-based detection [11], [22], [39] and tracking [22] algorithms, our model adopts an anchor-box free approach using a heat map-based object localization. The model uses a one-stage auto-encoder architecture HM-Net as the backbone. Detection and tracking combined throughput rate of the method is about 4.5 fps on a RTX2080Ti 11GB GPU for marked areas of interest (AOIs) in WPAFB dataset [5]. For detecting and tracking moving objects, we split objects into moving vehicle and stationary vehicle classes. Separation of moving and stationary objects enables model to learn certain transitions between classes and reduces false positive detections caused by recently stopped objects. (Solely moving objects are considered during performance evaluation.)

Our method utilizes a center-based joint object detection and tracking paradigm. HM-Net predicts object centers heat map $\hat{M} \in [0, 1]^{C \times W \times H}$ (C : The number of object classes, W : width of input image, H : height of input image), motion heat map $\hat{V} \in \mathbb{R}^{2 \times W \times H}$, and subpixel location refinement heat map $\hat{S} \in [0, 1]^{2 \times W \times H}$ (see Fig. 3). These heat maps hold predictions of object center locations ($\mathbf{m}_i \in \mathbb{R}^2$), motion vectors ($\mathbf{v}_i \in \mathbb{R}^2$), and subpixel refinement of object locations

($\mathbf{s}_i \in \mathbb{R}^2$), respectively. Each heat map shares same width and height dimensions with the input image $I \in \mathbb{R}^{3 \times W \times H}$. Object center points \mathbf{m}_i are represented with 2D Gaussian-shaped peaks. Each predicted peak $\hat{\mathbf{m}}_i = (\hat{x}_i, \hat{y}_i)$ in $M_{c,:,:}$ indicates that there is an object center at coordinates (\hat{x}_i, \hat{y}_i) and it belongs to class c . Instead of labeling just center location with full confidence score of one, fitting a Gaussian curve encourages generalization [22]. To render a Gaussian peak around each object center \mathbf{m}_i on heat map M , we use the rendering function R given in 1, such that $M_{c,:,:} = R(\{\mathbf{m}_0, \mathbf{m}_1 \dots \mathbf{m}_n\})$ [40]:

$$R_{\mathbf{p}}(\{\mathbf{m}_0, \mathbf{m}_1 \dots \mathbf{m}_n\}) = \max_i \exp\left(-\frac{(\mathbf{m}_i - \mathbf{p})^2}{2\sigma^2}\right) \quad (1)$$

where $\mathbf{p} \in \mathbb{R}^2$ is any given position in the heat map, and σ is a constant scale parameter that is determined by average object size.

Local maxima of the predicted heat map \hat{M} yield the integer coordinates of the pixels that hold the object centers. To obtain vital subpixel locations, we use a separate precision head to regress subpixel position of the center ($\mathbf{s}_i \in [0, 1]^2$):

$$\mathbf{s}_i = \mathbf{m}_i - \lfloor \mathbf{m}_i \rfloor \quad (2)$$

A. DETECTION AND TRACKING

Object localization process is guided by the predicted center location heat map \hat{M} . As mentioned above, each local maximum $\hat{\mathbf{p}}_i$ of $\hat{M}_{c,:,:}$ corresponds to the center of a detected object with class c and a confidence value $\hat{w}_i = \hat{M}_{c,\hat{\mathbf{p}}_i}$.

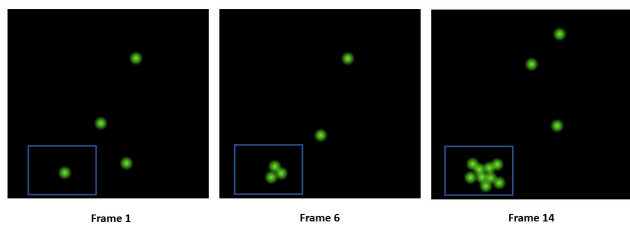


FIGURE 4. A false positive propagation example and its evolution through time is shown within blue rectangle.

The center location acquisition from \hat{M} is done within two steps: 1) Non-maxima suppression (NMS): max-pooling operation with window size 15×15 is applied to ensure adjacent high confidence pixels of a local maximum will not be mistaken as another local maximum. 2) Class specific threshold θ_c is applied to eliminate low confidence local maximums. Points that pass thresholding are considered as detected integer object locations. For subpixel center locations HM-Net uses $\hat{s}_i = \hat{S}_{\hat{p}_i}$ prediction which yields subpixel coordinates $\hat{x}_i \in [0, 1]$, $\hat{y}_i \in [0, 1]$ and $\hat{s}_i = (\hat{x}_i, \hat{y}_i)$. Finally integer and subpixel coordinate estimates are added to yield exact object location predictions: $\hat{\mathbf{m}}_i = \hat{\mathbf{p}}_i + \hat{s}_i$.

To track objects through time, we utilize our network to predict 2D motion vectors between consecutive frames using the motion head of the network: $\hat{\mathbf{v}} = \hat{V}_{\hat{\mathbf{p}}}$. Then, we organize all predictions to represent objects during tracking as $\hat{o} = \{\hat{\mathbf{m}}_i, \hat{w}_i, \hat{\mathbf{v}}_i, id_i\}_{i=0}^N$, where id_i is the assigned distinct track identity (ID) for each object. HM-Net uses $\hat{o}^{(t-1)}$ and $\hat{o}^{(t)}$ detections for object re-identification. From $\hat{o}^{(t)}$ center locations, $\hat{m}_i^{(t)}$ and motion vector $\hat{v}_i^{(t)}$ are utilized to predict potential previous location $\hat{b}_i = \hat{m}_i^{(t)} + \hat{v}_i^{(t)}$ of the object. Detections in $\hat{o}^{(t-1)}$ that are within an acceptable distance κ around \hat{b}_i are marked as potential matches (where κ a constant scale parameter that is determined by average object size). These marked detections in $\hat{o}^{(t-1)}$ are matched with the corresponding candidates in $\hat{o}^{(t)}$ according to their confidence scores (\hat{w}_i) by using Hungarian Matching algorithm [41]. Associated objects' tracking IDs are transferred from $\hat{o}^{(t-1)}$ to $\hat{o}^{(t)}$.

During center matching if a center has no match in the previous frame, we assume that center represents a new object and we generate a new ID for that object. For unmatched prior IDs, we keep such moving objects in a separate set, in case the object have stopped momentarily and classified as stationary or the object confidence is temporarily below the detection threshold. These object tracks could be re-activated in the upcoming frames if a high confidence match is found. Due to the fast motion characteristics of objects in WAMI, this re-identification of a passive track is only possible in the next frame.

Reuse of previous detections via feedback input holds an important place during tracking. Center-based trackers [19], [22] exploit the Gaussian-shaped object representation during feedback process. They render all detection information

into one channel heat map and use it as feedback. This feedback loop gives the prediction network further temporal information. Although feedback loop adds a short-term memory element to the network, it yields several problems like false-positive error propagation and vanishing or exploding confidence. False positives on a feedback heat map may lead network to generate many less confident false positives that are adjacent to each other. Early experiments revealed that within several iterations, false detection clusters may occupy a significant portion of the output (see Fig. 4).

Each detection has confidence score (between 0-1) determined by the local maximums on the predicted heat map. During the feedback process prediction network may amplify or suppress confidence of the detections. An effectively trained network can eliminate potential false positives or false negatives caused by feedback. Efficiency of training depends on the coverage of corner cases like sudden illumination changes within image due to stitching artifacts, minor image registration errors, and problematic areas that can cause ID switches between close objects like junctions etc. That is why countermeasures are taken against these challenges not only in feedback process but also during the training as well. Training techniques will be discussed in detail on the model training section.

Selective Gaussian Reconstruction (SGR): To tackle the problems related to changing object confidences during feedback loop, we design the SGR filter. Unlike simple thresholding methods used by [19], [22] on one channel feedback, we use a more complex filtering process with multi-channel feedback. Hence, class information is also provided during feedback by inputting the previous heat map in its full dimensions ($C \times W \times H$). For each class c , we use two thresholds: λ_c to control which local maximums will reappear on the filtered heat map; and θ_c to control which maximums will be considered as true detections. Extensive simulations show that class-specific thresholding gives better results due to unique confidence distributions of true positive detections in each class. Separation of detection (θ_c) and filtering thresholds (λ_c) also helps the network remember not only confident detections but also local peaks that may become future detections. After deciding which centers will be reconstructed on the heat map we render each Gaussian peak with amplified confidence scores. Confidence amplification factor (φ_c) is also class dependent. Functionality of SGR filter is shown in 3, where \hat{w}_i is the predicted confidence score of the local maximum and SGR filtered result is represented with w_i where $\theta_c > \lambda_c$:

$$w_i = \begin{cases} 0 & \lambda_c > \hat{w}_i \\ \hat{w}_i & \theta_c > \hat{w}_i \geq \lambda_c \\ \hat{w}_i \times \varphi_c & \hat{w}_i \geq \theta_c \end{cases} \quad (3)$$

A sample heat map input and output of SGR filter are shown in Fig. 5.

Algorithm 1 summarizes the major steps of the detection and tracking process. At each time step HM-Net uses current

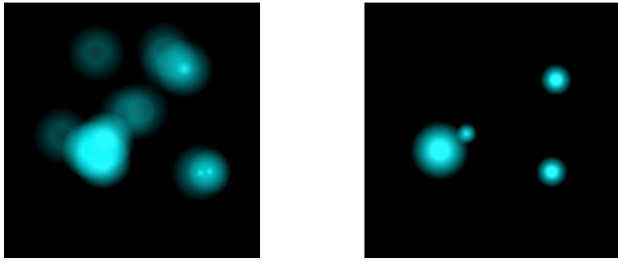


FIGURE 5. An heat map prediction sample before the SGR filter on the left, and after the SGR filter on the right.

Algorithm 1 Detection and Tracking Process

Input: Video Frames ($I^{(t)}$)
Output: Tracklets ($\hat{\delta}^{(t)}$)

- 1: **for** $t \leftarrow 1 \dots \text{Length of } I$ **do**
- 2: $\hat{M}, \hat{V}, \hat{S} \leftarrow \text{HM-Net}(I^{(t)}, I^{(t-1)}, \hat{M}^{(t-1)})$
- 3: $\hat{M} \leftarrow \text{NMS}(\hat{M})$
- 4: $\hat{\mathbf{p}}, \hat{\mathbf{c}} \leftarrow \text{Locate local maxima (where } \hat{M}_{\mathbf{c}} > \lambda_c)$
- 5: $\hat{\mathbf{w}} \leftarrow \hat{M}[\hat{\mathbf{c}}, \hat{\mathbf{p}}], \hat{\mathbf{v}} \leftarrow \hat{V}[:, \hat{\mathbf{p}}], \hat{\mathbf{s}} \leftarrow \hat{S}[:, \hat{\mathbf{p}}]$
- 6: $\hat{\mathbf{m}} \leftarrow \hat{\mathbf{p}} + \hat{\mathbf{s}}$
- 7: $\hat{\delta}^{(t)} \leftarrow \text{TRACKER}((\hat{\mathbf{m}}, \hat{\mathbf{c}}, \hat{\mathbf{w}}, \hat{\mathbf{v}}), \hat{\delta}^{(t-1)}, \theta)$
- 8: $\hat{M}^{(t)} \leftarrow \text{SGR}(\hat{\mathbf{p}}, \hat{\mathbf{c}}, \lambda, \theta, \varphi)$
- 9: **end for**

($I^{(t)}$) and previous ($I^{(t-1)}$) frames alongside filtered previous center heat map ($\hat{M}^{(t-1)}$) to predict \hat{M} , \hat{V} , and \hat{S} . Then non-maxima suppression (NMS) is applied to \hat{M} to extract local maxima. Local maxima are thresholded with λ_c to determine detections on \hat{M} together with their class ($\hat{\mathbf{c}}$) and position ($\hat{\mathbf{p}}$) information. By using $\hat{\mathbf{c}}$ and $\hat{\mathbf{p}}$, confidence scores, motion and precision vectors are extracted from predicted heat maps. Precision vectors are used to update the center position coordinates of detections. Extracted information and detection threshold are sent to Tracker to get tracklets. At the end of each time step SGR function is used to filter and render the feedback heat map.

B. HM-NET ARCHITECTURE

Center-based detection and tracking necessitates the prediction of large heat maps that contain location and class information of centers. The heat map prediction is a regression task that can be handled using deep-learning models. Overall heat map prediction task is similar to segmentation and human pose estimation tasks. These tasks can be handled using fully convolutional networks that produce output maps with dimensions proportional to input images. We have studied and tested several deep learning-based auto-encoder networks that gave state-of-the-art results on different tasks during development process of our own model, such as UNet [42] and Stacked Hourglass [43] architectures, in order to combine their strengths.

Although simple feed-forward deeper neural networks are expected to give better results, real-life tests show that deeper

plain neural networks face vanishing gradient problem that hinders shallow layers' learning capability [44]. Shortcuts between layers enable a more effective learning process for deeper and connected neural networks to reach expected high accuracy scores [44]. All examined architectures introduce shortcuts between layers alongside feed-forward path to increase connectivity. UNet architecture utilizes activation transfer between encoder and decoder to increase connectivity between distant layers. UNet uses concatenation for activation transfer. Stacked Hourglass [43] architecture uses summation operations between activations to ensure connectivity. Summation operation is faster; however it squeezes all activations into one. On the other hand, concatenation process yields larger activations that increase the volume of the network while preserving the information in separate channels. In HM-Net architecture, we aim to fuse these connection options for optimal trade-off between accuracy and model complexity.

Detailed structural diagram of the HM-Net is shown on Fig. 6. HM-Net architecture consists of two cascaded auto-encoder sections which have several shortcuts between modules. Model is designed as one-stage tracker and detector. It can obtain matched predictions in one feed-forward pass. One-stage nature of the architecture increases overall throughput speed. Cascaded auto-encoders are used to increase the depth of the model by facilitating deeper bottleneck blocks while not using too much memory. Deeper layers enable HM-Net to extract more complex features.

HM-Net takes two consecutive frames of input video and SGR filtered version of previous heat map to exploit temporal changes. Each input has a dedicated encoder for feature extraction process. Each encoder has 4 consecutive encoder blocks (Fig.7) and is followed by a deeper bottleneck layer. The role of encoder blocks is to increase the depth of activations ($\times 2$) while reducing channel dimensions to decrease memory utilization. At the end of first encoder stage separately extracted features are sent to 3 parallel bottleneck blocks which are later fused into one bottleneck by using summation operation, and fed through first stage decoder. First stage decoder consists of 4 consecutive decoder blocks (Fig.7). Second encoding and decoding modules have same data flow pattern with their counterparts in first auto-encoder. As Fig. 6 shows there are several shortcuts and activation adder blocks that are introduced to increase connectivity. Activation adder blocks simply calculate summations of connected blocks' outputs. The blue arrows in first stage encoder represent summation of corresponding activations coming from three separate encoder blocks (see Fig. 6). Except for the first stage bottleneck activation adder block, each adder block is followed by a concatenation operation which concatenates the input activation of a decoder block and the output of connected summation block. HM-Net has three separate prediction heads to host predictions separately, which are object center detection heat map head, precision head, and motion vector head. All outputs have the same width and height with the input image.

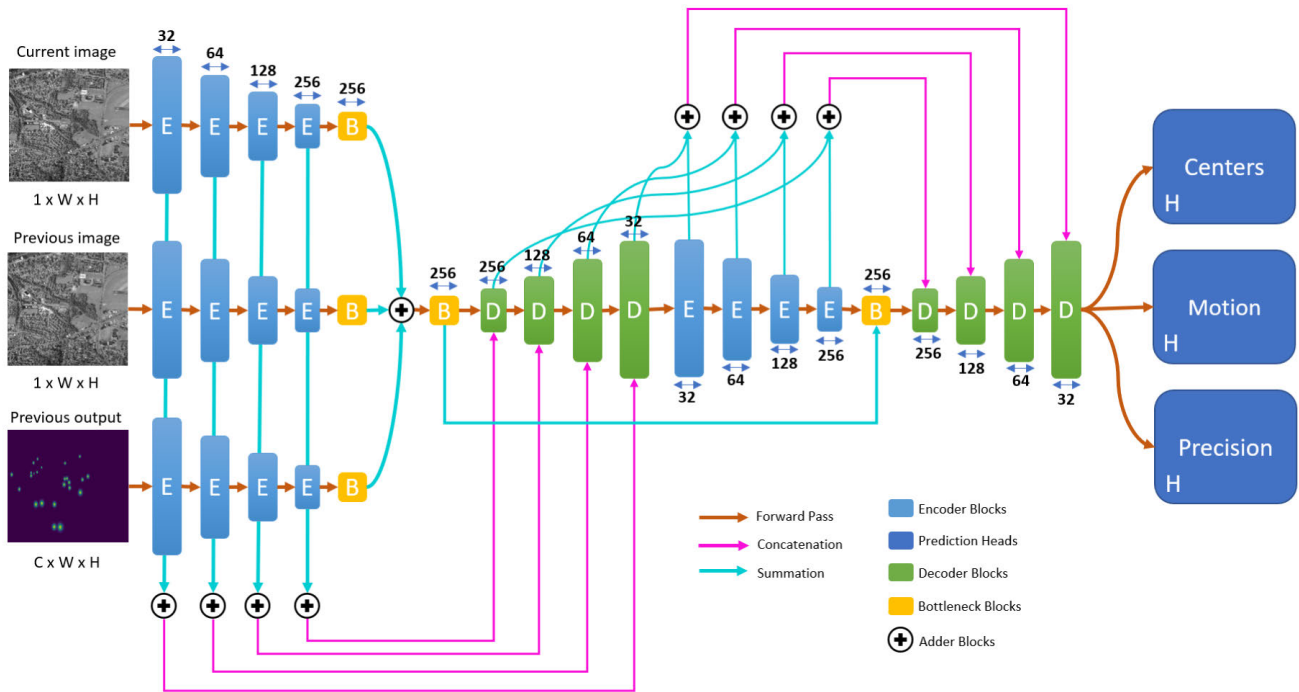


FIGURE 6. The HM-Net architecture diagram with block legend.

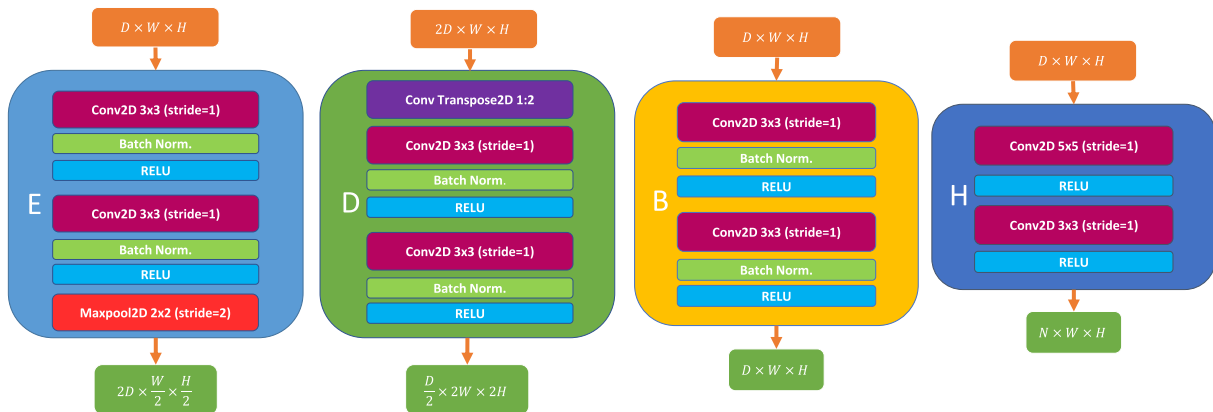


FIGURE 7. Detailed representations of the model. The Encoder block (E), the Decoder block (D), the Bottleneck block (B) and the Head block (H) are shown with their input (orange) and output (green) dimensions. Dimensions $D \times W \times H$ stand for depth, width and height. N is configurable parameter for head output depth; for Precision and Motion heads $N = 2$ and for Centers Head N equals to number of classes. Input depth $2D$ on encoder block is a result of concatenation based connections.

C. HM-NET TRAINING

WPAFB 2009 [5] dataset is used for both model training and evaluation. WPAFB dataset consists of roughly $25,000 \times 20,000$ grayscale stitched images with NITF format. In order to reduce computational complexity, the given set of images are downsampled by a factor of 2 to the average size of $12,500 \times 10,000$ pixels. We apply several preprocessing steps to obtain training-ready images. Images are converted to PNG format, and then a simple registration algorithm is applied to compensate for the camera motion. The training set consists of cropped versions images which have

544×960 dimensions. Smaller cropped training images reduces computational complexity and memory footprint during training. This enables usage of larger batch sizes during training.

As stated in Section III-A, our model uses previous image and output as feedback to exploit temporal and spatial information together. During inference, the SGR filter is applied on predicted heat map to improve prediction performance. SGR filter yields a reconstructed feedback heat map consisting of 2D Gaussian peaks according to their predicted confidence values. During training, we have to

simulate the actual feedback heat map of the model inference. This simulation should be realistic enough to encourage the model to learn corner cases that cause false positive propagation, vanishing confidence and location precision loss.

False positive propagation process is shown with three heat maps in Fig. 4. At first glance distribution of predicted confidence values have high variance and they are hard to simulate realistically during training. For feedback simulation in training, we use ground truth object centers of previous image when constructing the heat map. Setting confidence scores of ground truth objects as 1 does not lead to an effective training. In our initial trials, confidence scores \hat{w}_i of feedback detections were observed to be varying around 0.6. Since predictions during inference have varying confidence scores, a randomized confidence generation method that can simulate observed score variations is implemented for better feedback simulation during model training.

SGR filter ensures that heat map outputs have varying confidence scores but solid 2D Gaussian shapes as discussed above. Solidified Gaussian curves are more imitable than raw heat map outputs (see Fig.5). Our primary aim on training is simulating SGR filtered feedback heat maps in a realistic setup with several advanced data augmentations. These augmentation methods, such as random location shift and random center deletion, are key to increase generalization capability and emulate model inference. Random confidence reduction (RCR) method is applied on center confidences to simulate more realistic confidence scores on a feedback scenario. RCR applies up to 0.8 confidence reduction with 70% probability to each confidence ground truth w_i . The reduction factor and probability are set based on ablation studies to maximize training performance.

False positive propagation problem is shown in Fig. 4. Around one detection we see tree root shaped spreading false detection clusters. To reduce the occurrence of this unwanted situation we implement randomized center propagation (RCP). During heat map simulation RCP picks random labels and places several false detections around the ground truth to simulate the propagation problem. RCP is also used to place false positive object centers randomly on the feedback heat map to teach our network how to eliminate all types of false positive detections.

Data distributions among heat maps are imbalanced; in other words Gaussian peaks cover very small portion of heat maps. That is why we utilize a modified focal loss function from [11], [22], [40], [45]. The loss L_H is computed between the ground truth heat map $M_{c,x,y}$ and its prediction $\hat{M}_{c,x,y}$ for each class and at every pixel location:

$$L_H = -\frac{1}{N} \sum_{c,x,y} f(M_{c,x,y}, \hat{M}_{c,x,y}) \quad (4)$$

where

$$f(m, n) = \begin{cases} (1-n)^\alpha \log(n), & \text{if } m = 1 \\ (1-m)^\beta (n)^\alpha \log(1-n), & \text{otherwise} \end{cases} \quad (5)$$



FIGURE 8. Locations of cropped AOI's in a complete WAMI frame. Courtesy of [25].

TABLE 2. Size and location information of AOIs.

AOI	Center	Width	Height
01	(4700, 8050)	1200	1200
02	(5650, 5300)	1200	1200
03	(9400, 5950)	1200	1200
34	(5300, 6300)	2400	1600
40	(4850, 8050)	2000	1400
41	(6000, 8250)	1800	1600

where N is the total count of labeled objects and hyper-parameters of focal loss are set to $\alpha = 2$, $\beta = 4$.

For learning motion vectors and subpixel center locations, we use supervised linear loss (L_1 loss):

$$L_x = \left(\frac{1}{N}\right) \sum_{i=1}^N |\hat{\mathbf{x}}_i - \mathbf{x}_i| \quad (6)$$

where \mathbf{x}_i is \mathbf{v}_i for motion vectors and \mathbf{x}_i is \mathbf{s}_i for subpixel locations. The weighted sum of those loss functions is used during training. Weights are set by considering the importance of the corresponding head while maintaining training balance. For model optimization Adam optimizer is used with a learning rate of 0.0013. Model training with batch size 6 is concluded at 16th epoch to prevent overfitting.

IV. RESULTS AND DISCUSSIONS

In this section, first we discuss our detection and tracking results, in comparison with state-of-the-art methods from literature. Then an ablation study is presented to illustrate the effects of hyper-parameters and novel data augmentation methods on the detection and tracking performances.

A. EVALUATION OF DETECTION AND TRACKING PERFORMANCE

Our tests are conducted on the WPAFB dataset [5] at predetermined cropped AOIs (see Fig. 8 and Table 2). A detection is considered as true positive when there is at least one ground truth point within 10 pixels range and each

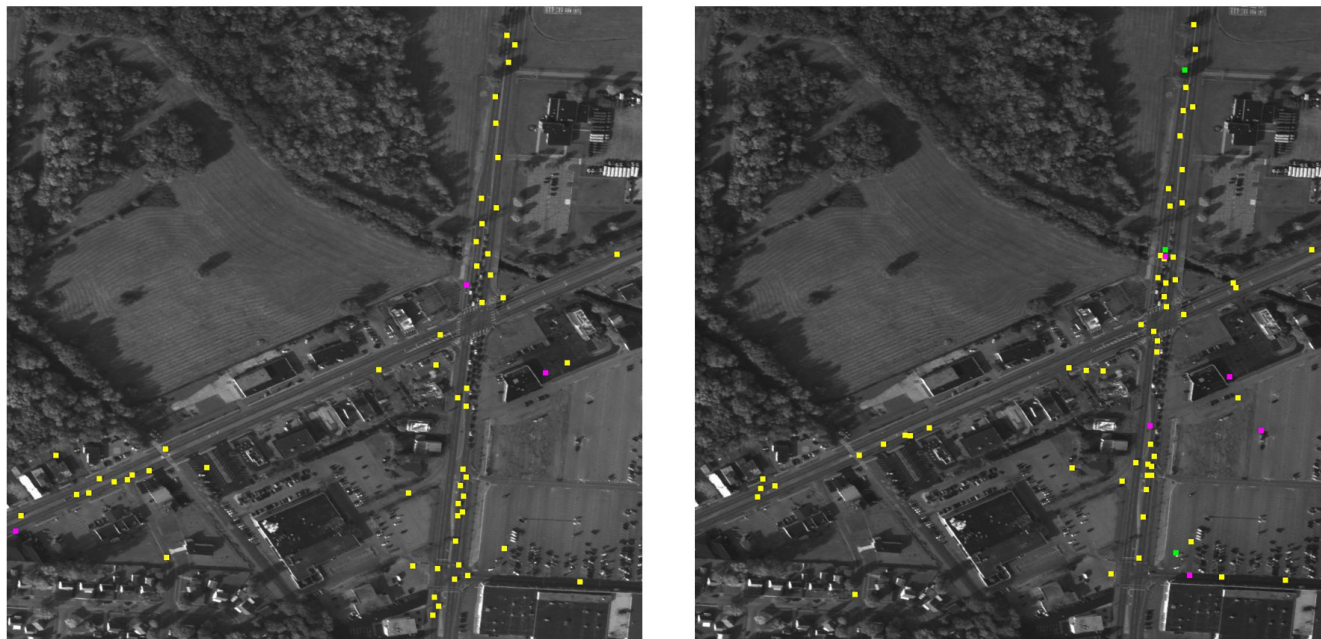


FIGURE 9. Visualization of true positives (yellow), false positives (green), and false negatives (magenta). High-resolution versions can be viewed at <https://sens.medipol.edu.tr/visionai-was/hm-net-detection-visual-results/>.

TABLE 3. The detection performance (F1 scores) comparison table for six AOIs of WPAFB dataset [47].

Method	01	02	03	34	40	41
[30]	0.783	0.793	0.876	0.755	0.749	0.762
[31]	0.738	0.820	0.868	0.761	0.733	0.700
[48]	0.866	0.890	0.900	-	-	-
[27]	0.850	0.876	0.889	0.826	0.817	0.799
[32]	0.743	0.825	0.876	0.763	0.737	0.708
[7]	-	-	-	0.900	0.900	0.930
[1]	0.947	0.951	0.942	0.933	0.983	0.928
[33]	0.935	0.947	0.945	0.953	0.935	0.934
[22]	0.944	0.967	0.964	0.967	0.938	0.948
HM-Net	0.952	0.971	0.972	0.973	0.953	0.950

ground-truth point can only be assigned to one detection (see [33]). Table 3 provides F1 scores for HM-Net and other state-of-the-art object detectors from literature in 6 different tested AOIs. HM-Net provides the highest F1 score in 5 regions and comes in second place for AOI 40. In Fig. 9 locations of true positive, false positive and false negative detections are shown on two sample frames. As discussed in Section III, HM-Net is trained to detect both moving and stationary vehicles, which helps in reducing false moving object detections. Stationary and moving vehicle detections are visualized in Fig.13.

Table 4 provides precision and recall values for HM-Net, [22] and [33]. (For training CenterTrack [22] in WPAFB dataset, bounding box prediction head of the architecture is ignored.) HM-Net is superior to [33] in all tested regions. Precision values for [22] and HM-Net are similar, but HM-Net is superior to [22] in terms of recall.

Table 5 compares the same three methods in terms mean Average Precision (mAP), which corresponds to the area under precision-recall curve as defined by object detection

TABLE 4. Comparison of average precision and recall values of [22] and [33] with our results for the six AOIs.

AOI	Precision			Recall		
	[33]	[22]	HM-Net	[33]	[22]	HM-Net
01	0.939	0.979	0.977	0.914	0.896	0.929
02	0.958	0.981	0.985	0.932	0.950	0.959
03	0.933	0.981	0.987	0.951	0.945	0.958
34	0.956	0.985	0.993	0.941	0.944	0.955
40	0.943	0.982	0.980	0.911	0.877	0.926
41	0.945	0.983	0.984	0.919	0.911	0.920

evaluation procedures [46]. mAP is calculated for two scenarios, by requiring either a maximum of 10 pixels ($d < 10$) or 5 pixels ($d < 5$) distance between matching detected and ground truth centers. Clearly, HM-Net surpasses both [22] and [33] in all AOIs and the performance difference is even more dramatic when $d < 5$ is required. [33] fails to provide precise center coordinates for most of the detected objects. While both HM-Net and CenterTrack [22] produce accurate object centers, HM-Net produces more precise centers due to the use of an extra Precision Head at the network output. The average distance of detected centers from ground truth coordinates is 1.8 pixels. Even though we upscale predictions at the network output, we surpass [1] and [33], which have average distances of 3.8 pixels and roughly 2 pixels, respectively.

Table 6 compares the tracking accuracy of HM-Net against CenterTrack (other methods could not be tested because their tracking codes were not available). mAP for tracking is evaluated as described in [46]. Tracks are sorted based on their length, i.e. longer tracks are assigned higher confidence and matched first. HM-Net provides significantly better track matching accuracy than [22] in all AOIs. Simulation results

TABLE 5. Comparison of mAPs (in percentage) of [22] and [33] with our results for the six AOIs.

AOI	mAP @d < 10			mAP @d < 5		
	[33]	[22]	HM-Net	[33]	[22]	HM-Net
01	86.41	87.72	92.71	54.20	85.34	91.12
02	90.16	93.66	95.91	57.82	92.22	95.15
03	89.22	93.05	96.09	63.72	92.51	95.59
34	90.59	93.37	95.90	59.92	90.64	94.38
40	86.30	86.26	92.41	52.55	83.94	90.80
41	87.35	89.75	93.27	50.08	86.54	91.84

TABLE 6. Comparison of tracking performance for the six AOIs with tracking mAP from [46].

Metrics	Methods	01	02	03	34	40	41
mAP @0.25	[22] HM-Net	78.71 78.39	92.30 90.31	82.66 80.74	87.92 90.86	72.97 79.10	77.64 77.21
mAP @0.50	[22] HM-Net	33.44 47.04	67.22 83.31	63.04 69.77	68.10 82.10	26.26 47.54	53.03 70.38
mAP @0.75	[22] HM-Net	10.57 16.73	42.54 51.96	18.19 31.90	40.56 51.13	6.05 19.89	20.84 42.89
mAP	[22] HM-Net	40.90 47.39	67.35 75.19	54.63 60.80	65.53 74.70	35.10 48.84	50.50 63.49

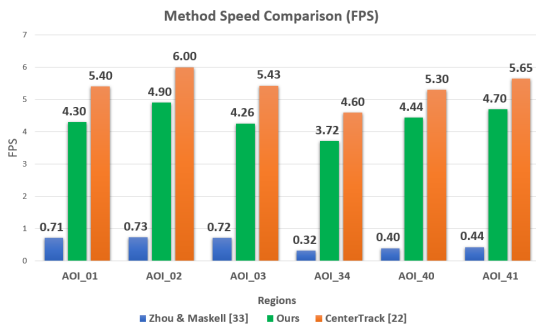


FIGURE 10. Speed (fps) comparison of different methods on AOIs.

also show that HM-Net tracking IoU (Intersection over Union) is significantly better than that of [22]. This means that HM-Net tracks provide on average a longer overlap with the ground truth tracks. This could also be seen by comparing mAP results of both methods at 0.50 and 0.75 IoU, for which HM-Net is far superior.

Fig. 10 provides inference speeds for HM-Net, [22] and [33] in terms of fps for different AOIs on a RTX2080Ti 11GB GPU. HM-Net out-performs [33] in speed by roughly 7 times. Although HM-Net is about 30% slower than [22], better detection and tracking results justify this slightly higher execution times. Since both HM-Net and [22] algorithms are faster than typical WAMI sensor frame rates, both models could perform real-time object detection in a medium resolution WAMI sequence.

In [33], the model utilizes background subtraction followed by CNN-based false positive filtering from cropped detection regions. As a result the model speed in [33] is vulnerable to image size and total target count variations. However, image size variations and tracking target count do not much affect the inference speed of HM-Net, due to its

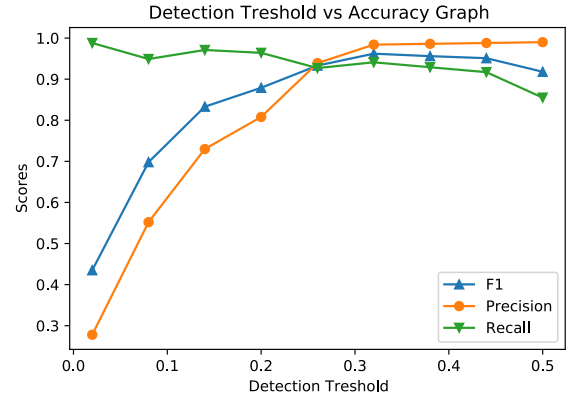


FIGURE 11. Effect of moving vehicle detection threshold (θ_c) on model accuracy.

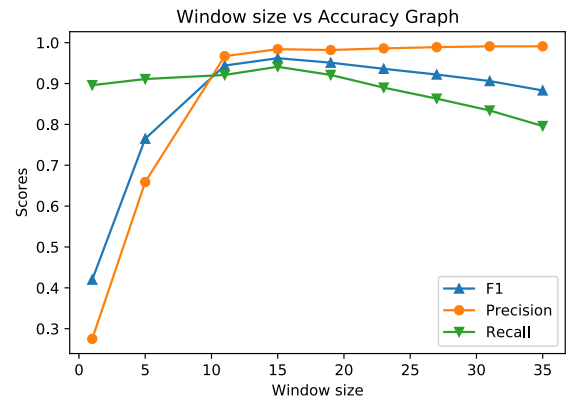


FIGURE 12. Effect of NMS window size on model accuracy.

one-stage network, which processes whole inputs in a single feed-forward pass.

B. ABLATION STUDY

In this section we investigate the effect of several hyper-parameters on the model’s performance for object detection and tracking. In our experiments we have tested various combination of these parameters to see their effects on overall model accuracy. Selected experiment samples will be discussed below to evaluate the effects of detection threshold (θ_c), combination of confidence boosting factor (φ_c) and feedback thresholding (λ_c), non maximum suppression (NMS) window size, and RCP/RCR during training.

Figure 11 shows the detection performance with respect to the detection threshold (θ_c). As seen from the figure, detection threshold (θ_c) and recall scores of the model are inversely correlated. Lower θ_c facilitates overall higher number of detections, resulting in an increase in the number of true positive detections at cost of more false positive detections, thereby decreasing F1 and precision scores. When θ_c is between 0.25 and 0.42 we observe a high accuracy plateau in the figure, which shows that the model performance is robust against changing detection threshold in this range. We have used $\theta_c = 0.32$ in all other experiments,

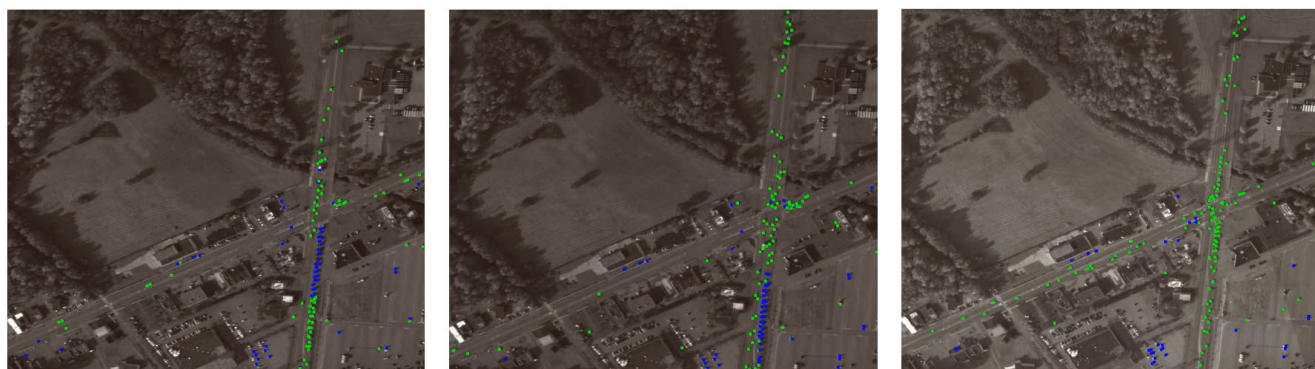


FIGURE 13. Visualization of moving (green) and stationary (blue) objects. Behavior of the vehicles can be observed on a signalized junction.

since it is relatively at the center of the plateau and also yields better scores on validation.

TABLE 7. Effect of feedback filtering threshold (λ_c) and confidence amplification factor (φ_c) values on detection accuracy.

λ_c	φ_c	F1	Precision	Recall
32	1.0	0.950	0.986	0.919
	1.2	0.953	0.998	0.912
28	1.0	0.957	0.974	0.939
	1.2	0.962	0.984	0.941
24	1.0	0.939	0.945	0.933
	1.2	0.948	0.972	0.925

On Table 7 we can observe the effect of different λ_c and φ_c combinations on F1, recall, and precision detection scores, where all other parameters are fixed to their best known values. The use of separate feedback filter and detection thresholds is proposed to be able to represent objects below the detection threshold on feedback heat maps (see Section III). The motivation in keeping those objects on the heat map is to encourage their reappearance as valid object detections in upcoming frames. Note that $\varphi_c = 1$ and $\lambda_c = 0.32$ on Table 7 corresponds to the base case where there is no confidence boosting and no SGR filtering, i.e. $\lambda_c = \theta_c$. Lowering λ_c with respect to θ_c increases recall and reduces precision, yet resulting in a higher overall F1 score. However, as λ_c is further reduced, the performance starts to deteriorate. An optimal value of $\lambda_c = 0.28$ puts on the feedback heatmap additional detections that have the potential of reappearing in upcoming frames, and yet at same time does not increase the number of false positive detections overwhelmingly. Amplifying confidences of detected objects on their feedback representations increases model precision and F1 scores as well. The best combination of parameters is at $\lambda_c = 0.28$ and $\varphi = 1.2$, which corresponds to a 1.2% improvement in F1 score over the base case.

As discussed in Section III-A, NMS window size is crucial for object detection and tracking operations. The primary role of the window is to eliminate adjacent high confidence scores close to the local maxima. However choosing the

optimal window is not a straightforward task. As Figure 5 shows the model does not yield perfect Gaussian shapes on the center location predictions. A large window size ensures the elimination of adjacent high confidence values that are on the same object, but might also eliminate the centers of other close by objects. On the other hand, for small window sizes, nearby high confidence pixels that belong to the same object can be misinterpreted as centers of separate objects. Fig. 12 justifies these discussions. For smaller window sizes we observe high number of false positives and a relatively high recall score. As window size exceeds 11×11 , the model enters a steady high accuracy plateau. However F1 score starts to descend as window size exceeds 15×15 . This negative trend for larger window sizes is a result of a slowing decrease in false positives and fast increase in false negatives. As a result, 15×15 is selected as NMS window size during both detection and tracking operations. Note that this optimal window size is in line with the 10-15 pixel vehicle length in WPAFB dataset.

TABLE 8. Effect of RCP/RCR training methods on detection and tracking accuracy.

AOI	mAP(%) @ $d < 10$		mAP(%) (Tracking)	
	Base Model	HM-Net	Base Model	HM-Net
01	89.93	92.71	44.81	47.39
02	95.33	95.91	74.67	75.19
03	95.35	96.09	60.55	60.80
34	95.41	95.90	73.91	74.70
40	87.98	92.41	45.74	48.84
41	91.94	93.27	61.33	63.49

Table 8 investigates the performance contributions of RCP and RCR methods applied during training of the model. Introduction of RCP and RCR aims to reduce false positive detection count and give model a better simulation of the feedback process during training (see Section III). Effect of RCP and RCR is tested by comparing against a Base Model that is completely identical but trained without RCP and RCR data augmentations. Trained base model is tested under identical testing conditions. Results are shown on Table 8. Introduction of RCP and RCR increases model

success in both detection and tracking. We observe significant improvement in scores, especially for AOI-01 and AOI-40. These AOIs have signalized busy intersections, which causes vehicles to stop and move suddenly. Also intersections can cause problems such as tracking id switches due to sharp changes in vehicle headings. With the help of RCP and RCR in training, the model learns to handle sudden transitions between motion states of the vehicles and alterations in their headings better than plain trained version.

V. CONCLUSION AND FUTURE WORK

In this work, we have proposed a combined detection and tracking method that utilizes center-based object detection and tracking. Our method relies on our one-stage CNN based architecture called HM-Net, which generates heat maps of object center locations and object displacements for detection and tracking tasks. Heat map-based approach facilitates simultaneous detection of high and varying number of objects with fast and stable speed. Our method uses both current and previous frames and filtered previous predictions as input to exploit spatio-temporal information. Novel model training tools and feedback filters address existing problems on the training and inference of heat map-based detectors, and these methods increase overall model performance, while reducing the number of false positive detections.

Our approach is significantly faster than frame differencing and background subtraction-based state-of-the-art WAMI detectors and yields compatible speed with one-stage counterparts. HM-Net yields state-of-the-art results on simulations conducted on WPAFB dataset [5] for detection and tracking of moving vehicles. To improve the overall performance, alterations in model architecture and their implications on both tracking and detection performances will be further examined. Currently HM-Net uses just a single prior frame and single prior heatmap; utilizing a higher number of prior frames and outputs will be investigated to increase overall performance. A complex vehicle motion model, in the form of a Kalman filter, could also be integrated to improve object tracking accuracy. Model will also be improved to work with larger resolution images that cover larger WAMI regions. Also, MOT applications for aerial imagery at different altitudes (i.e. drone videos) of HM-Net architecture will be investigated as future work.

REFERENCES

- [1] R. Lalonde, D. Zhang, and M. Shah, "ClusterNet: Detecting small objects in large scenes by exploiting spatio-temporal information," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 4003–4012.
- [2] T. T. Nguyen, H. Grabner, H. Bischof, and B. Gruber, "On-line boosting for car detection from aerial images," in *Proc. IEEE Int. Conf. Res., Innov. Vis. Future*, Mar. 2007, pp. 87–95.
- [3] R. Ruskone, L. Guigues, S. Airault, and O. Jamet, "Vehicle detection on aerial images: A structural approach," in *Proc. 13th Int. Conf. Pattern Recognit.*, 1996, pp. 900–904.
- [4] T. Zhao and R. Nevatia, "Car detection in low resolution aerial images," *Image Vis. Comput.*, vol. 21, no. 8, pp. 693–703, Aug. 2003.
- [5] *Wright-Patterson Air Force Base (WPAFB) Dataset*, U.S. Air Force Research Laboratory, Wright-Patterson Air Force Base, OH, USA, 2009.
- [6] G. Kopsiaftis and K. Karantzalos, "Vehicle detection and traffic density monitoring from very high resolution satellite video data," in *Proc. IEEE Int. Geosci. Remote Sens. Symp. (IGARSS)*, Jul. 2015, pp. 1881–1884.
- [7] R. Pflugfelder, A. Weissenfeld, and J. Wagner, "On learning vehicle detection in satellite video," 2020, *arXiv:2001.10900*.
- [8] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "SSD: Single shot multibox detector," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*. Cham, Switzerland: Springer, 2016, pp. 21–37.
- [9] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 2, pp. 386–397, Feb. 2020.
- [10] R. Girshick, "Fast R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1440–1448.
- [11] X. Zhou, D. Wang, and P. Krähenbühl, "Objects as points," 2019, *arXiv:1904.07850*.
- [12] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 580–587.
- [13] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 91–99.
- [14] Z. Cai and N. Vasconcelos, "Cascade R-CNN: Delving into high quality object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 6154–6162.
- [15] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 779–788.
- [16] C.-Y. Fu, W. Liu, A. Ranga, A. Tyagi, and A. C. Berg, "DSSD: Deconvolutional single shot detector," 2017, *arXiv:1701.06659*.
- [17] N. Wojke, A. Bewley, and D. Paulus, "Simple online and realtime tracking with a deep association metric," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2017, pp. 3645–3649.
- [18] X. Hou, Y. Wang, and L.-P. Chau, "Vehicle tracking using deep SORT with low confidence track filtering," in *Proc. 16th IEEE Int. Conf. Adv. Video Signal Based Surveill. (AVSS)*, Sep. 2019, pp. 1–6.
- [19] Y. Zhang, C. Wang, X. Wang, W. Zeng, and W. Liu, "FairMOT: On the fairness of detection and re-identification in multiple object tracking," 2020, *arXiv:2004.01888*.
- [20] K. Kang, H. Li, J. Yan, X. Zeng, and B. Yang, "T-CNN: Tubelets with convolutional neural networks for object detection from videos," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 28, no. 10, pp. 2896–2907, Oct. 2018.
- [21] P. Bergmann, T. Meinhardt, and L. Leal-Taixe, "Tracking without bells and whistles," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 941–951.
- [22] X. Zhou, V. Koltun, and P. Krähenbühl, "Tracking objects as points," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*. Cham, Switzerland: Springer, 2020, pp. 474–490.
- [23] S. A. Ahmadi, A. Ghorbanian, and A. Mohammadzadeh, "Moving vehicle detection, tracking and traffic parameter estimation from a satellite video: A perspective on a smarter city," *Int. J. Remote Sens.*, vol. 40, no. 22, pp. 8379–8394, Nov. 2019.
- [24] A. Xu, "Motion detection in satellite video," *J. Remote Sens.*, vol. 6, no. 2, pp. 1–9, 2017.
- [25] T. Yang, X. Wang, B. Yao, J. Li, Y. Zhang, Z. He, and W. Duan, "Small moving vehicle detection in a satellite video of an urban area," *Sensors*, vol. 16, no. 9, p. 1528, 2016.
- [26] X. Zhang and J. Xiang, "Moving object detection in video satellite image based on deep learning," *Proc. SPIE LIDAR Imag. Detection Target Recognit.*, vol. 10605, Nov. 2017, Art. no. 106054H.
- [27] V. Reilly, H. Idrees, and M. Shah, "Detection and tracking of large number of targets in wide area surveillance," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*. Berlin, Germany: Springer, 2010, pp. 186–199.
- [28] H. Li, L. Chen, F. Li, and M. Huang, "Ship detection and tracking method for satellite video based on multiscale saliency and surrounding contrast analysis," *J. Appl. Remote Sens.*, vol. 13, no. 2, 2019, Art. no. 026511.
- [29] W. Ao, Y. Fu, X. Hou, and F. Xu, "Needles in a haystack: Tracking city-scale moving vehicles from continuously moving satellite," *IEEE Trans. Image Process.*, vol. 29, no. 7, pp. 1944–1957, Oct. 2019.
- [30] I. Saleemi and M. Shah, "Multiframe many-many point correspondence for vehicle tracking in high density wide area aerial videos," *Int. J. Comput. Vis.*, vol. 104, no. 2, pp. 198–219, 2013.
- [31] J. Xiao, H. Cheng, H. Sawhney, and F. Han, "Vehicle detection and tracking in wide field-of-view aerial video," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Apr. 2010, pp. 679–684.

- [32] M. Keck, L. Galup, and C. Stauffer, "Real-time tracking of low-resolution vehicles for wide-area persistent surveillance," in *Proc. IEEE Workshop Appl. Comput. Vis. (WACV)*, Jan. 2013, pp. 441–448.
- [33] Y. Zhou and S. Maskell, "Detecting and tracking small moving objects in wide area motion imagery (WAMI) using convolutional neural networks (CNNs)," in *Proc. IEEE Int. Conf. Inf. Fusion (FUSION)*, Jun. 2019, pp. 1–8.
- [34] S. Ji, W. Xu, M. Yang, and K. Yu, "3D convolutional neural networks for human action recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 1, pp. 221–231, Jan. 2013.
- [35] K. Simonyan and A. Zisserman, "Two-stream convolutional networks for action recognition in videos," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 568–576.
- [36] M. Baccouche, F. Mamalet, C. Wolf, C. Garcia, and A. Baskurt, "Sequential deep learning for human action recognition," in *Proc. Int. Workshop Hum. Behav. Understand.* Berlin, Germany: Springer, 2011, pp. 29–39.
- [37] K. Kang, W. Ouyang, H. Li, and X. Wang, "Object detection from video tubelets with convolutional neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 817–825.
- [38] X. Li, Z. Jie, W. Wang, C. Liu, J. Yang, X. Shen, Z. Lin, Q. Chen, S. Yan, and J. Feng, "FoveaNet: Perspective-aware urban scene parsing," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2017, pp. 784–792.
- [39] K. Duan, S. Bai, L. Xie, H. Qi, Q. Huang, and Q. Tian, "CenterNet: Keypoint triplets for object detection," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 6569–6578.
- [40] H. Law and J. Deng, "Cornernet: Detecting objects as paired keypoints," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 734–750.
- [41] G. A. Mills-Tettey, A. Stentz, and M. B. Dias, "The dynamic Hungarian algorithm for the assignment problem with changing costs," Robotics Inst., Pittsburgh, PA, USA, Tech. Rep. CMU-RI-TR-07-27, 2007.
- [42] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Proc. Int. Conf. Med. Image Comput. Comput.-Assist. Intervent.* Cham, Switzerland: Springer, 2015, pp. 234–241.
- [43] A. Newell, K. Yang, and J. Deng, "Stacked hourglass networks for human pose estimation," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*. Springer-Verlag, 2016, pp. 483–499.
- [44] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.
- [45] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollar, "Focal loss for dense object detection," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2980–2988.
- [46] P. Zhu, L. Wen, D. Du, X. Bian, Q. Hu, and H. Ling, "Vision meets drones: Past, present and future," 2020, *arXiv:2001.06303*.
- [47] H. Pirsivavash, D. Ramanan, and C. C. Fowlkes, "Globally-optimal greedy algorithms for tracking a variable number of objects," in *Proc. CVPR*, Jun. 2011, pp. 1201–1208.
- [48] L. W. Sommer, M. Teutsch, T. Schuchert, and J. Beyerer, "A survey on moving object detection for wide area motion imagery," in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, Oct. 2016, pp. 1–9.



HAKKI MOTORCU (Student Member, IEEE) was born in Antalya, Turkey, in 1999. He is currently pursuing the B.S. degree in electrical and electronics engineering with Ozyegin University, Istanbul, Turkey. His current research interests include deep learning, machine learning, computer vision, signal processing, image processing, and autonomous systems.



HASAN F. ATES (Senior Member, IEEE) received the Ph.D. degree from the Department of Electrical Engineering, Princeton University, in 2004. He was a Research Associate with Sabanci University, between 2004 and 2005. He held positions of an Assistant Professorship, an Associate Professorship, and a Full Professorship with Isik University, between 2005 and 2018. He is currently a Professor with the Department of Computer Engineering, Istanbul Medipol University, which he joined in September 2018. He is the author/coauthor of more than 50 peer-reviewed publications in the areas of image/video processing/coding and computer vision.



H. FATIH UGURDAG (Senior Member, IEEE) received the B.S. degree from Bogazici University, Istanbul, Turkey, in 1986, with double major in EE and physics, and the M.S. and Ph.D. degrees in EE from Case Western Reserve University, Cleveland, OH, USA, in 1989 and 1995, respectively. He is currently a Full Professor with Ozyegin University, Istanbul. He is also a long-time IEEE Volunteer and also the Chair of the IEEE Turkey Section. He worked full-time in the industry for 13 years before joining academia, in 2004. His tenure in the industry was mostly in Silicon Valley and spanned companies, such as GE, GM, Lucent, Juniper, and Nvidia. His research interests include ASIC/SoC/FPGA implementation of various compute-intensive algorithms, design automation, and computer arithmetic.



BAHADIR K. GUNTURK received the B.S. degree in electrical engineering from Bilkent University, Turkey, and the Ph.D. degree in electrical engineering from the Georgia Institute of Technology, in 1999 and 2003, respectively. Between 2003 and 2014, he was with the Department of Electrical and Computer Engineering, Louisiana State University. Since 2014, he has been with Istanbul Medipol University, where he is currently a Professor. He has published more than 50 peer-reviewed journal/conference papers in the areas of image processing and computer vision.

...