Siamese Cross-Domain Tracker Design for Seamless Tracking of Targets in RGB and Thermal Videos

Chandrakanth V., V. S. N. Murthy, and Sumohana S. Channappayya, *Senior Member, IEEE*

***Abstract*—Multimodal (RGB and thermal) applications are swiftly gaining importance in the computer vision community with advancements in self-driving cars, robotics, Internet of Things, and surveillance applications. Both the modalities have complementary performance depending on illumination constraints. Hence, a judicious combination of both modalities will result in robust RGBT systems capable of all-day all-weather applications. Several studies have been proposed in the literature for integrating the multimodal sensor data for object tracking applications. Most of the proposed networks try to delineate the information into modality-specific and modality shared features and attempt to exploit the modality shared features in enhancing the modality specific information. In this work, we propose a novel perspective to this problem using a Siamese inspired network architecture. We design a custom Siamese cross-domain tracker architecture and fuse it with a mean shift tracker to drastically reduce the computational complexity. We also propose a constant false alarm rate inspired coasting architecture to cater for real-time track loss scenarios. The proposed method presents a complete and robust solution for object tracking across domains with seamless track handover for all-day all-weather operation. The algorithm is successfully implemented on a Jetson-Nano, the smallest graphics processing unit (GPU) board offered by NVIDIA Corporation.**

***Impact Statement*—Surveillance and tracking systems form an integral part of our society today. However, most of the systems are deployed in a non-real-time fashion where data is saved and processed later. Typical examples include traffic cameras, surveillance systems etc. Deep learning (DL) networks have paved the way to automate this application with minimal manual intervention. They work with RGB and thermal data independently to cater for all-day all-weather operation. However, with the architecture presented in this paper, we have demonstrated that a custom-designed single Siamese network can handle cross-domain data effectively and can operate continuously day and night. With automated day and night detection, the operator can be alerted based on the trained urgency minimizing the reaction time and operational costs. With the proposed network we are looking at a potential overhaul of the existing manual surveillance systems to automated Artificial Intelligence (AI) based systems.**

***Index Terms*—Convolutional neural network (CNN), domain translation, generative adversarial network (GAN), mean-shift algorithm, Siamese networks, target tracking.**

I. Introduction

Target tracking is the task of predicting and correlating the range of the objects of interest in consecutive scans of the scene. This is a well-established application in the radar literature, where state-space models are successfully employed for tracking the objects of interest in the field. Recently, this problem has received widespread attention in the computer vision community with the evolution of advanced driver assistance systems, robots, and automated surveillance systems. In imaging-based systems, the tracking problem is translated to localizing the object of interest with a bounding box (BB) in the input frame. Several studies have been reported on image tracking in the RGB domain [1]–[4]. While RGB systems have high spatial resolution and clean contours of objects amidst the background, their performance largely depends on the illumination of the scene. Hence, RGB systems are not suitable for application in low illumination conditions. Infrared systems (IR), on the other hand, work with the surface temperatures of objects and can ideally be applied in all-day and all-weather scenarios. However, they suffer from low spatial resolution and blurry edges, sensitivity to temperature variations, and high cost. Even with these limitations, detection in thermal imagery is becoming increasingly important today for ensuring round the clock capability because visible light is suboptimal in extreme weather situations (e.g., fog, heavy rain, etc.) and at night [5]. With the proliferation of Internet of Things, security-related applications, and the requirement of large-scale production, the cost per unit of the thermal sensor has significantly reduced, making them economical for commercial applications.

These dual-sensor (RGB and thermal) systems are termed as RGBT systems in the literature. In recent years, several studies have been carried out to prove that integrating data from RGB and thermal modalities can effectively improve tracking performance. They can reinforce each other and provide complementary information to promote the robustness and adaptability of trackers [6]. While several methods have been proposed for RGBT tracking [7]–[10], the core problem of data correlation across domains for effective tracker design is still an open challenge. Existing methods largely concentrate on extracting domain-specific and domain-independent features for cross-domain data and fuse the information to obtain improved tracking performance. In this work, we address the problem of surveillance and tracking of targets in RGB and thermal data for continuous and real-time operation with a Siamese network inspired architecture termed as Siamese cross-domain tracker

Manuscript received 25 August 2021; revised 7 November 2021, 10 January 2022, and 30 January 2022; accepted 8 February 2022. Date of publication 14 February 2022; date of current version 20 January 2023. This article was recommended for publication by Associate Editor Simon Yang upon evaluation of the reviewers' comments. *(Corresponding author: Chandrakanth V.)*

Chandrakanth V. and V. S. N. Murthy are with the Defense Research and Development Laboratory (DRDL), Defense Research and Development Organization (DRDO), Hyderabad 500058, India (e-mail: ee17resch01001@iith.ac.in; vsn1097@gmail.com).

Sumohana S. Channappayya is with the Department of Electrical Engineering, Indian Institute of Technology Hyderabad, Kandi 502285, India (e-mail: sumohana@ee.iith.ac.in).

Digital Object Identifier 10.1109/TAI.2022.3151307

(Siam-CDT). The proposed approach is inspired by the work done by Iwashita *et al.* [11]. In [11], they tried to establish a relation between RGB and thermal data empirically by correlating reflection, absorption, and emission coefficients of incident light on the object. We make a similar assumption and pose this problem in the supervised setting of the domain translation framework.

GANs [12]–[15] are extensively applied in domain translation problems for computer vision applications. The problem is studied in both the supervised (ST) and unsupervised (UST) settings. In both supervised and unsupervised settings, domain translation is seldom the final step. The transformed images are further processed to extract information for the final application. For example, when trying to identify a person whose appearance has changed over time, we try to generate images with different styles (e.g., hair, beard, glasses, etc.) using GANs. However, the final application is the correlation of the images with the original dataset for identifying the person. This task can be greatly simplified by considering the final application *a priori* and then trying to extract the information from the intermediate layer. For the above-mentioned application, we can try to extract a common feature vector for all the possible variations (or styles) and compare it with the original feature vector from the ground truth data. In this work, we attempt to converge the cross-domain information onto the assumed shared latent space and directly integrate the application in this space. The results obtained with "Siam-CDT" assert the existence of shared space for supervised networks as claimed in [11]. The proposed "Siam-CDT" network is independent of the input domains and can be applied to multiple image palettes in a supervised setting.

The rest of this article is organized as follows. Section II presents related work and Section III discusses the proposed methodology. Section IV explains the results followed by concluding remarks in Section V.

II. Related Work

The proposed method combines three different techniques to realize the tracking application. We present a brief literature survey of all the relevant techniques followed by a review of existing literature on RGBT systems.

A. Domain Translation

As mentioned earlier, domain translation is studied in two modes: 1) supervised translation (ST) mode and 2) unsupervised translation (UST) mode. In ST, example image pairs (X, Y) are available. For each image $x_i \in X$ in the source domain, there is a corresponding $y_i \in Y$ in the target domain, and we wish to find a mapping $G : X \to Y$ such that $G(x_i) \approx y_i$. Some of the works for ST are given in [16] and [17] and the more general Pix2Pix [18]. However, a major drawback with ST is the lack of labeled training data in the source and target domains. UST methods attempt to address this problem by trying to find a mapping function between independent source and target domains without any pairing information. Some of the successful recent approaches include UNIT [19], CycleGANs [15], CoGAN [14], and Disco-GAN [12]. GANs are very effective tools for generative modeling of images; however, they operate under restrictive assumptions that question the efficacy of translations. Recently, semisupervised translation has been proposed where labeled datasets in the source domain are available which will be paired with unlabeled target domain datasets [20], [21]. The final application is implemented using the domain translated data. In this work, we propose a novel method of integrating the final application in the shared latent space and thereby avoiding complete domain translation.

B. Siamese Networks

In general, convolutional neural networks (CNNs) require large training datasets to train the network for successful deployment. However, with every new input or task, the network has to be retrained to accommodate this new information. To address this problem, Koch *et al.* [22] proposed a Siamese inspired CNN for one-shot image identification by employing a distinct way to triage inputs depending on their similarity. Siamese networks have already been successfully applied in dimensionality reduction [23], face verification [24], and signature verification [25] problems. Zhang *et al.* [26] proposed SiamFT and [27] DSiamMFT for an RGBT application. They proposed two parallel Siamese network architectures for each modality and used the cropped ground truth image as a reference in both modalities for training the Siamese network. Finally, the output information from both modalities is fused to get an improved output. Peng *et al.* [28] present SiamIVFN for fusing RGB and thermal data using two subnetworks, complementary feature fusion network, and contribution aggregation network. They progressively couple filter coefficients throughout the network and finally fuse the original and processed feature vectors to derive the output tracking vector. In this work, we propose a single channel Siamese network with shared weights trained to converge cross-modality data onto a common shared space, thus significantly reducing the computational complexity for cross-domain object tracking.

C. Video Tracking

Video analysis captures the temporal variations in the scene for understanding the target and environmental dynamics. The analysis can be specifically focused on selected regions with targets for designing relevant applications like motion detection and target tracking [29]. Motion-based approaches can be divided into two main categories: 1) background subtraction [30], [31] and 2) optical flow [32], [33]. They are used in surveillance applications to alert the user against unauthorized movements. Video tracking, on the other hand, is the process of tracking the object of interest continuously to extract target parameters. Video tracking methods can be broadly categorized as 1) classical video tracking for single modality data, 2) artificial intelligence (AI) based video tracking for single modality data, and (3) AI-based RGBT video tracking.

1) Classical Video Tracking: In the classical video tracking problem, the position of the target in the first frame is known. From this information, a technique has to be designed to automatically detect the target in subsequent frames of the video.

This seemingly simple problem becomes extremely challenging because of practical issues such as the orientation of the target, occlusions, movement of the target, and related scale variations. Mean-shift tracking algorithms [34] use the Centroid of the image as a reference parameter to predict the position of the target in the next frame. This method fails whenever there is target maneuver (scaling), target orientation change, ambient illumination change, strong background clutter, or occlusion. Also, this method relies only on the pixel data and not on the features of the target which makes the algorithm unreliable. To take the features of the target into account for detection, correlation processing or template matching was proposed. This method uses a reference template or appearance model of the target which is compared with the input data in a sliding window fashion storing the correlation coefficient for each pixel shift in a vector. If the correlation coefficient exceeds a preset threshold, the target presence is declared. Several variations of this method are proposed in the literature [35]–[37]. Among these, discriminative correlation filters (DCFs) have proven successful on benchmark tracking datasets [38], [39]. However, DCF-based approaches [40], [41] suffer from boundary overlap because of the circular convolution property of fast Fourier transform causing data corruption. To overcome this, Danelljan *et al.* [36] proposed spatially regularized DCF (SRDCF). Even with SRDCF, the above methods are still vulnerable to scaling, illumination changes, and occlusions. Lowe [42] addressed some of these issues by proposing the scale invariant feature transform. However, better performance was reported using histogram-based feature vectors for tracking. Feature representations such as histogram of gradients [43] and deformable parts model [44] have been extremely successful in tracking of pedestrians. All the methods mentioned above use handcrafted features for the detection and tracking of the target.

2) AI-Based Video Tracking for Single Domain Data: Deep learning ushered a new era in image and video processing pushing toward automated detection, classification, and localization of targets. However, deep learning networks, by design, work on the principle of frame-independent processing. For a CNN, every frame is a new input and the network has to process every image to detect the presence of a target. However, to track the target, information from past data is necessary. Long short-term memory [45] and gated recurrent unit [46] propose two different methods to propagate past information to the current scan. They are successfully applied in time series based data analysis. Some of the other works published on deep learning based tracking are as follows. SiamRPN [37] uses image correlation to track objects of interest in the video. Danelljan *et al.* [47] proposed ATOM network which discriminates between target and background and provides improved performance over SiamRPN. Danelljan *et al.* [36] also present a variant of DCF using CNNs for object tracking. They used activations from convolutional layers to train the DCF and achieved good results. Milan *et al.* [48] proposed recurrent neural network (RNN) based multitarget tracking where the RNN learns the target models in the field and uses the model to predict target trajectories in unseen data.

3) AI-Based RGBT Video Tracking: We briefly review some of the works published on RGBT tracking next. Xu *et al.* [6] proposed CBPnet with channel attention, bilinear pooling, and quality-aware fusion modules and evaluated the performance of the network on GTOT [49] and custom RGBT234 [50] datasets. Li *et al.* [51] proposed a challenge-aware RGBT tracker where they design two parallel CNN networks for each modality and divide the processing challenges into modality-specific and modality shared components. They also propose sharing of the information across domains to enhance the performance of single modality networks with this additional information. In the final stage, they adaptively aggregate the information from both networks to realize a robust tracker. Zhu *et al.* [52] proposed TFNet with a feature aggregation block which combines feature vector outputs from multiple modalities followed by a feature pruning block to remove redundant features to reduce overfitting and a feature fusion module to integrate the information from individual modalities with past aggregated module data for accurate classification.

The RGBT trackers proposed above have considered the problems of scaling and illumination changes but ignored the case of occlusion and track loss scenarios. “Siam-CDT” network addresses these problems using the coasting technique widely used in radar literature. Coasting is the process of predicting the probable location of the target in the next frame from previous state information for a few predefined scans trying to reacquire the target. Two possibilities can occur during coasting: 1) the object changes its trajectory while passing through the occlusion or during the time taken for realigning the source; 2) the current trajectory is maintained and the object is reacquired after n frames. In 1), the track vector will be flushed out after n scans and the algorithm reinitiates the search for the target in incoming frames to establish a new track, and, in 2), the target will fall in the predicted coasting window after n scans and the track is continued. We address both the scenarios in this work and our results demonstrate the efficacy of the “Siam-CDT” in cross-domain object tracking applications. All the networks discussed above used two parallel Siamese channels, CNN1 (RGB) and CNN2 (thermal), for each mode of processing. In this work, we propose a novel approach using single-channel processing by fusing the final application in the shared latent space.

III. Proposed Method

In this work, we propose the fusion of a mean-shift tracker (MST) with a custom Siamese network (Siam-CDT) to realize a computationally efficient architecture for seamless track handover across domains. The proposed architecture is divided into three stages as enumerated below.

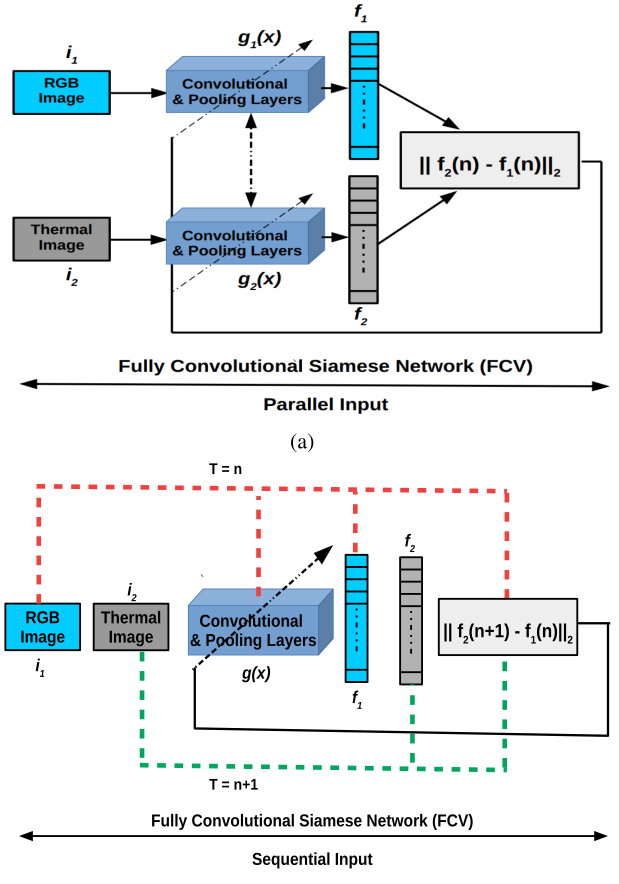
1) Design of a fully convolutional Siamese CNN (FCV) to extract common features for multidomain data in the assumed shared latent space.
2) Design of a fully connected network (FCN) to integrate the tracking application.
3) Fusion of an MST with “Siam-CDT” to design a computationally efficient robust tracker capable of tracking, coasting, and target reacquisition.

A. Siamese FCV Design

Siamese networks are a class of CNNs proposed to work with limited datasets for probabilistic data association. Apart from the applications discussed above, some literature has also reported Siamese network implementation for the object tracking problem. We briefly discuss some of the works for comparison with the proposed method. He *et al.* [55] proposed the SA-SIAM network derived from SiamFC [56] for real-time object tracking. They proposed a combination of appearance features trained on the similarity learning problem and semantic features trained on the image classification problem for tracking the target. Dong *et al.* [57] proposed a novel triplet loss for the Siamese framework to extract robust features for object tracking. Wang *et al.* [58] proposed RASNet tracker for online target tracking and validated the performance on object tracking benchmark (OTB) and visual object tracking (VOT) datasets.

The RGB and thermal domains capture information in completely different and unrelated ways. Theoretically, we cannot establish a correlation between RGB and thermal information derived from the same object. Iwashita *et al.* [11] present a logical explanation for the probable existence of a shared latent space for RGB and thermal images for the interested reader.

Fig. 1 shows the proposed Siamese FCV architecture. First, both the visual and thermal sources are aligned to view the same space for getting supervised inputs into the Siamese network. The network attempts to converge these inputs from both the modalities to a common feature vector representation which is propagated to FCN for integrating the final application. During training, the cost function converged to zero, resulting in the vanishing gradient problem during back-propagation. To overcome this, a small perturbation around zero (ϵ) is introduced into the desired vector input. The convergence of feature vectors (f_1, f_2) in Fig. 1 corroborates this assumption of a shared latent space proposed in [11]. We considered two input configurations for the FCV network as shown in Fig. 1(a) parallel RGBT input and Fig. 1(b) alternating sequential input. For parallel configurations, the output of both RGB and thermal input is available simultaneously. So the difference vector is calculated and the network is trained to map the difference vector to an ϵ-perturbed zero vector. In the sequential configuration, the difference vector is calculated for every alternate cycle which is back-propagated for updating the network. We formulated consistency checks in both configurations to ensure that the generated feature vector is a weighted combination of individual inputs based on illumination constraints.


i1
RGB Image
g1(x)
Convolutional & Pooling Layers
f1
|| f2(n) - f1(n)||2
Thermal Image
i2
Convolutional & Pooling Layers
g2(x)
f2
Fully Convolutional Siamese Network (FCV)
Parallel Input
T = n
f2
i2
RGB Image
Thermal Image
Convolutional & Pooling Layers
|| f2(n+1) - f1(n)||2
i1
g(x)
f1
T = n+1
Fully Convolutional Siamese Network (FCV)
Sequential Input

(a)

(b)

Fig. 1. Block diagram of a fully convolutional Siamese network trained with l_2 loss function for optimization to encode RGB and thermal images in a shared latent feature vector space. (I_1, I_2) are the input images from different domains. Transformation functions $g_1(x), g_2(x)$ map (I_1, I_2) to feature vectors (f_1, f_2) $[g_1(i_1) \rightarrow f_1, \quad g_2(i_2) \rightarrow f_2]$. (a) and (b) Two input configurations considered for FCV design.

B. Fully Connected Neural Network

The output of the FCV network now represents a unified representation for RGB and thermal data. This unified representation is used to train a fully connected neural network (FCN) to predict the BB coordinates of the target of interest in the input image. The loss function considered for this regression problem is the Euclidean distance between the predicted and ground truth BB coordinates as given in the following equation:

$$R_{\text{loss}} = ||\text{BB}_{\text{GT}} - \text{BB}_{\text{Pr}}||_2 \tag{1}$$

where R_{loss} is the regression loss used as a cost function, $\text{BB}_{\text{GT}} = (x_g, y_g, w_g, h_g)$ is the ground truth BB, and $\text{BB}_{\text{Pr}} = (x_p, y_p, w_p, h_p)$ is the BB predicted by the network.

With standard CNN architectures, the network could not converge because of the large variations in the statistics of the data across the domains. To understand the variation in cross-domain data, we quantified the difference information using mean square error (mse) and the structural similarity (SSIM) index [63]. The parameters are shown in Table I for paired RGB and thermal images from KAIST [64] and VOT [53] datasets. After the FCV-FCN network is trained for BB predictions, the target tracker is designed for trajectory estimation and tracking as explained in the next section.

C. Tracking by Fusion of CNN and Mean-Shift Algorithm

Tracking a target in the RF domain is accomplished using interacting multiple model filters [65] and their variants that use precise positional information obtained from an active transmit

TABLE I
METRICS FOR CROSS-DOMAIN IMAGE COMPARISON

Frame No.	Target	MSE	SSIM
1	Person on Bike*	30320	0.48
150	Person on bike*	32530	0.51
521	Person on bike*	40635	0.35
1	Car†	14572	0.54
71	Car†	8008	0.55
432	Car†	8531	0.46

*VOT 2020 Dataset
†KAIST Dataset
MSE ≈ 0, SSIM $= 1$

Algorithm 1: Proposed Algorithm for Target Tracking.

```
(a) Read the input image and pass it through CNN
    channel to obtain initial bounding box (BB)
    coordinates $(x_1, y_1, w_1, h_1)$ to start the track.
(b) Pass $(x_1, y_1, w_1, h_1)$ to mean-shift tracker channel
    along with subsequent input images for tracking the
    target. Control CNN disable flag $CNN_{flag} = 0$ with
    a preset counter, track loss flag and log the
    performance of mean-shift tracker.
(c) Pass the output of mean-shift tracker $(x_i, y_i, w_i, h_i)$
    through $k_{th}$ order delay filter bank working in Last In
    First Out Mode (LIFO).
(d) if $\left\|\left(MST_k - MST_{k-1}\right)\right\|_2 < \epsilon$ then
    $BB_n = BB_{n-1}$ (LIFO)
    else
        -Declare track loss
        -Predict coasting BB coordinates
        $CBB_n = \sum_{k=n-p}^{n-1} BB_{n-1}$
        if Coasting Counter $>$ n (predefined value)
          then
            -Declare Target Lost
            -Invoke CNN channel to reacquire the
             target
        end
    end
end
(e) if input image count (k) $>$ p (predefined value)
    then
    -Pass input image through CNN channel for
     bounding box prediction
    -update search window co-ordinate input for
     mean-shift tracker
end
(f) In case when output is generated in CNN channel
    and tracking filter. CNN channel output is given high
    priority to prevent deadlock.
```

TABLE II
COMPARISON OF MODEL PARAMETERS FOR STANDARD NETWORKS

CNN	**Squeezenet** [59]	**Mobilenet V2** [60]	**Tiny VGG** [61]	**Tiny Yolo** [62]
Trainable Parameters	421098	3470000	308220	25167720

signal. In video-based tracking, no such reference is available for range measurement. However, the video allows us to spatially localize the target and track the same by storing the information from previous frames. Classical autonomous tracking algorithms, namely mean-shift algorithm [66] and correlation algorithms [67], use Centroid (2)–(3) and target templates as reference parameters for target tracking. However, classical target tracking methods have a few problems: 1) In Centroid processing, initial BB coordinates have to be specified which requires a human in the loop to start the track, and 2) if the track is lost due to target passing behind occlusions, illumination changes, locking to false alarms, etc., it cannot be resuscitated, and 3) it is not possible to store templates for all possible scenarios and correlate in real time. All these problems are successfully addressed in CNN-based target tracking. However, a major disadvantage of CNNs is their computational complexity. Every forward pass of CNN for BB prediction requires a large number of computations. Table II presents the parameters for standard CNNs and “Tiny VGG”[61].

At a frame rate of 30 fps, a typical target moves very slowly between successive frames and the computation of all the CNN parameters for every single forward pass is redundant. In cases where the target is stationary for some frames, these computations can be completely avoided. As empirical evidence, we observed that, on average, the MST was predicting erroneous BB coordinates every 60 frames where a correction from the CNN channel was required. This number varies with the quality of data and the speed of the target. So, we predicted the BB coordinates from the CNN channel every 60th frame. A reasonable assumption of the search window space for the MST is of the order of 60 × 60 pixels (these numbers depend on the distance of the target from the source). The number of computations required for calculating the Centroid of the image of this size is “7200.” Therefore, by computing the CNN output every 60 frames, we significantly reduce the number of computations. The saving is directly proportional to the size of the network. Therefore, to make the system power efficient, we propose a fusion of the mean-shift algorithm with the CNN-based object detector. The CNN is used for initial target detection to avoid manual intervention, and, in subsequent scans, the track is continued using the mean-shift algorithm. However, since the mean-shift algorithm is susceptible to changes in orientation, illumination, scale variation, and occlusions, it occasionally locks on to false targets, from which it cannot recover. To solve this problem, we propose periodic CNN computation to correct the offsets in the mean-shift algorithm and to provide reference data for trajectory correction. Fig. 2 shows the complete architecture for the proposed method. The CNN channel is connected to the mean-shift channel to provide initial BB coordinates for MST to start the track. In subsequent frames, the input image is passed intermittently through the CNN channel indexed by counter k in Fig. 2. p is inversely proportional to the speed of targets in the field. For every p ($p = 60$ in our case) iteration, the input frame is passed once through the CNN channel for offset correction. From the initial BB input from the CNN channel,

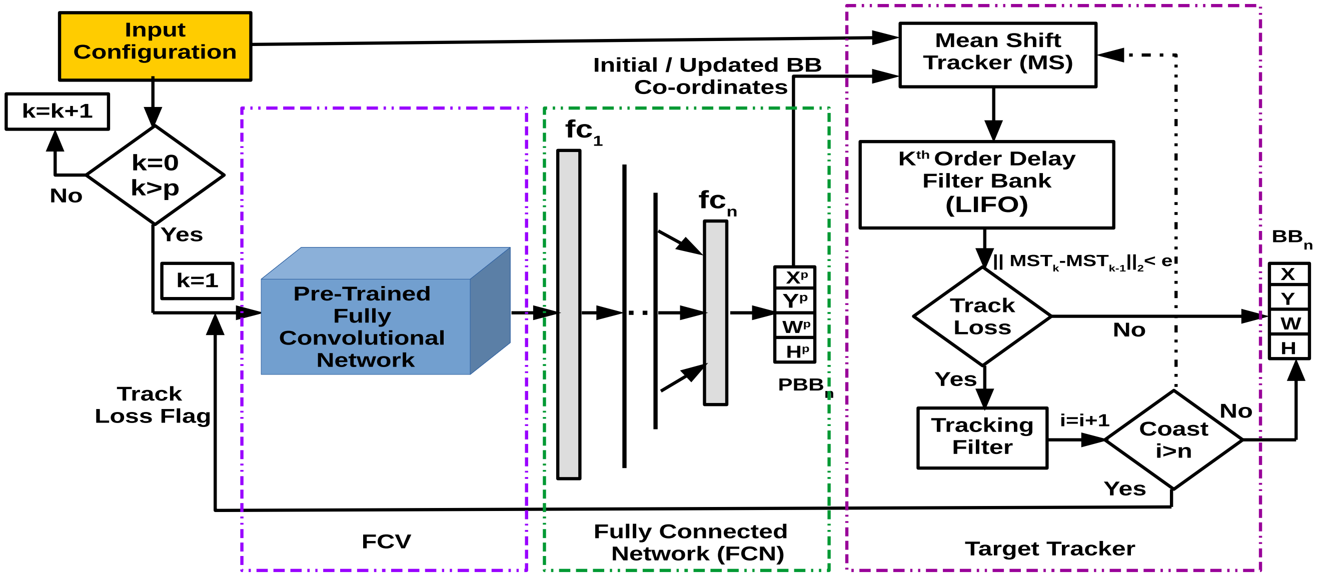
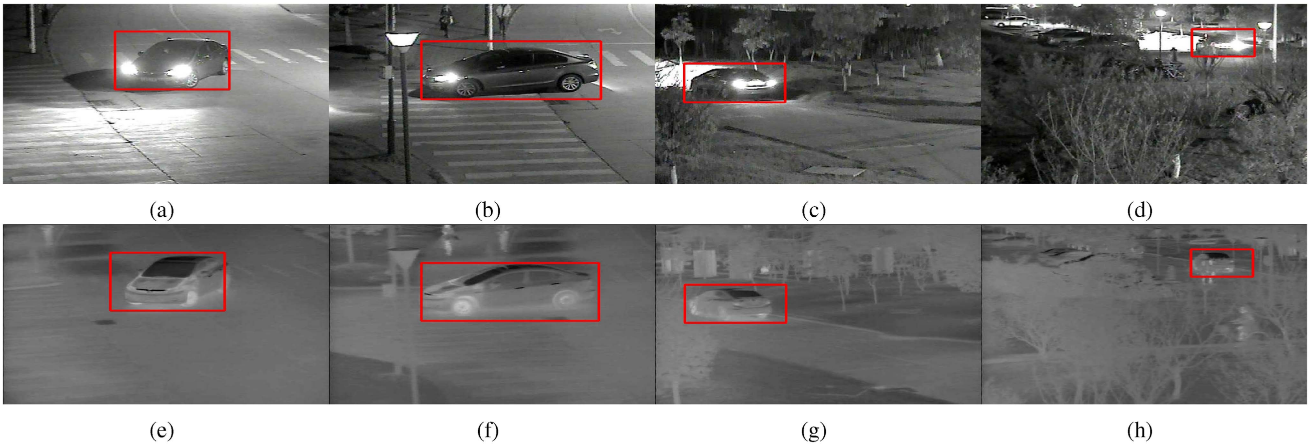


Fig. 2. Block diagram of the proposed architecture for computationally efficient fusion of CNN with mean-shift tracker. The CNN predicts the initial BB coordinates and passes them to the mean-shift tracker to start the track. The CNN is invoked at fixed intervals and under track loss scenarios to reduce computational load. The output of the mean-shift tracker is stored in the kth-order delay filter bank operating in LIFO mode. In the event of track loss, the output of the tracking filter predicts the coasting window coordinates for n frames. If the target is not reacquired, track loss is declared invoking the CNN to search and reacquire the target in incoming frames. This method addresses the problem of track loss due to occlusions in the field and target maneuvers.


(a) (b) (c) (d)

(e) (f) (g) (h)

Fig. 3. Tracking results for VOT2020 dataset [53] with moving car target in sequential frames of RGB and thermal videos. (a)–(d) Output of the network for RGB images. (e)–(h) Output of the same CNN for thermal images. In both the cases, track is consistent as observed.

the MST predicts the subsequent search space by extrapolating the Centroid data computed from (2) and (3), where M is the moment (zeroth and first-order) of the image $I(x, y)$.

There are two possibilities for track loss with this architecture: 1) the first case corresponds to the scenario that the predicted BB by the MST lies outside the frame size (i.e., the case of a target moving out of search space BB coordinates) and 2) the l_2 distance between the adjacent BB exceeds a predefined threshold ϵ as shown in (4). In both cases, track loss is declared and the tracking filter predicts the coasting BB for n iterations, where n is a predefined constant. During this phase, the search window for MST is updated by the coasting BB coordinates. If the adjacent BB predictions do not converge for n ($n = 5$ in our case) scans, track loss is declared invoking the CNN channel for updated BB. On rare occasions when the output is generated by the tracking filter and the CNN channel simultaneously, the CNN channel output holds priority to prevent deadlock

$$C_x = \frac{M_{10}}{M_{00}} \quad \text{and} \quad C_y = \frac{M_{01}}{M_{00}} \tag{2}$$

$$M_{ij} = \sum_x \sum_y x^i y^j I(x, y) \tag{3}$$

$$\left\| \left(\text{MST}_k - \text{MST}_{k-1} \right) \right\|_2 < \epsilon \tag{4}$$

where MST $=$ output of the MST (x, y, w, h), $k =$ time stamp, (C_x, C_y) are the Centroid coordinates of the target, and ϵ is a small value close to zero.

In the event of track loss, the output of the MST is passed through a unit delay filter bank working in last-in first-out

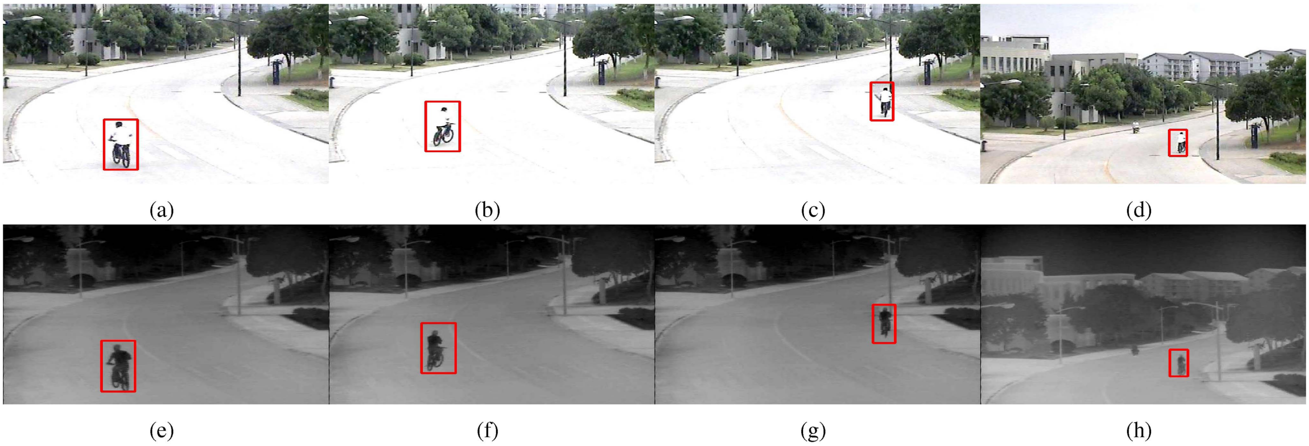


Fig. 4. Tracking results for VOT2020 dataset [53] with person on moving bike target in sequential frames of RGB and thermal videos. (a)–(d) Output of the network for RGB images. (e)–(h) Output of the same network for thermal images.

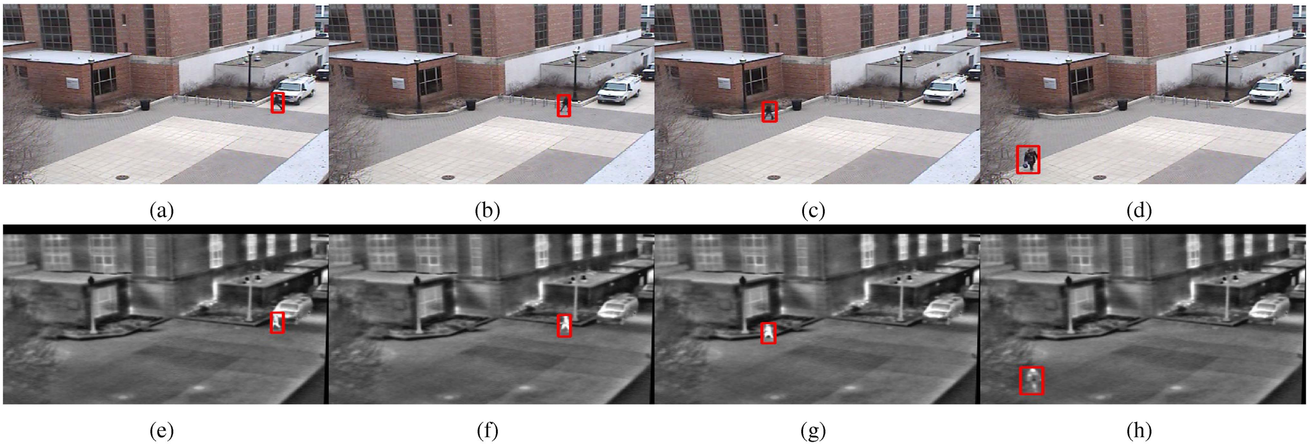


Fig. 5. Tracking results for OSU dataset [54] with pedestrian target in RGB and thermal frames. (a)–(d) Output for RGB images. (e)–(h) Output of the network for thermal images.

(LIFO) mode storing the information from previous k images. The stored information is accessed by the tracking filter to output filtered BB. We designed a tracking filter inspired by the cell averaging constant false alarm rate (CA-CFAR) [68] technique from radar literature for the proposed application. After training the complete network, the network is tested using a random mix of RGB and thermal data for multiple datasets. The proposed method successfully tracked the target of interest seamlessly without losing track across domains. Algorithm 1 briefly explains the sequence of operations for tracking. The results presented in the next section validate the performance of the proposed architecture.

IV. Results and Discussion

We evaluated the performance of "Siam-CDT" for cross-domain tracking of selective targets in multiple open-source datasets. The network is designed with only a regression head and the performance is quantified with standard regression metrics like R2 score, explained variance score (EVS) and mse. The dataset consists of multiple classes of targets to analyze the performance of the network in varied scenarios. Each dataset is split into training (64%), validation (16%), and testing (20%) subsets. The batch size is chosen to be one for all experiments except for sequential processing where a batch size of two was considered. The training is done using NVIDIA Quadro P4000 GPU and tested in real time on NVIDIA Jetson Nano board. Figs. 3 and 4 present the results of tracking a car and a person on a bike from VOT 2020 dataset. Both the targets were tracked successfully across domains with seamless track handover and without track loss. Fig. 5 shows the pedestrian tracking performance from the OSU dataset. The OSU dataset presented a potential problem that could occur in cross-domain applications. In some of the frames, the orientation of the camera blocked the RGB image information, which was visible in the thermal domain. Fig. 6(a) and (b) shows this partially occluded pedestrian in RGB images. Fig. 6(c) and (d) shows the completely captured thermal camera data. These cases will be outliers in Siamese implementations as the reference input images will have contradictory information. Therefore, the location of the camera plays a pivotal role in dual-sensor cross-domain applications. By fine-tuning the network with these outliers independently,

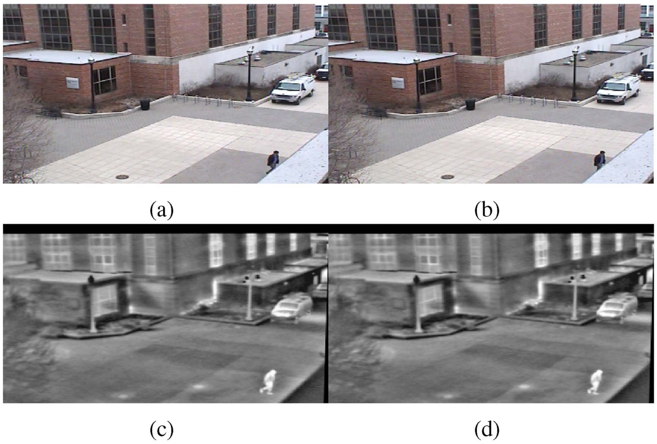


Fig. 6. (a) and (b) Partially occluded legs in RGB images which are fully visible in (c)–(d) of thermal images. These occluded images will be outliers in RGB/thermal tracking. Siamese method fails to handle this scenario, but fine-tuning the network can still successfully track the target with these outliers.

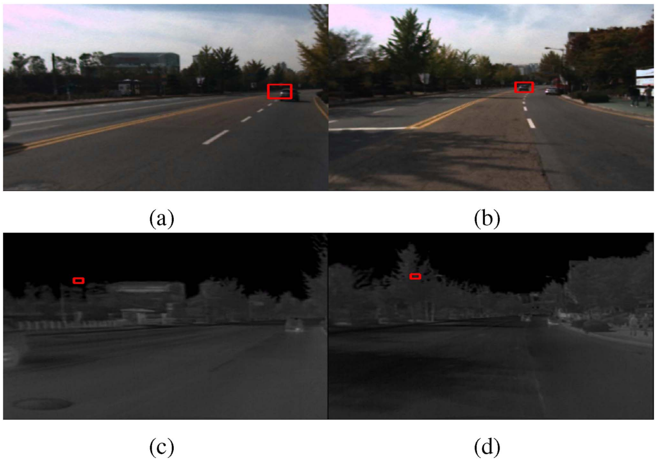


Fig. 7. (a) and (b) Tracking performance of the network for RGB input from the KAIST dataset. (c) and (d) Performance of the same network for thermal input of the KAIST dataset. Since the network was trained only on RGB data, the network could not detect the target in thermal images.

the network could successfully detect these outliers. We also incorporated l_2 norm check between consecutive feature vectors which checks for consistency of predictions and detects such outliers. The problem of standard CNNs working with multidomain data is explained in Figs. 7 and 8. We trained a "Tiny VGG" with only RGB data from the KAIST dataset and tested the network for both RGB and thermal datasets. Fig. 7(a) and (b) shows the tracking performance of the network for RGB (trained domain) data where the network successfully tracked the target. Fig. 7(c) and (d) shows the performance of the network for the thermal dataset. Though the images represent the same scene, the network could not detect the target in cross-domain data. We repeated the experiment by training "Tiny VGG" with only thermal data from VOT 2020 dataset and tested the network with both RGB and thermal images. The results obtained replicated the behavior of the previous experiment with the target being successfully tracked only in the trained domain and failing to do so in the other domain. The comparison metrics in Table I indicate the variation in data justifying these observations. Table III shows the comparison of performance metrics for the Siamese network and a single-domain RGB/thermal network. It can be observed that the performance of the Siamese network is almost similar to a single-domain network and, in some cases, even better. Table IV shows the cross-domain performance comparison of the Siamese network and single-domain RGB/thermal network. The performance of the Siamese network remains unchanged for cross-domain data, whereas single-domain trained CNNs demonstrate significant performance degradation for cross-domain data. In the KAIST dataset, the optical source is mounted on a moving vehicle and data is collected along the streets with vehicles, shops, and pedestrians. In this dataset, most of the targets of interest are close to the camera and quickly move out of the frame. So

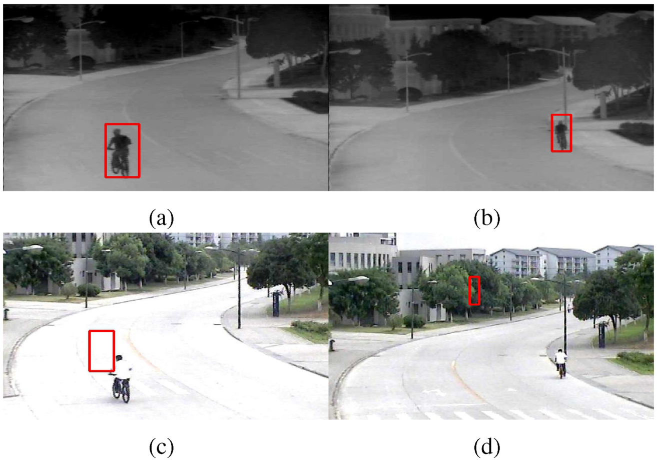


Fig. 8. (a) and (b) Tracking performance of network for RGB input from VOT 2020 dataset. (c) and (d) Performance of the same network for thermal version of same dataset. Since the network was trained only on RGB data, it could not detect the target in thermal images.

TABLE III
PERFORMANCE COMPARISON FOR SIAMESE AND SINGLE-DOMAIN NETWORK

Type of CNN	**Siamese CNN**	**RGB CNN**	**Siamese CNN**	**Thermal CNN**
Input Data (Domain)	RGB	RGB	Thermal	Thermal
KAIST (Target - Car)				
R2 Score	0.634	0.6912	0.574	0.6483
EVS	0.755	0.647	0.7461	0.8215
MSE	25.248	17.6303	27.417	22.4798
OSU (Target - Pedestrians)				
R2 Score	0.9763	0.9604	0.9817	0.9572
EVS	0.9814	0.9644	0.9789	0.9625
MSE	29.921	48.825	26.814	31.344
VOT (Target - Person on Bike)				
R2 Score	0.979	0.976	0.983	0.99948
EVS	0.9802	0.9898	0.9915	0.996
MSE	59.905	42.582	61.09	17.189
VOT (Target - Car)				
R2 Score	0.9761	0.9736	0.982	0.981
EVS	0.9819	0.9943	0.9842	0.9943
MSE	21.765	23.079	12.779	24.549

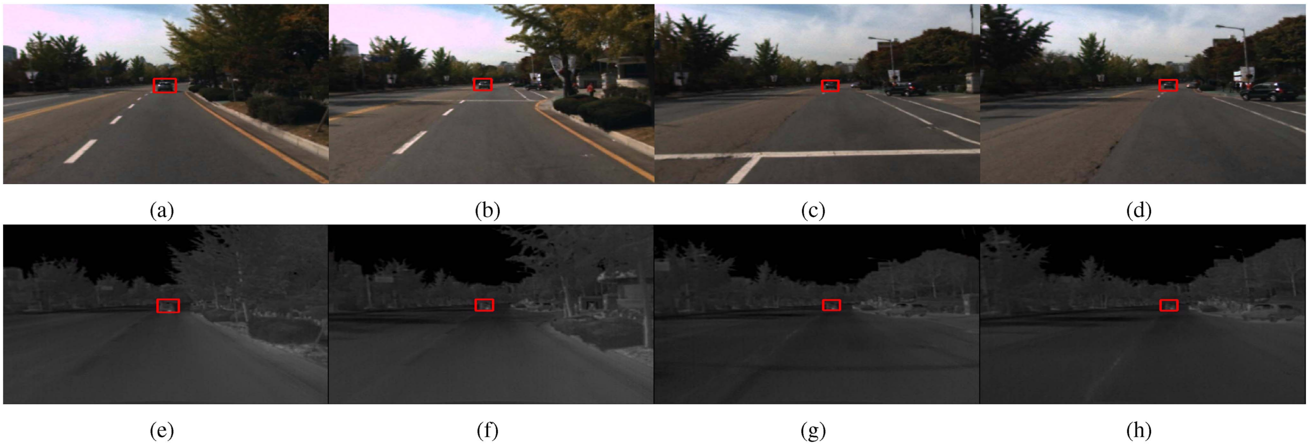


Fig. 9. Tracking results for KAIST [64] dataset with moving car target in consecutive frames of RGB and thermal videos. (a)–(d) Output of the network for RGB images. (e)–(h) Output of the same CNN for thermal images.

TABLE IV
PERFORMANCE COMPARISON FOR SIAMESE AND SINGLE-DOMAIN NETWORK FOR CROSS-DOMAIN INPUT

Type of CNN	**Siamese CNN**	**RGB CNN**	**Siamese CNN**	**Thermal CNN**
Input Data (Domain)	Thermal	Thermal	RGB	RGB
KAIST (Target - Car)				
R2 Score	0.574	-163.79	0.634	-37.104
EVS	0.7461	-9.371	0.755	-6.3294
MSE	27.417	13247	25.248	2575.22
OSU (Target - Pedestrians)				
R2 Score	0.9817	-0.8712	0.9763	-0.0805
EVS	0.9789	0.2296	0.9814	0.2905
MSE	26.814	5765.72	29.921	4677.42
VOT (Target - Person on Bike)				
R2 Score	0.983	-0.6413	0.979	-2.6092
EVS	0.9915	-0.071	0.9802	-0.9656
MSE	61.09	5792.17	59.905	13341.372
VOT (Target - Car)				
R2 Score	0.982	-2.012	0.9761	0.06179
EVS	0.9842	0.2586	0.9819	0.8180
MSE	12.779	6260.62	21.765	1175.24

for this dataset, we selected some instances from the videos where the target is visible at a distance and the camera mounted vehicle is trailing the target. Fig. 9 shows the performance of our algorithm for one such scenario. In all the cases, the test images were input sequentially from one domain and also a random mix of images across domains. In multiple target scenarios, the algorithm is susceptible to cross target tracking. This is more so in the thermal domain because of limited information about the target. To avoid this, we designed a selective region search technique based on initial target detection as shown in Fig. 10.

In CNN-based processing, every target in the frame is detected. Tracking a specific target among other potential targets is a challenging task. With the proposed method of fusing the mean-shift algorithm with a CNN, the search space is constrained around the predicted BB. We predict secondary search windows in directions other than target movement as shown in Fig. 10(a) and (b) to detect track changes. In case of track loss, the forward window (the window in the direction of target movement) is propagated and the remaining secondary search windows are locked in the space of last known detection as shown in Fig. 10(c) and (d) waiting for reacquisition of the target. If the target changes direction, based on the number of secondary windows designed, it is most likely to be detected in one of the secondary search spaces. Once the target is detected in the jth (one of the secondary search spaces) search window, it will be designated as the primary window and the track is continued.

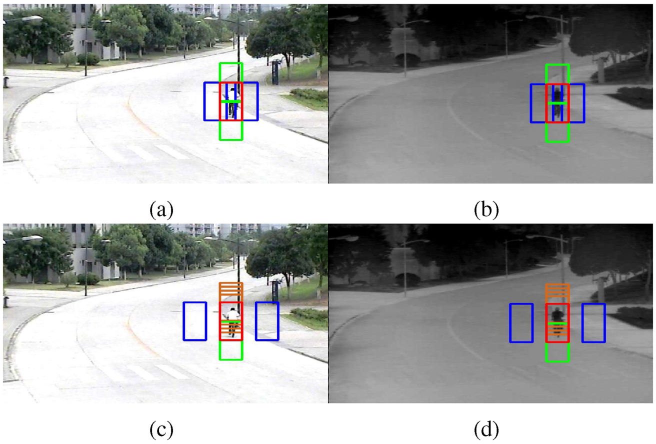
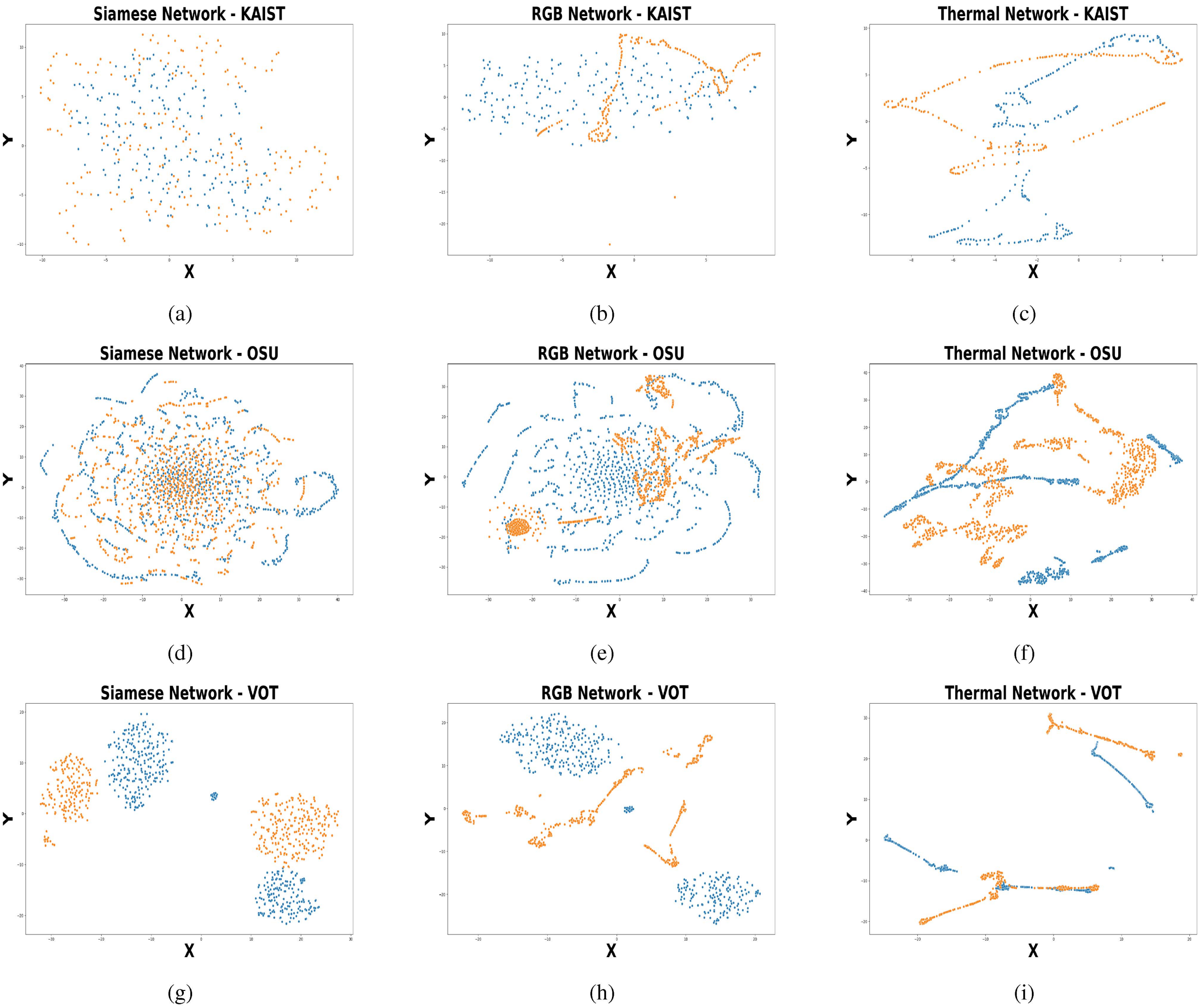


Fig. 10. (a) and (b) Search space in azimuth (x-axis) and elevation (y-axis) for tracking the target. (c) and (d) Coasting search space in the event of track loss. Blue windows show azimuth plane search space, green windows show the search space in elevation, and brown window shows the coasting search space. During coast phase, the search space in the direction of motion is propagated and other windows hold the same space in anticipation of acquiring maneuvered target as shown in (c) and (d).

To further validate the performance of "Siam-CDT," we generated tSNE plots for cross-domain data to understand the mapping of cross-domain data by "Siam-CDT" and standard single-domain networks. Fig. 11 shows the tSNE plot for cross-domain feature vectors at the output of the Siamese FCV network. It can be observed that the Siamese FCV network converges the cross-domain information to a similar feature vector representation as evident in Fig. 11(a), (d), and (g). However, the single-domain



(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

Fig. 11. Cross-domain performance of Siamese, RGB, and thermal networks for KAIST, OSU, and VOT dataset is compared with tSNE output. It can be observed that the feature vectors for cross-domain data in Siamese FCV network as shown in (a), (d), and (g) converge to a similar representation, whereas the performance of single-domain trained networks vary drastically as shown in (b), (c), (e), (f), (h), and (i). Blue dots represent RGB data points and orange dots represent thermal data points.

TABLE V
SPECIFICATIONS OF JETSON NANO

Parameter	Specification
GPU	128 Core Maxwell GPU
CPU	Quad-core ARM Cortex-A57 MPCore processor
Memory	4 GB 64-bit LPDDR4, 1600MHz 25.6 GB/s
Camera	2 lanes (3 × 4 or 4 × 2) MIPI CSI-2 D-PHY 1.1 (1.5 Gb/s per pair)
Dimension	69.6 mm x 45 mm

networks represent cross-domain data very differently as seen in the RGB and thermal network plots in Fig. 11(b), (c), (e), (f), (h), and (i). We used "Tiny VGG"[61] as the backbone for 'Siam-CDT" design. The network is successfully implemented on NVIDIA Jetson Nano board [69] and the results presented are obtained in hardware running the network in real time. Table V shows the specifications of the Jetson Nano board, the smallest GPU board offered by NVIDIA [69].

V. Conclusion

In this work, we addressed the problem of cross-domain tracking applications in computer vision and specifically implemented selective target tracking applications with seamless track handover across domains. We proposed a three-stage processing solution as shown in Fig. 2. The first stage of the solution is designed to generate a common feature vector representation for supervised cross-domain inputs using a Siamese FCV network. The converged Siamese FCV network is further augmented with an FCN to integrate the application in the shared latent space by directly predicting the BB coordinates. The final stage is designed to track the target continuously across domains with seamless track handover. We further proposed the fusion of the mean-shift algorithm with a CNN-based detector which drastically reduced the computational load by intermittently processing the data through the CNN channel. The proposed method is tested with multiple open-source datasets and for different types of targets. The results obtained conclusively prove the efficacy of Siamese-based design for RGBT systems.

References

[1] Y. Yu, Y. Xiong, W. Huang, and M. R. Scott, "Deformable Siamese attention networks for visual object tracking," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 6728–6737.

[2] Q. Guo, W. Feng, R. Gao, Y. Liu, and S. Wang, "Exploring the effects of blur and deblurring to visual object tracking," *IEEE Trans. Image Process.*, vol. 30, pp. 1812–1824, 2021.

[3] S. Moorthy, J. Y. Choi, and Y. H. Joo, "Gaussian-response correlation filter for robust visual object tracking," *Neurocomputing*, vol. 411, pp. 78–90, 2020.

[4] Y. Wang, X. Wei, H. Shen, L. Ding, and J. Wan, "Robust fusion for RGB-D tracking using CNN features," *Appl. Soft Comput.*, vol. 92, 2020, Art. no. 106302.

[5] R. Abbott, N. M. Robertson, J. Martinez del Rincon, and B. Connor, "Unsupervised object detection via LWIR/RGB translation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops*, 2020, pp. 90–91.

[6] Q. Xu, Y. Mei, J. Liu, and C. Li, "Multimodal cross-layer bilinear pooling for RGBT tracking," *IEEE Trans. Multimedia*, vol. 24, pp. 567–580, 2022.

[7] A. Lu, C. Li, Y. Yan, J. Tang, and B. Luo, "RGBT tracking via multi-adapter network with hierarchical divergence loss," *IEEE Trans. Image Process.*, vol. 30, pp. 5613–5625, 2021.

[8] Y. Wang, X. Wei, X. Tang, H. Shen, and H. Zhang, "Multimodal cross-layer bilinear pooling for RGBT tracking," *IEEE Trans. Multimedia*, 2021.

[9] C. Wang *et al.*, "Cross-modal pattern-propagation for RGN-T tracking," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 7064–7073.

[10] P. Zhang, J. Zhao, C. Bo, D. Wang, H. Lu, and X. Yang, "Jointly modeling motion and appearance cues for robust RGB-T tracking," *IEEE Trans. Image Process.*, vol. 30, pp. 3335–3347, 2021.

[11] Y. Iwashita, K. Nakashima, S. Rafol, A. Stoica, and R. Kurazume, "MU-Net: Deep learning-based thermal IR image estimation from RGB image," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, 2019, pp. 1022–1028.

[12] T. Kim, M. Cha, H. Kim, J. K. Lee, and J. Kim, "Learning to discover cross-domain relations with generative adversarial networks," 2017, *arXiv:1703.05192*.

[13] M.-Y. Liu and O. Tuzel, "Coupled generative adversarial networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2016, pp. 469–477.

[14] Y. Taigman, A. Polyak, and L. Wolf, "Unsupervised cross-domain image generation," 2016, *arXiv:1611.02200*.

[15] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 2223–2232.

[16] X. Wang and A. Gupta, "Generative image modeling using style and structure adversarial networks," in *Proc. Eur. Conf. Comput. Vis.*, Springer, 2016, pp. 318–335.

[17] C. Yang, T. Kim, R. Wang, H. Peng, and C.-C. J. Kuo, "Esther: Extremely simple image translation through self-regularization," in *Proc. BMVC*, 2018, p. 110.

[18] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 1125–1134.

[19] M.-Y. Liu, T. Breuel, and J. Kautz, "Unsupervised image-to-image translation networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 700–708.

[20] D. Jin, Z. Jin, J. T. Zhou, and P. Szolovits, "A simple baseline to semi-supervised domain adaptation for machine translation," 2020, *arXiv:2001.08140*.

[21] J. Donahue, J. Hoffman, E. Rodner, K. Saenko, and T. Darrell, "Semi-supervised domain adaptation with instance constraints," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2013, pp. 668–675.

[22] G. Koch, R. Zemel, and R. Salakhutdinov, "Siamese neural networks for one-shot image recognition," in *Proc. ICML Deep Learn. Workshop*, Lille, vol. 2, 2015.

[23] R. Hadsell, S. Chopra, and Y. LeCun, "Dimensionality reduction by learning an invariant mapping," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, 2006, pp. 1735–1742.

[24] S. Chopra, R. Hadsell, and Y. LeCun, "Learning a similarity metric discriminatively, with application to face verification," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 1, 2005, pp. 539–546.

[25] S. Dey, A. Dutta, J. I. Toledo, S. K. Ghosh, J. Lladós, and U. Pal, "Signet: Convolutional siamese network for writer independent offline signature verification," 2017, *arXiv:1707.02131*.

[26] X. Zhang, P. Ye, S. Peng, J. Liu, K. Gong, and G. Xiao, "SiamFT: An RGB-infrared fusion tracking method via fully convolutional siamese networks," *IEEE Access*, vol. 7, pp. 122122–122133, 2019.

[27] X. Zhang, P. Ye, S. Peng, J. Liu, and G. Xiao, "Dsiammft: An RGB-T fusion tracking method via dynamic siamese networks using multi-layer feature fusion," *Signal Processing: Image. Commun.*, vol. 84, 2020, Art. no. 115756.

[28] P. Jingchao, Z. Haitao, H. Zhengwei, Z. Yi, and W. Bofan, "Siamese infrared and visible light fusion network for RGB-T tracking," 2021, *arXiv:2103.07302*.

[29] A. Rozantsev, V. Lepetit, and P. Fua, "Flying objects detection from a single moving camera," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 4128–4136.

[30] N. M. Oliver, B. Rosario, and A. P. Pentland, "A Bayesian computer vision system for modeling human interactions," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 8, pp. 831–843, Aug. 2000.

[31] S. Noh and M. Jeon, "A new framework for background subtraction using multiple cues," in *Proc. Asian Conf. Comput. Vis.*, Springer, 2012, pp. 493–506.

[32] B. D. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," in *Proc. Int. Joint Conf. Artif. Intell.*, Vancouver, BC, Canada: Morgan Kaufmann Publishers Inc., 1981, pp. 674–679.

[33] T. Brox and J. Malik, "Large displacement optical flow: Descriptor matching in variational motion estimation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 3, pp. 500–513, Mar. 2011.

[34] R. Venkateswarlu, K. Sujata, and B. V. Rao, "Centroid tracker and aimpoint selection," in *Acquisition, Tracking, and Pointing VI*, vol. 1697. Bellingham, WA, USA: International Society for Optics and Photonics, 1992, pp. 520–529.

[35] D. S. Bolme, J. R. Beveridge, B. A. Draper, and Y. M. Lui, "Visual object tracking using adaptive correlation filters," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, 2010, pp. 2544–2550.

[36] M. Danelljan, G. Hager, F. Shahbaz Khan, and M. Felsberg, "Learning spatially regularized correlation filters for visual tracking," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2015, pp. 4310–4318.

[37] B. Li, J. Yan, W. Wu, Z. Zhu, and X. Hu, "High performance visual tracking with siamese region proposal network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 8971–8980.

[38] M. Felsberg *et al.*, "The thermal infrared visual object tracking vot-tir2015 challenge results," in *Proc. IEEE Int. Conf. Comput. Vis. Workshops*, 2015, pp. 76–88.

[39] Y. Wu, J. Lim, and M.-H. Yang, "Online object tracking: A benchmark," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2013, pp. 2411–2418.

[40] M. Danelljan, F. Shahbaz Khan, M. Felsberg, and J. Van de Weijer, "Adaptive color attributes for real-time visual tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2014, pp. 1090–1097.

[41] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista, "High-speed tracking with kernelized correlation filters," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 3, pp. 583–596, Mar. 2015.

[42] G. Lowe, "Sift-the scale invariant feature transform," *Int. J. Comput. Vis.*, vol. 2, pp. 91–110, 2004.

[43] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proc. IEEE Conf. Comp. Vision Pattern Recognit.*, vol. 1, 2005, pp. 886–893.

[44] P. Felzenszwalb, D. McAllester, and D. Ramanan, "A discriminatively trained, multiscale, deformable part model," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2008, pp. 1–8.

[45] Y. Yu, X. Si, C. Hu, and J. Zhang, "A review of recurrent neural networks: LSTM cells and network architectures," *Neural Comput.*, vol. 31, no. 7, pp. 1235–1270, 2019.

[46] K. Cho, B. Van Merriënboer, D. Bahdanau, and Y. Bengio, "On the properties of neural machine translation: Encoder-decoder approaches," 2014, *arXiv:1409.1259*.

[47] M. Danelljan, G. Bhat, F. S. Khan, and M. Felsberg, "Atom: Accurate tracking by overlap maximization," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 4660–4669.

[48] A. Milan, S. H. Rezatofighi, A. Dick, I. Reid, and K. Schindler, "Online multi-target tracking using recurrent neural networks," in *Proc. 31st AAAI Conf. Artif. Intell.*, 2017, pp. 4225–4232.

[49] C. Li, H. Cheng, S. Hu, X. Liu, J. Tang, and L. Lin, “Learning collaborative sparse representation for grayscale-thermal tracking,” *IEEE Trans. Image Process.*, vol. 25, no. 12, pp. 5743–5756, Dec. 2016.

[50] C. Li, X. Liang, Y. Lu, N. Zhao, and J. Tang, “RGB-T object tracking: Benchmark and baseline,” *Pattern Recognit.*, vol. 96, 2019, Art. no. 106977.

[51] C. Li, L. Liu, A. Lu, Q. Ji, and J. Tang, “Challenge-aware RGBT tracking,” in *Proc. Eur. Conf. Comput. Vis.*, Springer, 2020, pp. 222–237.

[52] Y. Zhu, C. Li, J. Tang, B. Luo, and L. Wang, “RGBT tracking by trident fusion network,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 32, no. 2, pp. 579–592, Feb. 2022.

[53] M. Kristan *et al.*, “A novel performance evaluation methodology for single-target trackers,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, pp. 2137–2155, Nov 2016.

[54] J. W. Davis and V. Sharma, “Background-subtraction using contour-based fusion of thermal and visible imagery,” *Comput. Vis. Image Understanding*, vol. 106, no. 2-3, pp. 162–182, 2007.

[55] A. He, C. Luo, X. Tian, and W. Zeng, “A twofold siamese network for real-time object tracking,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 4834–4843.

[56] L. Bertinetto, J. Valmadre, J. F. Henriques, A. Vedaldi, and P. H. Torr, “Fully-convolutional siamese networks for object tracking,” in *Proc. Eur. Conf. Comput. Vis.*, Springer, 2016, pp. 850–865.

[57] X. Dong and J. Shen, “Triplet loss in siamese network for object tracking,” in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 459–474.

[58] Q. Wang, Z. Teng, J. Xing, J. Gao, W. Hu, and S. Maybank, “Learning attentions: Residual attentional siamese network for high performance online visual tracking,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 4854–4863.

[59] F. N. Iandola, S. Han, M. W. Moskewicz, K. Ashraf, W. J. Dally, and K. Keutzer, “Squeezenet: Alexnet-level accuracy with 50x fewer parameters and 0.5 mb model size,” 2016, *arXiv:1602.07360*.

[60] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, “Mobilenetv2: Inverted residuals and linear bottlenecks,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 4510–4520.

[61] V. Chandrakanth, V. Murthy, and S. S. Channappayya, “UAV-based autonomous detection and tracking of beyond visual range (BVR) nonstationary targets using deep learning,” *J. Real-Time Image Process.*, pp. 1–17, 2021.

[62] 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.*, 2016, pp. 779–788.

[63] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, “Image quality assessment: From error visibility to structural similarity,” *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, Apr. 2004.

[64] S. Hwang, J. Park, N. Kim, Y. Choi, and I. S. Kweon, “Multispectral pedestrian detection: Benchmark dataset and baselines,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 1037–1045.

[65] W. Youn and H. Myung, “Robust interacting multiple model with modeling uncertainties for maneuvering target tracking,” *IEEE Access*, vol. 7, pp. 65427–65443, 2019.

[66] Z. Xie, W. Guan, J. Zheng, X. Zhang, S. Chen, and B. Chen, “A high-precision, real-time, and robust indoor visible light positioning method based on mean shift algorithm and unscented Kalman filter,” *Sensors*, vol. 19, no. 5, 2019, Art. no. 1094.

[67] J. G. Ellis, K. A. Kramer, and S. C. Stubberud, “Image correlation based video tracking,” in *Proc. IEEE 21st Int. Conf. Syst. Eng.*, 2011, pp. 132–136.

[68] M. Zhang and X. Li, “An efficient real-time two-dimensional CA-CFAR hardware engine,” in *Proc. IEEE Int. Conf. Electron Devices Solid-State Circuits*, 2019, pp. 1–3.

[69] N. Corp., “GPUs,” [Online]. Available: https://www.nvidia.com/