

Received May 15, 2018, accepted July 22, 2018, date of publication July 25, 2018, date of current version August 20, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2859595

Multiple-Model Fully Convolutional Neural Networks for Single Object Tracking on Thermal Infrared Video

MOHD ASYRAF ZULKIFLEY^{1,2}, (Member, IEEE), AND NIKI TRIGONI²

¹Centre for Integrated Systems Engineering and Advanced Technologies, Faculty of Engineering and Built Environment, Universiti Kebangsaan Malaysia, Bangi 43650, Malaysia

²Department of Computer Science, University of Oxford, Oxford OX1 2JD, U.K.

Corresponding author: Mohd Asyraf Zulkifley (asyraf.zulkifley@cs.ox.ac.uk)

This work was supported in part by Universiti Kebangsaan Malaysia under Grant GUP-2015-053 and through Dana Impak Perdana under Grant DIP-2015-006, and in part by the Ministry of Higher Education Malaysia through the Fundamental Research Grant Scheme under Grant FRGS/2/2014/TK03/UKM/02/5.

ABSTRACT The availability of affordable thermal infrared (TIR) camera has instigated its usage in various research fields, especially for the cases that require images to be captured in dark surroundings. One of the low-level tasks required by most TIR-based researches is the need to track an object throughout a video sequence. The main challenge posed by TIR camera usage is the lack of texture to differentiate two nearby objects of the same class. According to the VOT-TIR 2016 challenge, the best fully convolutional neural network (FCNN)-based tracker has only managed to obtain the third place. The discriminative ability of the FCNN tracker is not fully utilized because of the homogenous appearance pattern of the tracked object. This paper aims to improve FCNN-based tracker ability to predict object location through comprehensive sampling approach as well as better scoring scheme. Hence, a multiple-model FCNN is proposed, in which a small set of fully connected layers is updated on the top of pre-trained convolutional neural networks. The possible object locations are generated based on a two-stage sampling that combines stochastically distributed samples and clustered foreground contour information. The best sample is selected according to a combined score of appearance similarity, predicted location, and model reliability. The small set of appearance models is updated by using positive and negative training samples, accumulated from two periods of time which are the recent and parent node intervals. To further improve training accuracy, the samples are generated according to a set of adaptive variances that depends on the trustworthiness of the tracker output. The results show an improvement over TCNN, an FCNN-based tracker that won the VOT 2016 challenge with the expected average overlap increasing from 0.248 to 0.257. The performance enhancement is attributed to the better robustness with a 20% reduction in tracking failure rate compared to the TCNN.

INDEX TERMS Visual object tracking, thermal infrared video, fully convolutional neural networks.

I. INTRODUCTION

Thermal infrared (TIR) camera provides a monochrome heat map information of the scene by capturing radiation in the long infrared wavelength band [1]. It is widely used in military applications to detect objects of interest that have colder temperature compared to the background. Recent advances have made it possible for civilians to buy cheap TIR cameras starting from low £200 (FLIR TG130 Spot Thermal Camera) for the research purposes.¹ Some of the research fields that

have utilized input from TIR camera are human stress level predictor [2], ecology monitoring [3], poacher detection [4] and wildlife observation [5]. It is interesting to note that all the previously mentioned research requires some sort of tracking function to be embedded in their systems. By having tracking capability, trajectory information can be used to infer the object's state more accurately.

There are two main philosophies that have produced the best tracking performance in terms of accuracy and precision: trackers based on fully convolutional neural networks (FCNN) and trackers based on discriminative

¹<https://www.testers.co.uk/flir-tg130-spot-thermal-camera>

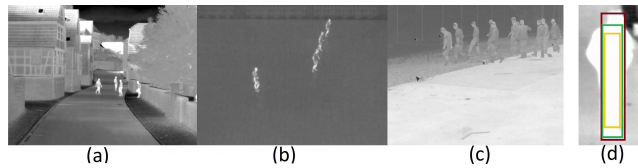


FIGURE 1. (a)-(c) Samples of objects with similar heat map, (d) samples of bounding boxes that produce almost similar heat maps after warping process to 75×75 pixels.

correlation filters (DCF). According to VOT 2016 (RGB input) challenge [6], the performance of both types of tracker do not differ too much with performance difference of 0.006 in expected average overlap (EAO) between C-COT (the best DCF-based tracker [7]) and TCNN (the best FCNN-based tracker [8]). However, the results of VOT-TIR 2016 (TIR input) challenge shows that the performance of FCNN-based trackers is not on par with the DCF-based trackers with EAO gap of 0.077. Hence, it serves as the motivation for us to build upon FCNN approach to produce a better TIR tracker.

The primary differences between image produced by TIR and RGB cameras are the lack of textures, structures and edge information of the tracked object. Furthermore, it is hard to distinguish between two nearby objects because of the similar heat maps as shown in Figure 1(a)-(c). But, TIR camera is very useful if the objects are camouflaged by the surrounding objects since heat information can still be captured clearly. Moreover, it can still capture good images during night or in the absence of good lighting source. These advantages are important for the surveillance and security applications, where object of interest has a high possibility to be hidden by the surroundings. The scopes of the proposed tracker follow the same rules as in VOT-TIR challenge that limit the system to be model free, single object of interest and causal tracker. Even though VOT 2016 challenge has implemented rotated bounding box representation, this work adopts the normal upright bounding box as used in VOT-TIR 2016 challenge, which provides the benchmark result for the state of the art trackers.

The basis for this work is the VOT 2016 challenge winner algorithm, TCNN by Nam *et al.* [9]. However, it only achieved third best performance in VOT-TIR 2016 challenge with significant EAO difference as compared to the winner. In this paper, the improvement efforts are focused mainly on two areas, which are sample generation and modeling compact multiple model CNNs. In TCNN, the samples are generated randomly through gaussian distributed sampling pivoted on the previous location of the object. There is no assumption on the movement direction that results in 360° samples being distributed all around the last known position of the object. This procedure limits the tracker performance as most of the samples will fall out of the true object direction. Thus, the pivot point in which the samples are generated plays an important role on the number of samples needed to find the object. Hence, a new resampling procedure is proposed in

which the first stage sampling focuses on finding the possible directions of the object, before it is sampled again to fine tune the possible object locations. A simple normalized cross correlation approach is used to find the possible directions, which will be the pivot points for the second stage sampling. This approach reduces the required number of test samples to be generated as a smaller set of variances can be used.

Some of the test videos in the VOT-TIR database consist of human as the object of interest, which is highly non-rigid in nature. Variations in the object size are hard to be captured perfectly if only random sampling is used. It is possible but the sampling numbers will be too large to accommodate the size variations and consequently the computational speed will be much slower. Therefore, objectness philosophy is adopted with an assumption that a combination of small foreground contour blobs will make up the true object size. The insight is to increase the possibility of creating better size samples of the tracked object, where CNN warping process in TCNN tends to shrink the object size. Figure 1(d) is an example where a smaller bounding box produces the same appearance score as the bigger box, especially in the absence of texture information as in thermal infrared image. However, careful implementation is needed as samples built upon the foreground segmentation approach suffers performance reduction if the camera moves abruptly or the object encounters other moving objects.

In modeling the multiple-FCNN, TCNN updates its appearance model tree with the most similar parent node. If the appearance difference between the parent and child nodes is minute, unnecessary advantage has been given to that particular tree branch where two models with similar appearance are kept. The scoring scheme used to rank the samples is the average score of all CNN nodes and with more representations, the result will skew towards that particular branch. This limits the variability of the appearance model, even though 10 nodes are used in the TCNN. Hence, our proposed algorithm, which we refer to as multiple-model FCNN (MMFCNN) maintains a compact collection of models with at most three models, in which the new node will replace the parent node. From our observation, maintaining a set of few nodes is necessary, especially in the occlusion case. The appearance of the object before and during the occlusion will be different depending on the occlusion severity. If only one node is maintained, the model will have higher possibility to be updated with the occlusion information. But, if at least two nodes are maintained, the first node might get updated with the noisy information, while the second node still remain unchanged. Once the occlusion is over, the second node can act as a good point of reference for the object appearance model to reduce model drifting.

Even with careful design, it is rare for the output bounding box to perfectly capture the true size of the object. Thus, the proposed algorithm is trained by using positive and negative samples with adaptive variance to cater for uncertainty in the imperfect output bounding box. The training samples are generated based on gaussian distribution with variances

derived from top n -score samples. The variances will be small if the fluctuation in localization of top n -score samples is stable and vice versa. Briefly, our contributions are in two folds:

- **Samples generation:** A two-stage sampling procedure is proposed that combines stochastic sampling and clustered foreground contours.
- **Multiple-model FCNN:** We propose a combination of appearance, reliability and distance scores based on compact n -model fully convolutional neural networks with adaptive variance update.

In the next section, a brief summary on single object trackers based on FCNN is reviewed. Then, Section 3 introduces the proposed MMFCNN tracker whose performance is evaluated and discussed in Section 4. Finally, Section 5 concludes the paper.

II. RELATED WORK

Visual object tracking is a very popular research topic. It has several annual dedicated challenges to tackle different aspects of tracking, mainly divided into single and multi-object. The motivation behind this work is to improve FCNN-based single object tracking performance for thermal infrared camera input. Therefore, some of the main concerns for multiple objects tracking such as track merging, track splitting, identification swapping and optimal observation association as in [10]–[13] are not considered. However, it can be observed that most of the single object tracking datasets consist of videos with higher difficulty in the object appearance evolution compared to the multi-object tracking datasets.

The main challenges in single object tracking are the presence of occlusions, illumination changes, dynamic changes, camera motion, abrupt movement and nonrigid transformation of the object appearance. To cater for many possibilities of the appearance model, convolutional neural networks (CNN) has been successfully used to extract tracked object features. Wang and Yeung [14] introduce one of the earliest versions of deep learning feature representation that uses stacked denoising auto encoder, trained on tiny image dataset [15]. The trained encoder component is then used to extract features of the tracked object. If the confidence level of the encoder output drops below a predefined threshold, it will be retuned again. Instead of using the deep feature directly to classify the object, Hong *et al.* [16] add a support vector machine at the output layer of pretrained CNN layers developed by Girshick *et al.* [17]. A similar motivation is used in [18], where a CNN feature is used as input feature representation to discriminative correlation filter to find the best object localization. A fully convolutional neural network consists of convolution layers and fully connected (FC) layers is applied in [8] that trains multiple domains specific to the object. A tree structured fully convolutional network is also applied in [9] to maintain an ensemble of n models which are updated periodically in a fixed interval. This structure allows the child node to be spawned with the help from

positive training samples from the parent node. It maintains several appearance representations of the object based on the training batch. To avoid the overfitting problem, BranchOut mechanism is introduced in [19] by randomly dropping a FC branch during the training process. A similar idea is applied in [20] to avoid overfitting issue by employing a random binary mask to force the tracker to learn different portions of the object.

Instead of using the last layer of convolutional output, method in [21] uses the output of every CNN layers to build a hierarchical feature representations of the object. The authors argue that the top CNN layer captures a better spatial information, while the last CNN layer captures more discriminative features of the tracked object. A correlation filter is then applied on the resized feature maps to localize the object position. The same idea is used in [22], in which they have used the output of the fourth and fifth layers of the CNNs to build the tracked object feature maps. The fourth layer focuses on generating the saliency map used for positioning, while the fifth layer focuses more on the discriminative features of the tracked object. Hence, nearby objects with similar appearance can be distinguished by putting more emphasis on the fourth layer output. In [23], CF has been applied to the output of each CNN layer, which are later combined by using the Kullback-Leibler divergence metric. Instead of using a fixed set of weights to combine the CFs, a hedging approach is introduced in [24] to assign the weights adaptively. A different strategy is used in [25], such that several discriminative CFs of different scale are generated first before applying them to the CNN layers to produce the tracked object feature. A hybrid method is also explored in [26], where a fully convolutional network is used to get the initial estimate of the object, which is then fine-tuned by using the information from each CNN layer to adjust the spatial localization of the object.

III. MULTIPLE-MODEL FULLY CONVOLUTIONAL NEURAL NETWORK TRACKER

The proposed multiple-model fully convolutional neural network (MMFCNN) tracker shares some fundamental similarities with MDNet-N [8] and TCNN [9] but differs a lot in manipulation strategies of the fully connected layers (FC). FC in MDNet-N is trained with large bias towards the last layer. Moreover, MDNet-N uses only the most recent batch of training data for model update. This approach does not perform well when the recent batch of positive training data is contaminated by background information which affects future update of the appearance models. TCNN implemented a tree structure to handle the FC nodes, so that positive training data can be selected from the best matched appearance model, instead of the most recent one. This allows a badly updated appearance model to be discarded in the future. However, FC nodes retention for TCNN still favor continuous temporal node update, in which the oldest node will be deleted and the child node will branch out from the most similar node. Hence, MMFCNN modifies the node retention procedure by keeping a small set of frequently updated nodes only. The new child

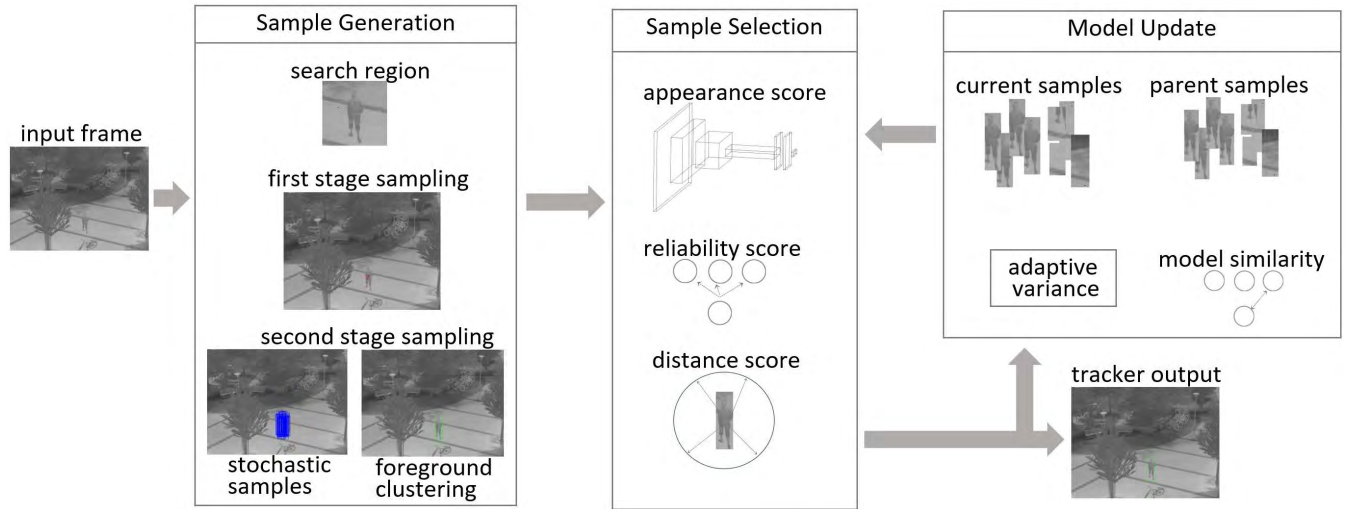


FIGURE 2. Overview of MMFCNN: 1) Samples are generated by a two-stage procedure where the second stage consists of normally distributed samples and clustered foreground blobs, 2) total samples scores depend on the appearance, predicted distance and reliability models and 3) adaptive variance is used to draw out training samples for MMFCNN update, where the new model will replace the most similar model.

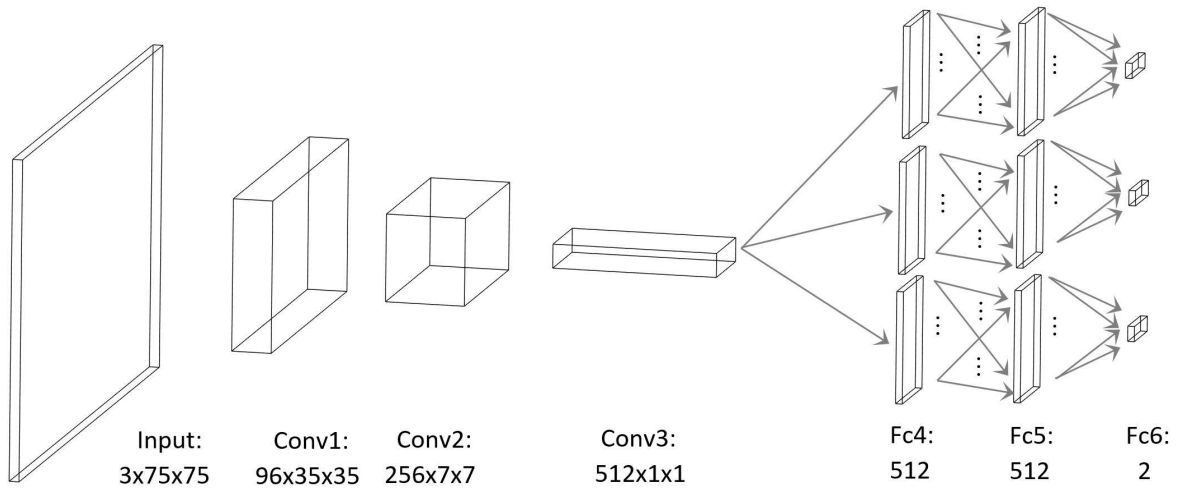


FIGURE 3. Network architecture of MMFCNN with three CNN layers and three FC layers.

node will replace the parent node so that no particular branch carries more weight than the others. This allows MMFCNN to maintain diverse appearance model with a smaller set of nodes. Possible positions of the tracked object is first inferred by using a two-stage sampling which is then evaluated based on combined scores of appearance, predicted distance and reliability models. The proposed system overview is shown in Figure 2.

A. MODEL DEFINITION

The proposed network consists of three convolution layers and three fully connected layers. Input to the first convolution layer is fixed to 75×75 , and hence all training and testing samples will be warped to the corresponding size. Figure 3 shows the MMFCNN architecture with three FC models.

B. SAMPLE GENERATION

MMFCNN is a track-by-detection tracker that finds the best matched bounding box from a set of samples $\mathbf{S} = \mathbf{S}_{\text{fg}} \cup \mathbf{S}_{\text{gauss}} = \{s^1, \dots, s^{|\mathbf{S}|}\}$. The usual approach for generating samples is by using the previous location as the pivot point. This approach requires a wide selection of variance set as the object movement can be fast and slow intermittently. Hence, the idea of resampling in the particle filter [27] is also applied to MMFCNN to create a group of samples at more strategic locations. Yet, to run the CNN twice will incur a lot of computational burden to the system. Hence, a weak tracker based on correlation filter is applied to sample a set of points \mathbb{P} to be the pivot points for the second stage sampling. This step allows MMFCNN to have more concentrated search regions with small variance because of high confidence on \mathbf{P} locality.

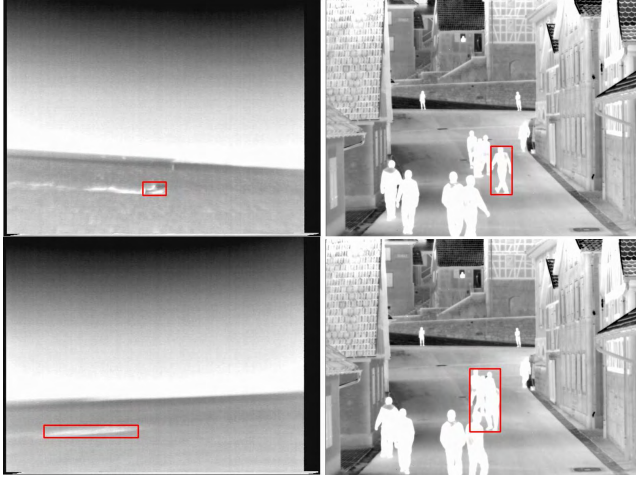


FIGURE 4. The first row shows that a correct bounding box of the tracked objects can be inferred from objectness information, while this is not the case in the second row images.

For the second stage sampling, a combination of normally distributed samples (S_{gauss}) around \mathbf{P} points and clustered foreground blobs samples (S_{fg}) are generated. From our observation, a set of normally distributed samples alone will eventually cause object size drift. Hence, on top of normally distributed samples, clustered foreground blobs that represent the objectness information is also integrated into our approach. In [28], the objectness is derived from gradient information, which we found to be less effective for thermal infrared video, especially if the input range is saturated or blurred as shown in Figure 4.

However, foreground based samples are only effective during non-clutter situations. Cluttered scenes can happen because of two main reasons: 1) existence of other moving objects in the vicinity of the tracked object and 2) abrupt camera movement. In both cases, MMFCNN will revert to just normally distributed samples as foreground-based samples are deemed to be too noisy. The method in [29] is applied to analyze the clutter level of the search region. With the hybrid samples, size drift is less likely to occur as clustered foreground blobs will provide good information on the tracked object bounding box.

Given a set of clustered foreground regions $\mathbf{C} = \{c^1, \dots, c^{|\mathbf{C}|}\}$, a base region s_{base} is first constructed to indicate the lower limit of the samples size. s_{base} is obtained by combining all foreground regions c that fulfill certain overlapping criteria. The combined regions $c = \{c^x, c^y, c^w, c^h\} \in \mathbb{R}^{1 \times 4}$ forms the foreground regions base set $\mathbf{C}_{\text{base}} \subset \mathbf{C}$ which are constrained by the size limit of the previous output bounding box, s_{output}^{t-1} where $\|c^x - s_{\text{output}}^{x,t-1}\|_2 < \frac{s_{\text{output}}^{w,t-1}}{2}$ and $\|c^y - s_{\text{output}}^{y,t-1}\|_2 < \frac{s_{\text{output}}^{h,t-1}}{2}$. Then, a set of foreground-based samples S_{fg} are generated by combining s_{base} with foreground region c which has not been used to construct s_{base} .

$$s_{\text{base}} = \bigcup_{j=0}^{|\mathbf{C}_{\text{base}}|} c^j, c^j \in \mathbf{C}_{\text{base}} \quad (1)$$

The left over foreground clusters set \mathbf{C}_{fg} is a complementary set $\mathbf{C} \setminus \mathbf{C}_{\text{base}}$. Each foreground clusters in \mathbf{C}_{fg} is then used to generate S_{fg} with respect to s_{base} .

$$S_{\text{fg}} = \{s_{\text{base}} \cup c^j | c^j \in \mathbf{C}_{\text{fg}}, j = 1, \dots, |\mathbf{C}_{\text{fg}}|\} \quad (2)$$

C. SAMPLE SELECTION

MMFCNN maintains n -model of object appearance at any particular time. Let M^j be the total score for $s^j, j = \{1, \dots, |\mathbf{S}|\}$, which is dependent on the appearance score $f(A)^j$, distance score $f(D)^j$ and reliability score $f(R)^j$.

$$M^j = f(A)^j f(D)^j f(R)^j, j = \{1, \dots, |\mathbf{S}|\} \quad (3)$$

$f(A)^j$ is the appearance similarity measure of the j^{th} sample given a set of \mathbf{N} FCNN models. The last fully connected layer (Fc6) of each $i \in \mathbf{N}$ model serves as the appearance score, $f(A)_i^j$. Then, the final appearance score for a sample j is the average score of all its $f(A)_i^j, i = \{1, \dots, |\mathbf{N}|\}$ scores.

$$f(A)^j = \frac{1}{n} \sum_{\forall i \in \mathbf{N}} f(A)_i^j \quad (4)$$

However, appearance modeling alone might be misleading if any of the model has been updated with noisy training samples. Hence, reliability of the updated model should be considered in calculating M . For a sample $j, f(R)^j$ is obtained based on the average of accumulated scores M of the tracker output over a fixed m interval frames.

$$f(R)^j = \frac{1}{m} \sum_{j=t-m}^{t-1} M_{\text{output}}^j \quad (5)$$

Another important consideration in finding total score is the distance of a sample with regards to its predicted location. Standard thermal cameras do not provide much texture information and it is hard to distinguish between two objects with similar size as shown in Figure 1(c). Hence, movement consistency should be factored in to distinguish the two possible objects. Let r^j be the radius of the samples j from the predicted location X_{predict} , σ_D is the variance of the search region and z is the normalization constant.

$$f(D)^j = \frac{1}{z} \exp^{-\frac{r^2}{\sigma_D}} \quad (6)$$

The variance is set to a big value so that it will limit the contribution of distance score compared to the total score. Furthermore, reliance on a single sample output will lead to a sudden jump in localization and object size. Hence, the output of the tracker for each frame, s_{output}^t is set to be dependent on the best- k samples set, $\mathbf{L} \subset \mathbf{S}$ that corresponds to the top- k highest M .

$$s_{\text{output}}^t = \frac{1}{|\mathbf{L}|} \sum_{\forall l \in \mathbf{L}} s_l^t \quad (7)$$

D. MODEL UPDATE

The primary consideration in maintaining a set of models is to make sure that they are diverse enough to cater uncertainties without storing repetitive models. The strategy of using n models allows MMFCNN to perform well during and right after an occlusion. If only a single model is maintained, most likely, the appearance model will drift with the inclusion of noisy information. Right after an occlusion finished, it is hard for a single model system to find a good match if it has been updated continuously with the occlusion data.

The original TCNN spawns a new model by using a tree structure where the parent node will still be maintained and the oldest node is deleted. MMFCNN opts to kill the parent node once the child node is born out. This is inline with the decision model that uses average score so that the weight distribution will not be skewed towards a certain object appearance. If two similar models are maintained, an unnecessary advantage is given to that particular appearance. Hence, MMFCNN maintains a diverse set of models by replacing the parent model. The positive and negative training samples are accumulated through out a fixed m recent frames. Let the training samples set be $\mathbf{S}_{\text{train}} = \{s_{\text{train}}^0, \dots, s_{\text{train}}^{|S|}\}$ and t denotes the time.

$$\mathbf{S}_{\text{train}, +ve} = \bigcup_{j=t-m}^t \mathbf{s}_{\text{train}, +ve}^j \quad (8)$$

$$\mathbf{S}_{\text{train}, -ve} = \bigcup_{j=t-m}^t \mathbf{s}_{\text{train}, -ve}^j \quad (9)$$

Note that sampling variance is not fixed in MMFCNN since we might trust the output differently for different frames. The top k -best samples during output selection are used to infer the adaptive variance. If the variance σ_{update} in the samples position is small, the training sample variance will also be small as the likelihood of the result to be trustworthy is high and vice versa. X^j is the (x, y) coordinate of the middle point of a sample j and $k := |\mathbf{L}|$ is the number of top samples used to get s_{output}^t .

$$\sigma_{\text{update}} = E[X - \bar{X}] = \frac{1}{|\mathbf{X}|} \sum_{j=0}^k X^j - \bar{X}^2 \quad (10)$$

σ_{update} is critical to determine the sampling range of positive training data. TCNN assumes a fixed variance regardless of how good or bad the current smoothed position is. We argue that the variance should be increased if the current update is not trustworthy enough. The assumption is that position variations in the top- K candidate boxes are good indicator to the update quality. If the variation in the x coordinates of the boxes is small, there is high certainty that we do not need to sample training data with big variance.

A new appearance model, n_{new} will replace the most similar model, n_{parent} where the training samples used during spawning the parent model are also included in the current $\mathbf{S}_{\text{train}}$ as shown in Figure 5.

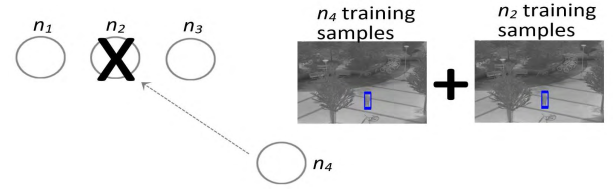


FIGURE 5. Model update: n_4 will replace n_2 with training combined training samples from two frames interval.

IV. EXPERIMENTS

A. IMPLEMENTATION DETAILS

The full implementation of MMFCNN is written in Python with Tensorflow library. MMFCNN is a binary class tracker, which is trained to distinguish between foreground (the tracked object) and background classes. The sampling procedures to extract foreground and background boxes are explained in section IV.B. One-hot encoded labeling is used to represent the class label. It is advisable to set a minimum of $n \geq 3$ appearance models to cater for occlusion and noisy update cases. Adam optimizer [30] is used to train the network with training and initialization learning rate of 0.001 and 0.0005, respectively. The total number of samples for the second stage sampling is limited to 250. The top 12 algorithms from VOT-TIR 2016 challenge are used to benchmark the performance of the proposed algorithm that include SRDCFir [18], EBT [28], TCNN [9], Staple-TIR [31], SHCT [32], MDNet-N [8], Staple+ [33], DSST2014 [34], MvCF [35], DPT [36], deepMKCF [37] and MAD [38]. The VOT-TIR 2016 dataset \mathbb{A} is categorized into three classes according to [39], which are challenging set \mathbb{D} , moderate set \mathbb{M} and easy set \mathbb{E} . All 25 video sequences are evaluated 15 times, inline with the stochastic sampling protocol.

B. TRACKER INITIALIZATION

Since a model free tracker is assumed, the only information given in the first frame is the bounding box of the object of interest. The appearance model of MMFCNN is generated based on the given ground truth bounding box $s_{gt}^{t=0}$ where $s = [x, y, \text{width}, \text{height}]$. The pre-trained weights for convolution layers are obtained from VGG-M [40] that has been trained on the ImageNet dataset [41]. During initialization, all n models will be initialized and trained by using the same positive and negative samples. All models need to be trained in the first frame so that the tracker can handle any early occlusion case. If the model is added incrementally, the subsequent models might be exposed to noisy data and cause model drift. Both positive training set, $\mathbf{S}_{\text{train}, +ve}$ and negative training set, $\mathbf{S}_{\text{train}, -ve}$ are sampled with uniform distribution around $s_{gt}^{t=0}$ with limiting factor of intersection over union (IoU) as in [9]. Let the sampling range $[\beta_a, \beta_b]$ be the upper and lower limit while α_1 and α_2 are the IoU threshold for positive and negative training data, respectively. Let r^t be $\frac{w^{t-1} + h^{t-1}}{2}$, then σ_D^t is $0.3r^t$ and $[\beta_a^t, \beta_b^t]$ for positive and negative training data

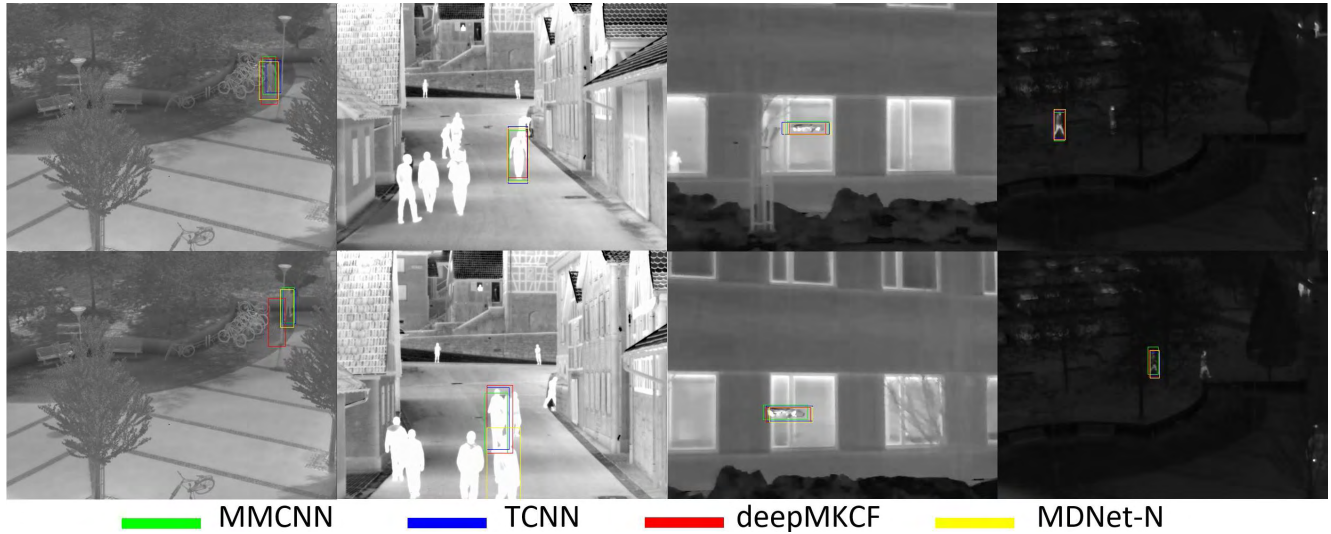


FIGURE 6. Output samples of FCNN-based trackers.

TABLE 1. Expected average overlap, accuracy, robustness and reliability results of the FCNN-based tracker separated by test videos difficulty level on VOT-TIR 2016 dataset.

Method	EAO				Ac			Ro			Re		
	A	D	M	E	D	M	E	D	M	E	D	M	E
MMFCNN	0.257	0.106	0.264	0.290	0.455	0.579	0.588	4.65	0.35	0.08	0.52	0.93	0.98
TCNN	0.248	0.063	0.281	0.288	0.565	0.623	0.633	5.80	0.38	0.20	0.49	0.94	0.96
deepMKCF	0.246	0.065	0.240	0.251	0.557	0.613	0.651	8.63	0.63	0.44	0.25	0.84	0.97
MDNet-N	0.235	0.059	0.260	0.299	0.604	0.645	0.638	7.27	0.62	0.15	0.40	0.85	0.98

are $[-0.1r^t, 0.1r^t]$ and $[-r^t, r^t]$, respectively. Both α_1 and α_2 are set to 0.3 and 0.7, respectively.

$$S_{\text{train}, +ve} = \{s_{\text{train}, +ve} \sim \mathcal{U}(\beta_a, \beta_b) | s_{\text{train}, +ve} < \alpha_1 \text{IoU}\} \quad (11)$$

$$S_{\text{train}, -ve} = \{s_{\text{train}, -ve} \sim \mathcal{U}(\beta_a, \beta_b) | s_{\text{train}, -ve} > \alpha_2 \text{IoU}\} \quad (12)$$

C. PERFORMANCE EVALUATION MEASURE

Four VOT-TIR evaluation metrics that include accuracy (Ac), robustness (Ro), reliability (Re) and expected area overlap (EAO) are used to quantify MMFCNN performance. Ac concerns on how well the tracked output bounding box compared to the ground truth s_{gt} while Ro measures the number of tracking failures F in a video sequence of Q length. Since Ro does not have upper bound, Re is introduced to measure the likelihood of successful tracking after T frames, which is set to 100. ψ denotes number of frames for the test videos, which vary from 72 to 1451.

$$Ac = \frac{1}{\psi} \sum_{i=1}^{\psi} \frac{s_{i,\text{output}} \cap s_{i,\text{gt}}}{s_{i,\text{output}} \cup s_{i,\text{gt}}} \quad (13)$$

$$Ro = \sum_{j=0}^Q F^j \quad (14)$$

$$Re = e^{-T \frac{Ro}{Q}} \quad (15)$$

Both Ac and Ro follow re-initialization protocol, which is triggered if the tracker output IoU is less than zero. On the

contrary, EAO does not requires re-initialization, in which it averages the IoU over a range of frames between upper limit g^{up} and lower limit g^{low} . It is used to rank the tracker by integrating the tradeoff between accuracy and robustness of the algorithm.

$$EAO = \frac{1}{g^{\text{up}} - g^{\text{low}}} \sum_{j=g^{\text{low}}}^{g^{\text{up}}} \text{IoU}^j \quad (16)$$

D. COMPARISON TO FCNN-BASED TRACKERS

The motivation for this work is to improve FCNN-based tracker performance, in which three other trackers fall into this category (TCNN, deepMKCF and MDNet-N). Table 1 states the full results according to difficulty level of the test videos. MMFCNN performs the best with the highest EAO of 0.257, followed by TCNN, deepMKCF and MDNet-N with EAO of 0.248, 0.246 and 0.235, respectively. The good performance of MMFCNN is attributed to its ability to perform well in the challenging dataset, while maintaining good performance in the moderate and easy datasets. Even though the average accuracy of MMFCNN is the lowest for all test set, it performs the best for both robustness and reliability measures. Indeed, this is the normal case for a tracker that sustains a longer period of successful tracking, in which the accuracy drops as time goes on. On the other hand, a tracker with higher failure rate will have to re-initialize more

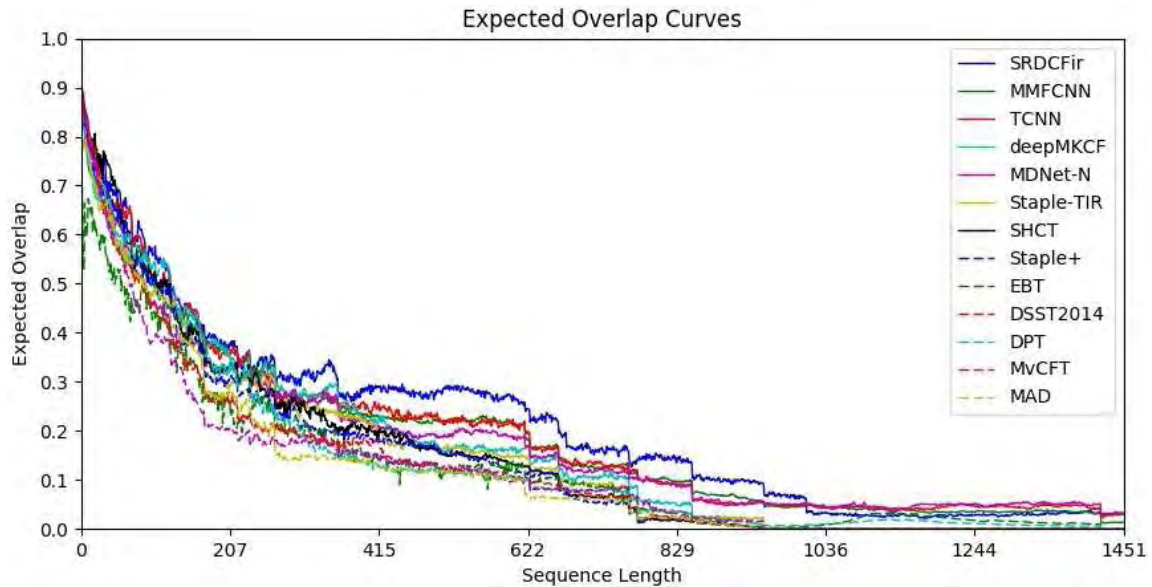


FIGURE 7. Expected average overlap curve tested on VOT-TIR 2016 dataset.

TABLE 2. Total number of zero *IOU* (failure) for the test range of [223,509] as used in VOT-TIR 2016 challenge.

Method	MMFCNN	TCNN	deepMKCF	MDNet-N
Failure	9	12	12	10

frequently, and hence, improve its average accuracy. Interesting to note that MMFCNN has also a smaller number of completely failed tracking cases as measured by zero average *IOU* with 9 videos as compared to 12 videos for TCNN as shown in Table 2. This indicates that 12 out of 25 test videos contribute zero *IOU* to the $EO_{[223,509]}$ calculation of TCNN. Therefore, the capability to handle track swapping and track drifting are important to avoid failure in the early phase of tracking. TCNN is unable to distinguish between two nearby similar objects because of its reliance on the stochastically generated samples without any predictive information on the motion model. On the other hand, MMFCNN integrates a basic motion model in calculating sample scores to give advantage to the smooth movement trajectory. Some output samples are shown in Figure 6.

E. COMPARISON TO TOP-50% TRACKERS IN VOT-TIR 2016 CHALLENGE

The performance results for 12 trackers that represent the top 50% and MMFCNN are shown in Table 3. The best overall tracker is SRDCFir which uses discriminative correlation filter with EO_1 of 0.300 as compared to the second best tracker, MMFCNN with EO_1 of 0.257. The good performance by SRDCFir can be attributed to its ability to maintain successful tracking in more test videos which is proved by having the best robustness of 1.12. However, MMFCNN is still the best FCNN-based tracker and one of the only

TABLE 3. Expected average overlap, accuracy, robustness and reliability results tested VOT-TIR 2016 dataset. EO_1 is obtained by using a range of [223, 509] (range used by VOT-TIR 2016 challenge) while EO_2 is obtained by using a range of [1, 1451].

Method	EO_1	EO_2	Ac	Ro	Re
SRDCFir	0.300	0.210	0.599	1.12	0.84
MMFCNN	0.257	0.184	0.543	1.63	0.82
TCNN	0.248	0.177	0.608	2.05	0.80
deepMKCF	0.246	0.155	0.609	3.12	0.70
MDNet-N	0.235	0.176	0.630	2.58	0.75
Staple-TIR	0.226	0.146	0.626	2.80	0.71
SHCT	0.221	0.145	0.593	2.76	0.73
Staple+	0.204	0.136	0.588	3.04	0.73
EBT	0.174	0.125	0.426	0.88	0.81
DSST2014	0.173	0.123	0.599	3.56	0.62
DPT	0.170	0.136	0.558	3.16	0.68
MvCF	0.158	0.116	0.531	3.12	0.69
MAD	0.148	0.118	0.560	3.44	0.70

TABLE 4. Performance comparison between MMFCNN and SRDCFir for test video 'ragged' that contains blur noise caused by abrupt movement change.

Method	EO	Ac	Ro	Re
MMFCNN	0.495	0.483	0.40	0.97
SRDCFir	0.075	0.764	10.00	0.38

two trackers with robustness below 2.0. SRDCFir utilizes gradient information which happened to model the object appearance better for TIR video, especially when there is a lack of texture information. This is in contrast to FCNN-based trackers which are not able to fully utilize the FCNN advantage because of the same reason. Interesting to note that MMFCNN works much better than SRDCFir in the situation of blur image caused by abrupt movement pattern as shown by results in Table 4. The Ro of SRDCFir is 10, which is 25 times higher than MMFCNN with $Ro = 0.4$. MMFCNN

is able to locate the object by using CNN appearance model with the help from the surrounding information, while SRDCFir cannot correlate the appearance model well and leads to many tracking failures. Furthermore, the performance of EBT tracker is significantly lower compared to MMFCNN with EAO_1 of 0.174. This is consistent with the situation in Figure 4, where objectness information alone is not enough to infer object localization. Hence, MMFCNN has integrated objectness information and stochastic sampling to better predict future location and size of the tracked object. Besides, the advantage of SRDCFir over MMFCNN is reduced if the full range EAO of the test videos is considered with 0.210 as opposed to 0.184. Figure 7 also indicates that TCNN works slightly better after frame sequence > 1000 .

V. CONCLUSION

In conclusion, MMFCNN performs the best among CNN-based trackers but still remains the second best tracker when compared to the SRDCFir. The lack of texture information in TIR modality does not fully exploit the advantage of convolutional layers. However, we found that MMFCNN works the best in the situation of blur image caused by abrupt movement changes as proved by the test video 'ragged'. Overall, MMFCNN has reduced the performance gap to the DCF-based trackers with a better sampling procedure and scoring models. Further improvement can be made by utilizing each convolutional layer information instead of the last layer only and applying smoothness function to readjust the output bounding box.

REFERENCES

- [1] C. Corsi, "Infrared: A key technology for security systems," *Adv. Opt. Technol.*, vol. 2012, Oct. 2012, Art. no. 838752.
- [2] V. Engert, A. Merla, J. A. Grant, D. Cardone, A. Tusche, and T. Singer, "Exploring the use of thermal infrared imaging in human stress research," *PLoS ONE*, vol. 9, no. 3, p. e90782, 2014.
- [3] N. I. Hristov, M. Betke, and T. H. Kunz, "Applications of thermal infrared imaging for research in aerocology," *Integr. Comparative Biol.*, vol. 48, no. 1, pp. 50–59, Jul. 2008.
- [4] A. G. Hart *et al.*, "Can handheld thermal imaging technology improve detection of poachers in African bushveldt?" *PLoS ONE*, vol. 10, no. 6, p. e0131584, 2015.
- [5] J. Cilulko, P. Janiszewski, M. Bogdaszewski, and E. Szczygielska, "Infrared thermal imaging in studies of wild animals," *Eur. J. Wildlife Res.*, vol. 59, no. 1, pp. 17–23, Feb. 2013.
- [6] M. Kristan *et al.*, "The visual object tracking VOT2015 challenge results," in *Proc. IEEE Int. Conf. Comput. Vis. Workshops (ICCVW)*, Dec. 2015, pp. 564–586.
- [7] M. Danelljan, A. Robinson, F. S. Khan, and M. Felsberg, "Beyond correlation filters: Learning continuous convolution operators for visual tracking," in *Computer Vision—ECCV*. Cham, Switzerland: Springer, 2016, pp. 472–488.
- [8] H. Nam and B. Han, "Learning multi-domain convolutional neural networks for visual tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 4293–4302.
- [9] H. Nam, M. Baek, and B. Han, (Aug. 2016). "Modeling and propagating CNNs in a tree structure for visual tracking." [Online]. Available: <https://arxiv.org/abs/1608.07242>
- [10] J. Son, M. Baek, M. Cho, and B. Han, "Multi-object tracking with quadruplet convolutional neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 3786–3795.
- [11] S. Schuster, P. Vernaza, W. Choi, and M. Chandraker, "Deep network flow for multi-object tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 2730–2739.
- [12] Q. Chu, W. Ouyang, H. Li, X. Wang, B. Liu, and N. Yu, "Online multi-object tracking using CNN-based single object tracker with spatial-temporal attention mechanism," in *Proc. IEEE Int. Conf. Comput. Vis.*, Aug. 2017, pp. 4846–4855.
- [13] M. A. Zulkifley and B. Moran, "Robust hierarchical multiple hypothesis tracker for multiple-object tracking," *Expert Syst. Appl.*, vol. 39, no. 16, pp. 12319–12331, 2012.
- [14] N. Wang and D.-Y. Yeung, "Learning a deep compact image representation for visual tracking," in *Advances in Neural Information Processing Systems*, C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, Eds. Red Hook, NY, USA: Curran Associates, 2013, pp. 809–817.
- [15] A. Torralba, R. Fergus, and W. T. Freeman, "80 million tiny images: A large data set for nonparametric object and scene recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 11, pp. 1958–1970, Nov. 2008.
- [16] S. Hong, T. You, S. Kwak, and B. Han, "Online tracking by learning discriminative saliency map with convolutional neural network," in *Proc. 32nd Int. Conf. Int. Conf. Mach. Learn. (ICML)*, vol. 37, 2015, pp. 597–606.
- [17] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 580–587.
- [18] M. Danelljan, G. Hager, F. S. Khan, and M. Felsberg, "Learning spatially regularized correlation filters for visual tracking," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2015, pp. 4310–4318.
- [19] B. Han, J. Sim, and H. Adam, "BranchOut: Regularization for online ensemble tracking with convolutional neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 3356–3365.
- [20] L. Wang, W. Ouyang, X. Wang, and H. Lu, "STCT: Sequentially training convolutional networks for visual tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 1373–1381.
- [21] C. Ma, J. Huang, X. Yang, and M. Yang, "Hierarchical convolutional features for visual tracking," in *Proc. IEEE Int. Conf. Comput. Vis.*, Jun. 2015, pp. 3074–3082.
- [22] L. Wang, W. Ouyang, X. Wang, and H. Lu, "Visual tracking with fully convolutional networks," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2015, pp. 3119–3127.
- [23] H. Lou, D. Wang, Z. Jiang, A. Men, and Y. Zhou, "Learning spatial-temporal consistent correlation filter for visual tracking," in *Proc. IEEE Int. Conf. Multimedia Expo Workshops*, Jul. 2017, pp. 501–506.
- [24] Y. Qi, S. Zhang, L. Qin, H. Yao, Q. Huang, J. Lim, and M.-H. Yang, "Hedged deep tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 4303–4311.
- [25] D. Wu, W. Zou, X. Li, and Y. Zhao, "Kernelised multi-resolution convnet for visual tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, Jul. 2017, pp. 2241–2248.
- [26] Z. Teng, J. Xing, Q. Wang, C. Lang, S. Feng, and Y. Jin, "Robust object tracking based on temporal and spatial deep networks," in *Proc. IEEE Int. Conf. Comput. Vis.*, Oct. 2017, pp. 1153–1162.
- [27] M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking," *IEEE Trans. Signal Process.*, vol. 50, no. 2, pp. 174–188, Feb. 2002.
- [28] G. Zhu, F. Porikli, and H. Li, "Beyond local search: Tracking objects everywhere with instance-specific proposals," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 943–951.
- [29] M. Kristan *et al.*, "The visual object tracking VOT2015 challenge results," in *Proc. IEEE Int. Conf. Comput. Vis. Workshop*, Dec. 2015, pp. 564–586.
- [30] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. 3rd Int. Conf. Learn. Represent.*, 2014, pp. 1–15.
- [31] L. Bertinetto, J. Valmadre, S. Golodetz, O. Miksik, and P. H. S. Torr, "Staple: Complementary learners for real-time tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 1401–1409.
- [32] D. Du, H. Qi, L. Wen, Q. Tian, Q. Huang, and S. Lyu, "Geometric hypergraph learning for visual tracking," *IEEE Trans. Cybern.*, vol. 47, no. 12, pp. 4182–4195, Dec. 2017.

- [33] Y. Li and J. Zhu, "A scale adaptive kernel correlation filter tracker with feature integration," in *Computer Vision—ECCV*. Cham, Switzerland: Springer, 2015, pp. 254–265.
- [34] M. Danelljan, G. Häger, F. S. Khan, and M. Felsberg, "Accurate scale estimation for robust visual tracking," in *Proc. Brit. Mach. Vis. Conf.*, M. Valstar, A. French, and T. Pridmore, Eds. Durham, U.K.: BMVA Press, 2014, pp. 1–11.
- [35] X. Li, Q. Liu, Z. He, H. Wang, C. Zhang, and W.-S. Chen, "A multi-view model for visual tracking via correlation filters," *Knowl.-Based Syst.*, vol. 113, pp. 88–99, Dec. 2016.
- [36] A. Lukežič, L. Č. Zajc, and M. Kristan, "Deformable parts correlation filters for robust visual tracking," *IEEE Trans. Cybern.*, vol. 48, no. 6, pp. 1849–1861, Jun. 2018.
- [37] M. Tang and J. Feng, "Multi-kernel correlation filter for visual tracking," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2015, pp. 3038–3046.
- [38] S. Becker, S. B. Krah, W. Hübner, and M. Arens, "Mad for visual tracker fusion," *Proc. SPIE*, vol. 9995, p. 99950L, Oct. 2016.
- [39] M. Felsberg *et al.*, "The thermal infrared visual object tracking VOT-TIR2016 challenge results," in *Proc. Eur. Conf. Comput. Vis. Workshops (ECCVW)*, Oct. 2016, pp. 824–849.
- [40] K. Chatfield, K. Simonyan, A. Vedaldi, and A. Zisserman, "Return of the devil in the details: Delving deep into convolutional networks," in *Proc. Brit. Mach. Vis. Conf.*, 2014, pp. 1–11.
- [41] O. Russakovsky *et al.*, "ImageNet large scale visual recognition challenge," *Int. J. Comput. Vis.*, vol. 115, no. 3, pp. 211–252, Dec. 2015.



current research interests are visual object tracking and medical image analysis.



MOHD ASYRAF ZULKIFLEY (M'18) received the B.Eng. degree in mechatronics from International Islamic University Malaysia in 2008 and the Ph.D. degree in electrical and electronic engineering from The University of Melbourne in 2012. He is currently a sponsored Researcher with the Department of Computer Science, University of Oxford. He is also an Associate Professor with the Faculty of Engineering and Built Environment, Universiti Kebangsaan Malaysia. His current research interests are visual object tracking and medical image analysis.

NIKI TRIGONI is currently a Professor with the Department of Computer Science, University of Oxford. She is also the Director of the EPSRC Centre for Doctoral Training on Autonomous Intelligent Machines and Systems, where she leads the Cyber Physical Systems Group. Her research interests lie in intelligent and autonomous sensor systems with applications in positioning, health-care, environmental monitoring, and smart cities.

• • •