VISUAL TRACKING UTILIZING OBJECT CONCEPT FROM DEEP LEARNING NETWORK

Changlin Xiao[a], Alper Yilmaz[a], Shirui Li[a,b]

[a] Photogrammetric Computer Vision Laboratory, The Ohio State University, USA- (xiao.157, yilmaz.15)@osu.edu
[b]Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China - lishiruilishirui@gmail.com

KEY WORDS: Visual Tracking, Deep Learning, Object Concept, Heat Map, Image Understanding

ABSTRACT:

Despite having achieved good performance, visual tracking is still an open area of research, especially when target undergoes serious appearance changes which are not included in the model. So, in this paper, we replace the appearance model by a concept model which is learned from large-scale datasets using a deep learning network. The concept model is a combination of high-level semantic information that is learned from myriads of objects with various appearances. In our tracking method, we generate the target's concept by combining the learned object concepts from classification task. We also demonstrate that the last convolutional feature map can be used to generate a heat map to highlight the possible location of the given target in new frames. Finally, in the proposed tracking framework, we utilize the target image, the search image cropped from the new frame and their heat maps as input into a localization network to find the final target position. Compared to the other state-of-the-art trackers, the proposed method shows the comparable and at times better performance in real-time.

1. INTRODUCTION

Single-target tracking is the most fundamental problem in many vision tasks such as surveillance and autonomous navigation. In the past, there have been many successful object trackers that use features and appearance descriptors (Felzenszwalb et al., 2010, Kalal et al., 2012). These trackers can be categorized as either generative or discriminative which use appearance-based models to distinguish the target from the background. Researchers have also introduced sophisticated features and descriptors (Kim et al., 2015, Rublee et al., 2011, Zhang et al., 2014), yet there are still many issues in practical applications. The main limitation of these low-level hand-crafted features is that they only address the texture of the object which may frequently change.

Recently, Deep Neural Networks (DNNs) have demonstrated promising performance in tasks like image classification (Krizhevsky et al., 2012), object detection (Redmon et al., 2015, Ren et al., 2015) and segmentation (Tsogkas et al., 2015). Different from the low-level features, the DNNs, especially the convolutional neural networks (CNNs) have been shown to learn high-level semantic information of the object. After training on large-scale datasets like ImageNet (Deng et al., 2009), it has been shown it can learn distinctive information for different object categories. Motivated by this fact, many CNN based trackers have been proposed (Wang et al., 2015, Wang and Yeung, 2013). However, most of them have only consider the CNNs feature extraction capability and use the traditional methods to do the tracking.

For humans, an object is not just about its appearance at limited view angle, but its concept which may include every appearance about it. Since the CNN has the capability to learn a general semantic representation of objects, we think it can learn some concept as well. Inspired by this idea, we propose a visual tracker which adopts the one-thousand-object concepts learned from ImageNet instead of directly modeling the appearance of the novel target. The basic idea is that, with a pre-trained classification network, a novel target can be modeled as a combination of several existing categories. In another word, we calculate and define

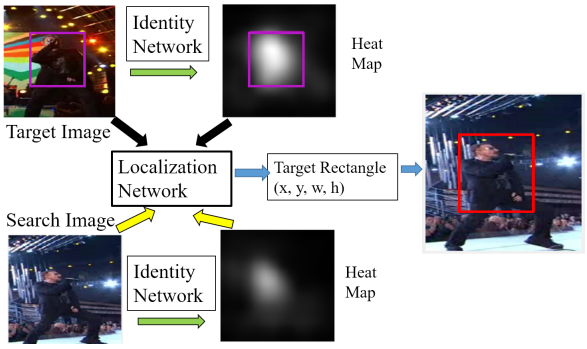


Figure 1. The tracking flow of proposition. In the proposed tracking framework, there are two different networks: one is for the target identity concept recognition and the other is for the target location detection.

one more set of weights for the last full connection in the network to define the novel target. This is not an online training and will not change the parameters of the pre-defined network; it uses the learned high-level features to construct a new object concept which is used to define the target. After getting the concept, a heat map, which is generated by fusion of the last convolution feature maps with the concept, is used to highlight the target in new frames. Then, the target image, the search image and the heat maps are used to find the final target location by the localization network. The flow diagram can be seen in Fig 1. Since we use a pre-trained network, there is no on-line training needed which makes the tracker work at high frequency.

The main contributions of this paper include: (1) to the best of our knowledge, it is the first paper that proposes using object concept from CNNs instead of learning feature appearance in visual tracking; (2) offers an efficient way to define target as combination of pre-learned object categories without on-line training; (3) combines the heat map and target image to form a cascaded tracking network that works at high frequency.

This contribution has been peer-reviewed. The double-blind peer-review was conducted on the basis of the full paper.
doi:10.5194/isprs-annals-IV-1-W1-125-2017

The rest of the paper is organized as follows: The related work is in Section 2, which followed by the proposed tracking methodology in Section 3; The experiments are given in Section 4; Finally, we conclude in Section 5.

2. RELATED WORK

Appearance feature based trackers. For the past several decades, it has been common to use a form of well-studied target representations, such as points, lines, and target patches in tracking (Yilmaz et al., 2006, Zhang et al., 2014, Kim et al., 2015). Despite reported success in some benchmark sequences, these methods are sensitive to noise and background clutter. In order to overcome this, new models and features were proposed (Hong et al., 2015b). In order to address the background clutter, researchers use discriminative information generated from the target and non-target regions. Most of these methods train a classifier and choose discriminative descriptors (Sui et al., 2015). For the classifier training, Multiple instance learning (MIL) use a set of ambiguous positive and negative samples which in part belong to the target or the background. Similarly, in TLD, the authors use positive and negative experts to improve the learning process (Kalal et al., 2012). MEEM uses SVM to detect target based on the entropy of the score function which uses an ensemble of experts about historical snapshots during tracking (Zhang et al., 2014). DLSSVM uses dual linear SSVM to enable fast learning the features with more robustness and less computation cost (Ning et al., n.d.).

Neural network trackers. Since the neural networks have shown breakthrough performance in object classification, it has also been adopted in tracking. For tracking, the network is typically used as a feature extractor (Wang et al., 2015, Zhang et al., 2016, Hong et al., 2015a). These features have higher semantic information than low-level features used in the past. In (Wang et al., 2015, Ma et al., 2015), the authors use the characteristics of lower and higher feature maps in the network to represent target at the category and individual levels. In (Zhang et al., 2016), previous target patches are stored in a filter bank to do a convolution operation at new frames to highlight the object location. In (Hong et al., 2015a), sampled feature maps are classified by SVM to generate a saliency map. More recently, a number of studies use Recurrent Neural Networks (RNNs) for visual tracking (Bertinetto et al., 2016, Held et al., 2016, Chen and Tao, 2016, Tao et al., 2016). In (Held et al., 2016), Held et. al introduced a tracker which has a network with siamese architecture. The input to their system are target and search images and the output is the bounding box within the search image. However, it is based on the previously tracked target, if there is a drift or serious appearance change, the tracking fails. In (Bertinetto et al., 2016), siamese architecture with a different output which is a confidence map that shows the similarities of the corresponding sub-windows. Unlike the sliding window methods, this siamese network can directly generate all possible locations' similar scores by only one scan; but it handles the scale variations by repeated estimations.

Many of the aforementioned trackers predict a heat map (or a confidence map) that is generated by a correlation filter (Zhang et al., 2016), a sparse combination of feature map (Wang et al., 2015) and direct use of a convolutional network (Bertinetto et al., 2016). However, their limitation is that they only consider the texture similarity which may fail when appearance changes. In (Zhou et al., 2015), Zhou et. al, revised the CNNs to demonstrate the localization ability of feature map trained on image-level labels. They use global average pooling on the last convolutional feature maps for a fully-connected layer to produce category output. For a specified class, the connection weights in the last layer are used to generate an activation map based on the last covolutional feature maps. This activation map can highlight the structure or region of the image that belongs to the given class. Their method is used to detect objects which have been pre-trained and the accuracy is critically dependent on the classification quality. However, in visual tracking, a novel target may not always belong to one of the pre-trained object categories and there also may not be enough data for a new training. So, it is not preferred to directly use this heat map for a tracking prediction.

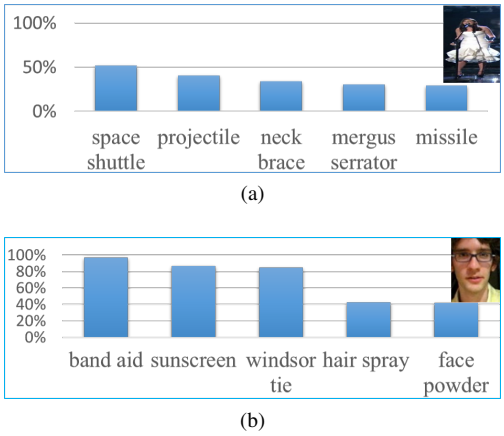


Figure 2. The classification results of TB50 target using the GoogLeNet. The vertical axes is the percentage of the target identified as different categories.

3. METHODOLOGY

3.1 Classification on Unknown Target

The proposed tracking idea is based on the assumption that the CNNs can learn high-level object category concepts. In order to test this assumption, we conducted a number of experiments using the tracking dataset TB50 (Wu et al., 2013) with the GoogLeNet (Szegedy et al., 2015). GoogLeNet originally designed to perform classification of 1000 object categories in the ImageNet challenge. We note that most of the targets in TB50 are not labeled in the dataset. Our test is to verify if the GoogLeNet can consistently classify TB50 targets that it has not learned before as one or several categories.

Based on the groundtruth, we crop targets from all image sequences and classify each cropped image with the GoogLeNet network.. We record the top 5 classes for each classification. We record the top 5 categories as the target classification results. Each category's score is estimated as the ratio of the recorded time and the index of the target image. Two of the results are shown in Fig. 2. Since the network is not trained with the labels like standing body or the face, it classifies the targets as different objects. In the "singer sequence", the target classified as space shuttle with the highest score. From this perspective, the network learns the novel object semantics despite scale, illumination and appearance distortions. However, in the image (b), the human face is identified as band-aid, sunscreen, and Windsor-tie with high scores. From the texture point, the face has nothing similar with them, however, all these classified categories are related to skin or face. Hence, when the network finds a band-aid, or the similar texture object, it may actually also finds a face.

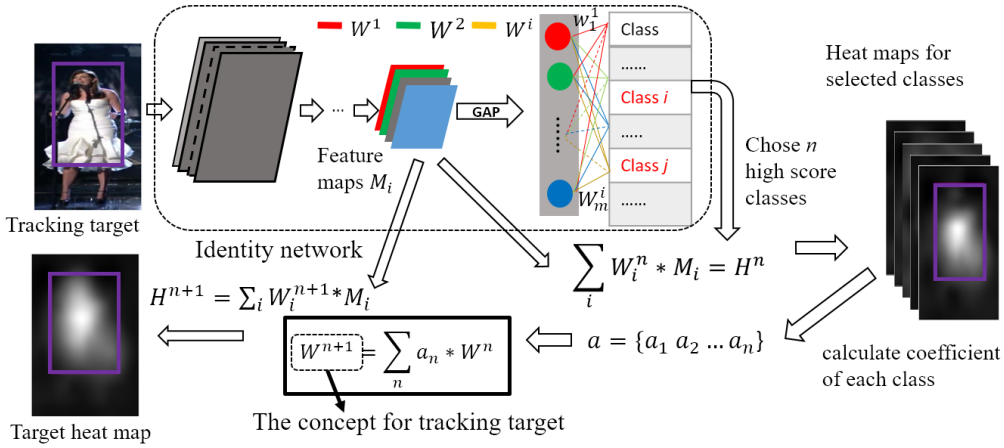


Figure 3. The new target concept estimation. Given a target, we can generate a number of heat maps for top n classes that the target may belong to like the red classes. Then based on the ground truth on the heat maps, the coefficients for the combination of these classes' weights can be calculated to generate the new concept for the target.

We also consider the repeatability of this pre-learned semantics. For that purpose, we choose the top 5 classified categories in the previous test as "true classes" that represent a target and for each tested target, if one of the top 5 highest score classes belongs to the selected representative classes, then we mark the classification as correct. The classification accuracy is the ratio of correctly classified against the total number. We have consistently observed that for almost every sequence, the classification accuracy reach 100 percent, which can be seen in the supplemental materials. The results suggest that the target has semantic features that spread along the 5 classes, and during the sequence, the target will have high response to these semantic features.

3.2 The Target Concept

Before we explain how to generate a new target semantic concept, we will first introduce the semantic concept, then define how to generate heat map for specified object category with the concept.

During training, the parameters in the network are adjusted to make sure the output have the same labels. In our approach, we separate the parameters in the network into two. One is used to extract high-level semantic features which are the weights of filters in convolution layers. The other one are the weights of the connector from high-level semantic features to object categories in the full connection layers. The convolution parameters decide how to extract features while the full connection parameters decide how to use these features. So, from this perspective, the object category score actually is dependent on the weights that decide the connections to the last convolution feature maps in the network. In other words, the object category is defined as a combination of these high-level features, and the connection weights decide this combination. Hence, we define the concept of an object category as a set of weights that connects the category and the last convolution feature map.

Even when the full connection layers lose the spatial information of the high-level features, many classification networks have demonstrated a remarkable object localization ability (Zhou et al., 2015, Oquab et al., 2015, Cinbis et al., 2016). In (Zhou et al., 2015), the net's last convolution layer is replaced by a global average pool (GAP) and the full connection layer is reduced to one to make the connection between category and high-level features very simple. Following the results of these studies, we combine all the last feature maps by the category weights and generate a class specified heat map (Fig. 3 shows an example). The heat map can highlight areas that include the high-level features which are trained as components of a given category.

It is impossible to train a network on all possible targets. Considering the conjecture we made in the last section, unknown target shares many semantic features that were learned from other objects, we use this fact to extract features from CNN and combine them to learn and define new targets. Unlike on-line learning, we don't change weights in the network. Instead, we select a new combination of the feature maps to generate the concept of a novel target. In order to do this, one can use the idea of sparse coding to generate a new concept with all high-level feature maps, but the process is time consuming and what's more important, the method converges to the same as appearance based tracking method instead of the target conception. The sparse coding offers a set of weights that make the combination of the feature maps more similar to the appearance of the target. For example, if the target is a cat face, the spare code only generate a concept about the appearance of a cat's face which is variant even totally different in the following frames. In order to avoid this, we weight the feature maps at the category level. The cat face, which the network does not know before, may be classified as dog or rabbit. On the class level, the concepts about the cat will include dog's body and limbs, rabbit' nose and tail, which make the concept more sensible, especially when the target cat shows his arms and body in the next frames. Based on this consideration, we treat the connections between the feature map and the class category as one set for each category and we do not change it since it's already been trained for the category. Let:

$$W_i^n = \{w_1^n, \quad w_2^n, \quad \dots\dots \quad , w_m^n\}, \tag{1}$$

represent the n^{th} category weights for all m feature maps in the last convolutional layer. Then the goal is to present the new object as estimating a set of coefficients that combine W^n as:

$$W^{n+1} = \sum_n a_n * W^n. \tag{2}$$

Here, we use an efficient way to estimate the coefficients while

strengthening the same feature parts and weakening the difference. We first use the network to get classification results $Score$ and weights W_i^n that show the mapping of all feature maps for each category. Then we generate a heat map for each high score category as:

$$H^n = \sum_i W_i^n * M_i. \quad (3)$$

where M_i = Feature map from last layer

Since we know the ground truth of the target during the tracking, we can evaluate how good each heat map is for the target by measuring the values inside the target area against the outside as:

$$a_n = (V_{inside}(H^n) - V_{outside}(H^n))/V(H^n), \quad (4)$$

where a_n = final combination coefficient of each category n and $V(H^n)$ = sum of all values in the heat map

Using this approach, we generate a set of connection weights to represent the target concept from learned object categories without online training.(see Fig. 3 for target concept generation).

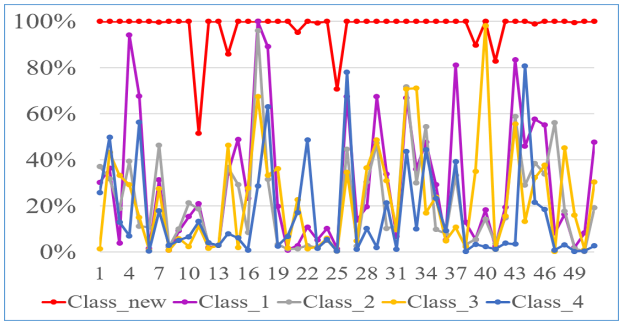


Figure 4. The classification results of the target in TB50. The vertical axis is the percentage that the target classified as different class in all 51 tracking sequence.

In order to test the performance of the new target concept, we conducted another set of experiments with the same TB50 dataset. From the initial target image, we generate the new concept as well as the other four top classified categories as comparison set. Then we use the new concept to generate the 1001th category. For all target images in the sequence, and we test if it belong to this 1001th new category, we mark the result as correct if it belongs to the top 5 classified categories. The results can be seen in Fig. 4. As can be observed, except six in fifty-one tracking sequences, all the sequences have scores close to 100 percent, and it is far more accurate than any other class in the initial frame. That suggests that even we only get the tracking target concept from the initial frame, the network can recognize the target in the following frames as this newly defined class.

3.3 Heat Map for Target Localization Prediction

Since heat map can highlight the interesting parts in the input image for a specified class, it is helpful for tracking. We can generate the concept of the target in the initial frame, and for following

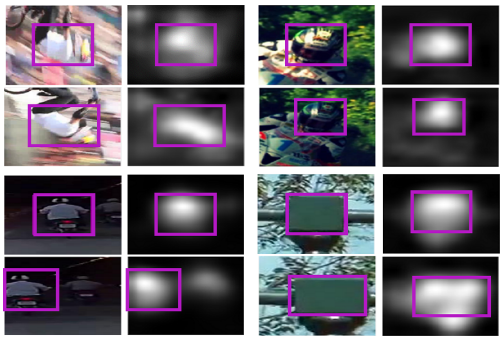


Figure 5. The heat map for prediction. In each sub four-images, the left top is the target image with ground truth to generate the heat map and calculate the target concept also the initial score of the concept (top left). The lower left image is the search image, and the right bottom image is the heat map generated by the concept from the target image. The ground true in search image is used to calculate the predict score.

frames, we extract their high-level feature maps, and based on the target concept, generate heat maps to estimate the location of the target. To test the ability of the heat map in prediction, we evaluated if the heat maps generated from the concept helps to localize the target in the new frame.

In this evaluation, we use the dataset from ALOV300+ (Smeulders et al., 2014) which has 314 image sequences. In this dataset, approximately every 5th frame of each sequence has a label to locate the target. During the test, we randomly select two frames from a random sequence which has more than one label. In the first image, the target is cropped as target template with some background texture. Using the sample generation idea from (Held et al., 2016), we randomly shift and scale the target in the second image as search image to simulate the motion of the target and camera simultaneously. In tracking, we generally cropped a search image in the new frame based on the target's previous location instead to track the target in the whole image. Hence, we set the image window size both as two times as the target rectangle. Also, the network need some contextual texture of the target to help the classification and generate the concept. We don't input the whole image into the network. We test a number of sizes for the input image and finally select 1.6 time of the target rectangle as best. In addition, we choose $n = 100$ as the number of the top classes that are chosen to generate the new concept in (2). The parameter selection test can be seen in the supplemental materials. In Fig. 5, we show some of the prediction results. As can be seen, the heat map predicts the target location in new frames. Even in the case of appearance blurring and rotation (left top), illumination changes (right top), translation (left bottom), scale change (right bottom), the heat map still can highlight the target area well. The left bottom example also shows us that, the learned concept is category based, it will highlight all the object that belong to the category. So, if there are similar objects in the image, the heat map will highlight all of them. To quantitatively measure the highlighting ability of the heat map, we use (4) to score the heat map for both the target image and the search image. Since the search image size is four times as the target, with even distribution (which we consider as no predictive ability), the score will be -0.5 and for the best prediction, the score is 1. In order to centralize the score, and make positive value as good prediction, negative as bad ones, we scale them from 0 to 1 by:

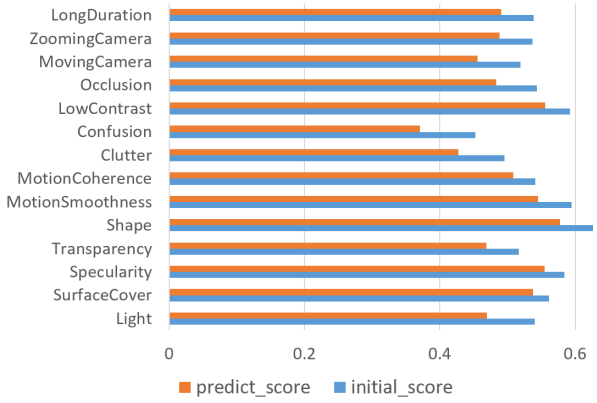


Figure 6. The predict and initial score in different situations.

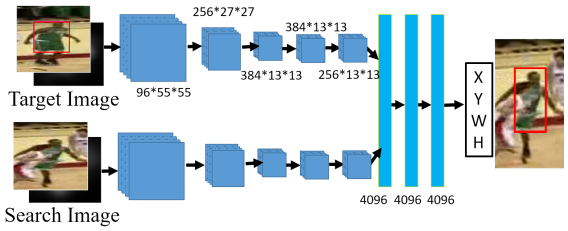


Figure 7. The architecture of localization network.

$$S_{scale} = (a_n + 0.5) * 2/3. \quad (5)$$

In order to report results, we randomly choose 500 image pairs in each image category in the dataset and average the scores as shown in Fig. 6. In the figure, the blue bars are the initial scores of the new concept on the target image, and the orange bars are predicted scores on the search image. As we see, the predict scores are proportional to initial scores in all sequence categories. That implicates that we can use the initial score to estimate the prediction ability of the heat map. If the initial score is very low, we don't use the heat map to predict the target. Additionally, for different scenarios, the confusion has a low score, when there is background clutter, transparency and moving camera. In these situations, the targets are either affected by the background or the appearance is not stable which makes it hard to distinguish the targets. Even in such challenging situations, the prediction score is still higher than 0.35 which suggest that algorithm can mark the target region successfully.

3.4 The Localization Network

The use of concept alone is not adequate to locate the target, the detailed appearance in relation to the concept is also important. Hence, we use both the target image and search image with their corresponding heat maps together as two sets of $height * width * 4(R, G, B, HeatMap)$ data blocks to feed to a siamese localization network to get the tracking results. The architecture of the localization network is similar to CaffeNet(Jia et al., 2014) which can be seen in Fig. 7.

To train the localization network, we keep generating samples from the ALOV300+ and ImageNet 2015 (Russakovsky et al., 2015). For the ALOV300+ data, the overlap sequences that are also in our experiment TB50 dataset are removed and the samples are generated as described in section 3.3, but, we don't only select continuous frames. Since we want the localization network to find target location by both the texture and concept information, the search image may not always be similar to the target image which is important in the training. For the static images in ImageNet, the search images are randomly cropped with scale and translation changes respect to the target location. Considering the changes are smooth in most cases, the cropping of location and size follows the Laplace distribution given by:

$$f(x|u,b) = \frac{1}{2b} exp(-\frac{|x-u|}{b}), \quad (6)$$

where $u = 0$, for both scale and translation changes
for scale changes, $b = b_s = 1/5$
for translation change, $b = b_t = 1/15$,

Also, we enforce the scale change to be less than ± 0.4 and the center of the translated target is still in the search image similar to the work in (Held et al., 2016). The ImageNet dataset is mainly used to teach localization network to find object boundary and the smooth motion as complementary data in the limited ALOV300+ dataset.

3.5 Tracking Framework

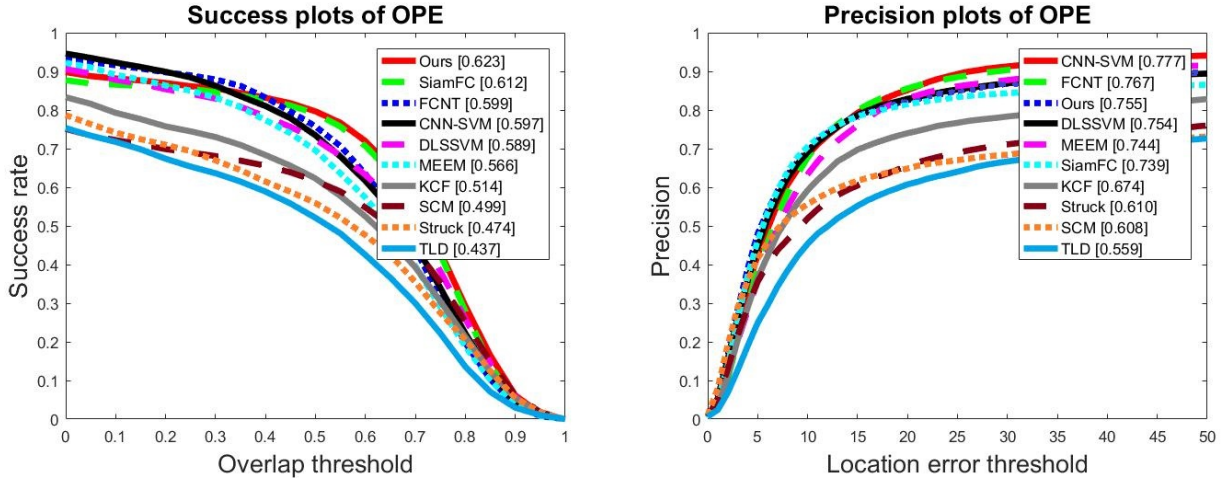
For single object tracking, the target is selected at the first frame with an initial rectangle. First, we define the target and get its concept as described above. For tracking, the target image is cropped as twice as the initial rectangle. When the new image comes, a search image is cropped based on the previous target location with twice the rectangle size. After that, the heat maps of target and search images are feed to the localization network to find the final target position.

Since the initial target image is the only example appearance of the target and the localization network performance is better than if we use the appearance from tracked target in the previous frame, we fix the initial target as target image in the localization network. By fixing the initial target, the tracker can efficiently avoid the drift problem. But, when a similar object comes nearby, the concept will highlight those regions and confuse the tracker. So, during the tracking, the target concept (the connecting weights of the last layer) needs to be updated:

$$Concept_n = a * Concept_{old} + (1-a) Concept_{current}, \quad (7)$$

where a = the learning rate
$C_{current}$ = current target concept

Performing this update will suppress the concept which contain the confusing regions. Hence, in the heat map, the confusion region will not be highlighted. Since the prediction score is proportional to the initial score, we can use (4) to estimate the tracking quality. If the tracking quality score is lower than a set threshold Λ, then we assume there are too many distractors, we keep the target location unchanged to further avoiding drifting.



(a) (b)

Figure 8. Average success plot (left) and precision plot (right) for the OPE on TB50.

4. EXPERIMENTS

4.1 Experimental Setup

Beside the parameters discussed above, we set the learning rate $a = 0.5$ and prediction quality check threshold $\Lambda = 0.3$ for all experiments. We use Caffe as the deep learning tool (Jia et al., 2014). We implement the tracking algorithm in Matlab and run it with the NVIDIA graphic card GeForce GTX 950 at average 32 FPS.

We test the performance of the proposed approach using a large dataset referred as the online tracker benchmark (TB50) (Wu et al., 2013). The performance scores are provided from the one-pass evaluation (OPE) toolkit. There are two measurements in the evaluation: Precision plot and success plot. The precision plot is used to measure the distance between tracking results and ground truth. The success plot is used to evaluate the overlap score of them. For more details, please refer to (Wu et al., 2013). We also compared the proposed tracker on the benchmark with other 35 popular trackers including 29 trackers in (Wu et al., 2013) and KCF (Henriques et al., 2015), MEEM (Zhang et al., 2014), DLSSVM (Ning et al., n.d.), and some more recent deep learning based trackers CNN-SVM (Hong et al., 2015a), SiamFC-5 (Bertinetto et al., 2016) and FCNT (Wang et al., 2015).

4.2 Experimental Result

The tracking results from TB50 are given in Fig. 8. More results for different attributes are given in the supplemental materials. As we can see, the proposed approach is one of the best trackers compared to all others. Especially in fast motion, motion blur, in and out plane-rotation and illumination change sequences. We believe the excellent performance is mainly because the tracking target concept from deep network has very high semantic information which is invariant to rotation, illumination, translation and scale changes. Use of the concept generated heat map to highlight target area is robust to this appearance changes as shown in both Fig. 5 and Fig.9. In this figures, when the target undergoes serious appearance changes, the concept heat map can still find it correctly. As one can expect, the occlusion and clutter sequences reduce the performance. This may due to the fact that

Figure 10. Illustration of the occlusion scenarios.

when the sequence contains seriously confusing situations, both the appearance and concept clue fail to work. We will discuss more in the coming paragraphs.

Compared to other trackers, our methods shows excellent performance in overlap score but a slightly lower performance at precision. This can be attributed to the fact that the localization network can find the target boundary and make the output rectangle fit the target very well. But in complicated scenarios, the tracker may lose the target completely which reduces the precision score. Compared other all trackers, ours works in real-time (32 FPS). The DLSSVM runs at 5.4 FPS, FCNT runs at 3 FPS, CNN-SVM doesn't show their time permanence in their paper but should be far away from real-time. Only the SiamFC compares to ours at 58 FPS, but with a better hardware(GTX Titan X) that is more powerful than ours(GTX 950).

Occlusion: Most realistic sequences may contain target occlusions. While the trackers based on a holistic target model may have some problems, the proposed target concept still works under the partial occlusions. Even when there is only a small part of the target is visible, the concept still highlights visible parts. Based on the texture information, the target can be found after recovering from the occlusion. As we can see in Fig. 10, the tracked player is partially occluded by another player. The concept can still highlight the target part in the heat map. And based on the texture information, the target player can be located correctly.

Background clutter: In the case when the background and target have similar descriptors, the tracking task becomes challenging. If we only use the unchanged concept to detect the target, it is

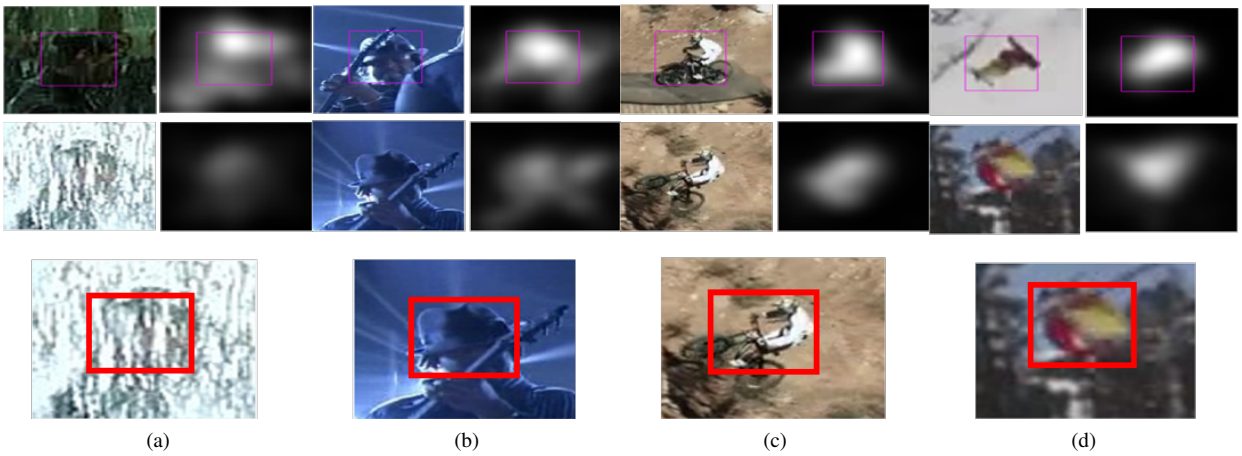


Figure 9. Illustration of good tracking with (a) significant illumination change, (b) texture and edge change, (c) deformation and out plane rotation, (d) fast motion and motion blur.

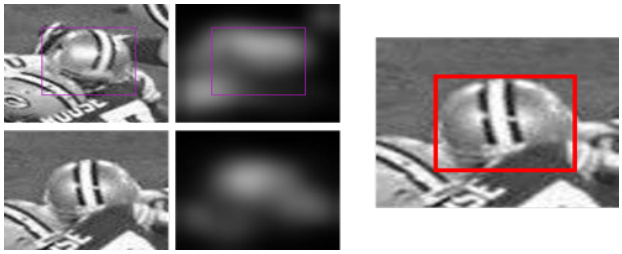


Figure 11. Illustration of suppressing distractions.

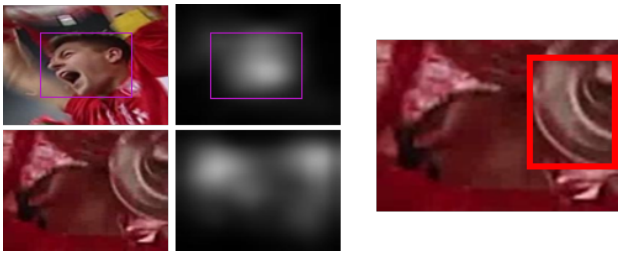


Figure 12. Illustration of concept and appearance information both lost the target.

hard to remove the distractors, as shown in the left bottom image in Fig. 5. However, during the tracking, the concept is updated which can suppress the similar concept and increase the different ones by our concept generation process. When a similar object is observed, since it is not in the target rectangle, their similar concepts will be treat as outlier and is suppressed in the next frame. Fig. 11 illustrates an example. In the first row, the concept about the helmet is highlighted for both players. However, after the concept updating, some similar concepts were suppressed in the heat map to help identify the true target.

The Lost Target: Our tracker may have problems in some complicated situations. The tracking is based on the concept and the target appearance information. When they both fail, they tracker may fail. In Fig. 12, the concept captured from the upper image are mainly about the shape and the edges of the face which is completely lost in the lower image. Also, the face texture disappeared in the search image and leaves no clue to find the target. Besides that, when a new similar object appears, the concept will be weaker to highlight the target area. At the same time, if the target's appearance undergoes changes, both of the two tracking cues will be lost. Since proposed tracker does not update in such situations, the target can be retrieved after it reappears.

5. CONCLUSION

In this paper, we introduce a new target tracking method that uses the concepts learned from deep learning network to represent a novel target. The target concept is generated by combining high-level features from the deep network pre-trained on unrelated objects. These high-level features are invariant to scale, rotation and translation changes, even a serious deformation. Also, the concept and high-level features can be used to generate a heat map which highlights potential target area in the search map. In the tracking phrase, the initial target image is used as constant texture information in a siamese localization network. Regardless of only using the constant initialization target exampler, our method shows good performance in the case when the object appearance undergoes significant changes.

REFERENCES

Bertinetto, L., Valmadre, J., Henriques, J. F., Vedaldi, A. and Torr, P. H., 2016. Fully-convolutional siamese networks for object tracking. *arXiv preprint arXiv:1606.09549*.

Chen, K. and Tao, W., 2016. Once for all: a two-flow convolutional neural network for visual tracking. *arXiv preprint arXiv:1604.07507*.

Cinbis, R. G., Verbeek, J. and Schmid, C., 2016. Weakly supervised object localization with multi-fold multiple instance learning.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K. and Fei-Fei, L., 2009. Imagenet: A large-scale hierarchical image database. In: *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, IEEE, pp. 248–255.

Felzenszwalb, P. F., Girshick, R. B., McAllester, D. and Ramanan, D., 2010. Object detection with discriminatively trained part-based models. *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 32(9), pp. 1627–1645.

Held, D., Thrun, S. and Savarese, S., 2016. Learning to track at 100 fps with deep regression networks. In: *European Conference Computer Vision (ECCV)*.

Henriques, J. F., Caseiro, R., Martins, P. and Batista, J., 2015. High-speed tracking with kernelized correlation filters. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 37(3), pp. 583–596.

Hong, S., You, T., Kwak, S. and Han, B., 2015a. Online tracking by learning discriminative saliency map with convolutional neural network. *arXiv preprint arXiv:1502.06796*.

Hong, Z., Chen, Z., Wang, C., Mei, X., Prokhorov, D. and Tao, D., 2015b. Multi-store tracker (muster): A cognitive psychology inspired approach to object tracking. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 749–758.

Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., Guadarrama, S. and Darrell, T., 2014. Caffe: Convolutional architecture for fast feature embedding. *arXiv preprint arXiv:1408.5093*.

Kalal, Z., Mikolajczyk, K. and Matas, J., 2012. Tracking-learning-detection. *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 34(7), pp. 1409–1422.

Kim, H.-U., Lee, D.-Y., Sim, J.-Y. and Kim, C.-S., 2015. Sowp: Spatially ordered and weighted patch descriptor for visual tracking. In: *Proceedings of the IEEE International Conference on Computer Vision*, pp. 3011–3019.

Krizhevsky, A., Sutskever, I. and Hinton, G. E., 2012. Imagenet classification with deep convolutional neural networks. In: *Advances in neural information processing systems*, pp. 1097–1105.

Ma, C., Huang, J.-B., Yang, X. and Yang, M.-H., 2015. Hierarchical convolutional features for visual tracking. In: *Proceedings of the IEEE International Conference on Computer Vision*, pp. 3074–3082.

Ning, J., Yang, J., Jiang, S., Zhang, L. and Yang, M.-H., n.d. Object tracking via dual linear structured svm and explicit feature map.

Oquab, M., Bottou, L., Laptev, I. and Sivic, J., 2015. Is object localization for free?-weakly-supervised learning with convolutional neural networks. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 685–694.

Redmon, J., Divvala, S., Girshick, R. and Farhadi, A., 2015. You only look once: Unified, real-time object detection. *arXiv preprint arXiv:1506.02640*.

Ren, S., He, K., Girshick, R. and Sun, J., 2015. Faster r-cnn: Towards real-time object detection with region proposal networks. In: *Advances in neural information processing systems*, pp. 91–99.

Rublee, E., Rabaud, V., Konolige, K. and Bradski, G., 2011. Orb: an efficient alternative to sift or surf. In: *Computer Vision (ICCV), 2011 IEEE International Conference on*, IEEE, pp. 2564–2571.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A. C. and Fei-Fei, L., 2015. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)* 115(3), pp. 211–252.

Smeulders, A. W., Chu, D. M., Cucchiara, R., Calderara, S., Dehghan, A. and Shah, M., 2014. Visual tracking: An experimental survey. *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 36(7), pp. 1442–1468.

Sui, Y., Tang, Y. and Zhang, L., 2015. Discriminative low-rank tracking. In: *Proceedings of the IEEE International Conference on Computer Vision*, pp. 3002–3010.

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V. and Rabinovich, A., 2015. Going deeper with convolutions. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–9.

Tao, R., Gavves, E. and Smeulders, A. W. M., 2016. Siamese instance search for tracking. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

Tsogkas, S., Kokkinos, I., Papandreou, G. and Vedaldi, A., 2015. Semantic part segmentation with deep learning. *arXiv preprint arXiv:1505.02438*.

Wang, L., Ouyang, W., Wang, X. and Lu, H., 2015. Visual tracking with fully convolutional networks. In: *Proceedings of the IEEE International Conference on Computer Vision*, pp. 3119–3127.

Wang, N. and Yeung, D.-Y., 2013. Learning a deep compact image representation for visual tracking. In: *Advances in neural information processing systems*, pp. 809–817.

Wu, Y., Lim, J. and Yang, M.-H., 2013. Online object tracking: A benchmark. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2411–2418.

Yilmaz, A., Javed, O. and Shah, M., 2006. Object tracking: A survey. *Acm computing surveys (CSUR)* 38(4), pp. 13.

Zhang, J., Ma, S. and Sclaroff, S., 2014. Meem: Robust tracking via multiple experts using entropy minimization. In: *Computer Vision–ECCV 2014*, Springer, pp. 188–203.

Zhang, K., Liu, Q., Wu, Y. and Yang, M.-H., 2016. Robust visual tracking via convolutional networks without training. *IEEE Transactions on Image Processing* 25(4), pp. 1779–1792.

Zhou, B., Khosla, A., Lapedriza, A., Oliva, A. and Torralba, A., 2015. Learning deep features for discriminative localization. *arXiv preprint arXiv:1512.04150*.