Learned Filters for Object Detection in Multi-object Visual Tracking

Victor Stamatescu[a], Sebastien Wong[b], Mark D. McDonnell[a], and David Kearney[a]

[a]University of South Australia, Mawson Lakes, South Australia, Australia;
[b]Defence Science & Technology Group, Edinburgh, South Australia, Australia

ABSTRACT

We investigate the application of learned convolutional filters in multi-object visual tracking. The filters were learned in both a supervised and unsupervised manner from image data using artificial neural networks. This work follows recent results in the field of machine learning that demonstrate the use learned filters for enhanced object detection and classification. Here we employ a track-before-detect approach to multi-object tracking, where tracking guides the detection process. The object detection provides a probabilistic input image calculated by selecting from features obtained using banks of generative or discriminative learned filters. We present a systematic evaluation of these convolutional filters using a real-world data set that examines their performance as generic object detectors.

Keywords: visual tracking, feature learning, transfer learning

1. INTRODUCTION

Multi-object visual tracking involves estimating the trajectories of multiple objects in an image sequence. The work described in this paper has the additional goal of tracking multiple *types* of interesting objects in a way that allows the decision on *what is an interesting object* to be postponed until some later classification stage. Under this *track everything* approach, we have previously developed a general, automated multi-object tracking system[1] that can self-initialize on all objects in the scene and adapt to changes in their appearance or motion.

The role of object detection in multi-object visual tracking is to supply the tracking stage with information on the presence of objects across the scene. A visual tracking algorithm can be described as *generative* if it only models the object appearance. Alternatively, *discriminant* tracking algorithms cast the detection of the object from its local background as a two-class classification problem. Discriminant trackers have achieved excellent performance on recent single object tracking benchmarks.[2,3] The work presented here relies on the discriminant tracking approach proposed by Collins *et al.*,[4] where the detection stage yields a *saliency* map reflecting the likelihood that each pixel belongs to the object. In our system this detection map serves as the input to a probabilistic state space hierarchy[5] that is learned online to perform object tracking and guide future detections, following the *track-before-detect* paradigm.[6]

An open question in computer vision, including visual tracking, relates to the design of robust image *feature* representations (e.g.[7–10]) that can enhance performance on any vision task. In this work we consider only bandpass (e.g. Gabor[7]) filters that carry out linear transformations of the image via convolution. Recent experimental work in this area shows that the features learned in the multiple layers of Convolutional Neural Networks[11] (CNN) can act as generic image descriptors in a wide range of tasks involving object recognition[12] and object detection.[13] Convolutional features transferred from CNNs have also been used in place of hand-crafted features to enhance the state-of-the-art single object tracking performance achieved by discriminative correlation filters[14,15] (DCFs). In particular, Ma et al.[15] note that while the deeper CNN convolutional layers encode semantic information and have inbuilt invariance to object appearance changes, it is the early CNN convolutional layers that capture fine-scaled spatial patterns, which are useful for object localization. Moreover, Danelljan *et al.*[14] find that, unlike in image classification tasks,[16] the features extracted from the first CNN layer lead to better tracking performance compared to features from deeper layers.

Further author information:
Victor Stamatescu: E-mail: Victor.Stamatescu@unisa.edu.au, Telephone: +61(0)883023781

In this paper we systematically examine the use of learned first-layer convolutional filters as generic object detectors by using them to simultaneously track three types of objects across the same scene. The filters are taken from the first layer of the *OverFeat*[17] CNN or, alternatively, learned in an unsupervised manner using a Convolutional Restricted Boltzmann Machine[18] (CRBM), which typically serves as the first layer in a Convolutional Deep Belief Network[18] (CDBN). To the best of our knowledge, this paper presents the first direct comparison between the use of discriminative (CNN) and generative (CRBM) learned filters in visual tracking. Furthermore, we learn separate generative filter banks at four different scales and investigate their impact on the tracking performance. Rather than applying the commonly used image pyramid approach,[19] scale effects are explored by varying the CRBM filter size.

The remainder of this paper is organized as follows. Section 2 summarizes recent advances in multi-object visual tracking and in feature learning for object detection. In Section 3 we introduce our multi-object visual tracking system and describe the discriminative and generative convolutional filter banks used to perform object detection. We demonstrate the effect of these filters on multi-object tracking performance across the three kinds of objects in Section 5. Finally, Section 6 presents a summary of our results.

2. RELATED WORK

This section provides a overview of multi-object visual tracking, followed by an outline of recent developments in the application of *deep learning* to visual tracking. Particular focus is given to new results in the area of *transfer learning*,[20] which refers to the use of learned features in different vision tasks for which the original models were not specifically trained.

2.1 Multi-object visual tracking

In this paper we are interested in the online tracking of multiple types of objects in natural scenes, where the term *online* indicates a reliance on information acquired only up to the current frame. By contrast, most of the recent state-of-the-art online multi-object trackers focus on pedestrian tracking (e.g. see[21–25]). These systems typically employ a *tracking-by-detection* approach, where objects are detected independently in each new frame and uniquely associated with system tracks. A persistent problem in this approach is that noisy or missed detections can lead to incomplete system tracks. One way in which to improve the tracking performance is to design better object detectors, another being better methods for data association. An alternative approach, which is used in this work, is *track-before-detect*.[6] Under this paradigm, the top-down guidance provided by the tracking process allows weak detections to be correlated over multiple frames.

2.2 Visual Tracking using Deep Learning

Motivated by the success of ImageNet[26]-trained deep (many-layered) CNNs (e.g. AlexNet,[27] *OverFeat*,[17] VGG-Net[28]) on image classification benchmarks, a number of authors have successfully applied features learned by these models to auxilliary tasks such as scene and fine-grained recognition,[12] texture recognition and scene segmentation,[16] and object localisation.[13] In particular, the work on CNN-based object detection of Girshick *et al.*[13] shows that an effective solution when task-specific training data is scarce is to pre-train the CNN for image classification, where sufficient labeled data exists, and then fine-tune the network for object localisation.

The direct application of deep learning in visual tracking has recently gained popularity (e.g. see[29–32]), and has set the current benchmark in performance through MDNet,[33] the winner of VOT2015,[3] which uses a CNN trained disciminatively on a large set of annotated videos. Earlier examples developed ways of transfering prior information learned offline to online object tracking,[29] and trained deep, general-purpose neural networks.[30] Nwang *et al.*[31] learned generic features offline using a denoising autoencoder, and used these in a classification network that was tuned online. Wang *et al.*[34] learned hierarchical features offline in a shallow CNN that was subsequently adapted online. Nwang *et al.*[32] pre-trained a CNN to recognize what is an object and generate a probability map, while the CNN was adapted online. Recently Hu *et al.* presented the first CDBN-based visual tracker.[35]

Our paper is motivated by the approach of two recent DCF single object trackers[14,15] that use convolutional filters learned offline in deep CNNs. These filters are used as generic object detectors, without online fine tuning, to provide better (more discriminant) image features to existing visual tracking systems.

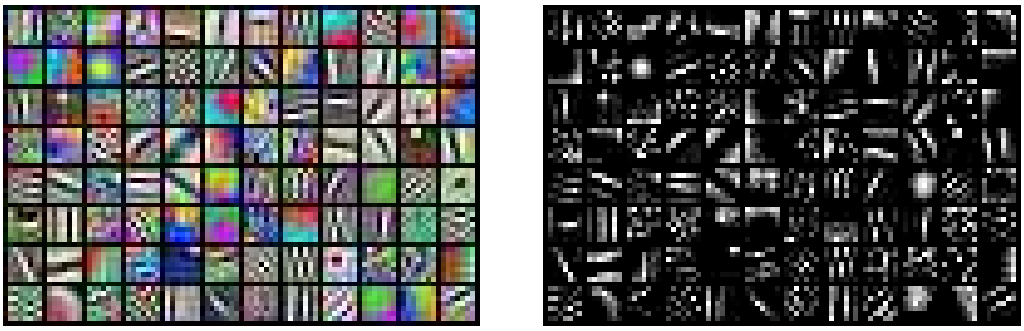

Figure 1: 96 first-layer filters from the *OverFeat* CNN.[17] The supervised training of the *OverFeat* CNN was carried out using *ImageNet* data.[26] These discriminative learned filters of size 7×7 pixels are shown in their original 3 channel (RGB) form (*left*) and after conversion to greyscale (*right*).

3. MULTI-OBJECT TRACKING USING LEARNED CONVOLUTIONAL FILTERS

The following section describes the filter banks used to detect multiple types of objects. It also provides key details regarding the multiple object tracking system that is used to evaluate the performance of the filters.

3.1 Learned first-layer convolutional filters

We are interested in assessing the ability of learned discriminative and generative convolutional filters to serve as generic object detectors. In the discriminative case, 96 convolutional filters were extracted from the first layer of *OverFeat*,[17] which was learned during the supervised training (using ground truth labels) of this network on the large-scale *ImageNet*[26] data set. These 7×7 pixel RGB filters are shown in Figure 1 together with their corresponding set of greyscale filters, which are also used in our comparative study. In the generative case, four separate banks of 24 convolutional greyscale filters, which are displayed in Figure 2, were learned using the publicly available CRBM code of Lee *et al.*[18]

3.1.1 Learning Generative Convolutional Filters

A CRBM is an extension of Restricted Boltzmann Machine (RBM), which is bipartite graphical model that encodes the statistical dependencies between the input image pixels. An RBM consists of visible unit (image pixels) and hidden unit layers with symmetric connections (or weights) between them. In a CRBM, these weights take the form of convolutional filters, which are shared across the image.[18] This provides invariance to translations of the input, so that a learned spatial feature may be detected at any location. The hidden layer is made up of K groups (here $\mathrm{K} = 24$ is used), each of which has an associated $\mathrm{w} \times \mathrm{w}$ pixel filter. The CRBM code trains a generative model that is a sparse, overcomplete[36] representation of the input and, in the process, learns convolutional filters that are typically object edge detectors.

We trained each CRBM model in an unsupervised manner on the first frames of *Neovision2*[37] Tower data set image sequences $010 - 024$. The original 15 images were first downsampled by a factor of two, so that their dimensions (960×540 pixels) match those of images in the sequence used to evaluate multi-object tracking performance. Using the CRBM code of Lee *et al.*, these images were pre-processed by converting to greyscale, applying the whitening function employed by Olshausen & Field,[36] subtracting the image mean and normalizing the result by the root mean square (*rms*) of the image. The only difference to the code of Lee *et al.* is that here the input images were not resized to have square dimensions. A sample RGB image and corresponding pre-processed input image are shown in Figure 3.

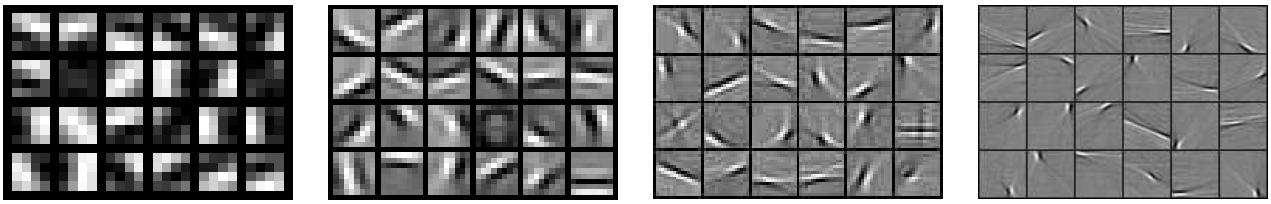

Figure 2: Four banks of 24 generative filters learned using a Convolutional Restricted Boltzmann Machine (CRBM).[18] The unsupervised training of the four separate CRBMs was carried out using the first frames of *Neovision2*[37] Tower sequences $010 - 024$. These images were downsampled by a factor of two, resulting in 15 RGB images of size 960×540 pixels, which were then converted to greyscale and pre-processed (see main text for details). In each filter bank the learned filters are of size: 4×4 pixels (*far-left*), 8×8 pixels (*middle-left*), 16×16 pixels (*middle-right*), and 32×32 pixels (*far-right*).

Figure 3: The first frame from the *Neovision2*[37] Tower sequence 011, which has been downsampled by a factor of two, where *left* is the resulting 960×540 pixel RGB image, *middle* is the greyscale and pre-processed (see main text for details) version of the same image, and *right* is the image after additional factor of two downsampling (i.e. dimensions of 480×270 pixels) followed by the same pre-processing.

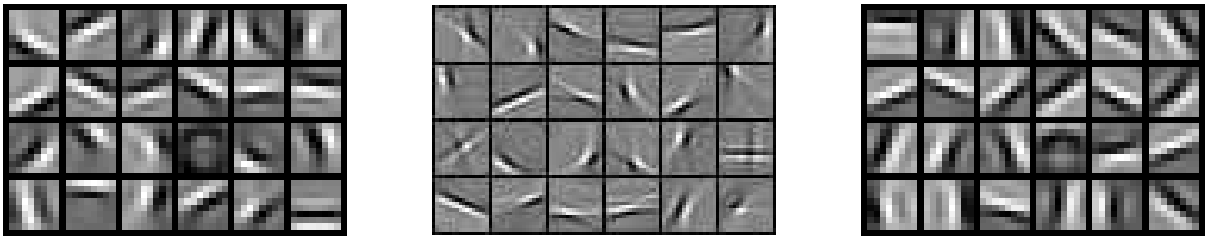

Figure 4: Banks of 24 filters trained using the Convolutional Restricted Boltzmann Machine (CRBM) unsupervised learning algorithm[18] on the first frames of 15 *Neovision2*[37] Tower sequences ($010 - 024$). The *left* (8×8 pixel) and *middle* (16×16 pixel) filters were trained on images with dimensions of 960×540 pixels, whereas the (8×8 pixel) filters on the *right* were trained on further downsampled correspoding images with dimensions of 480×270 pixels.

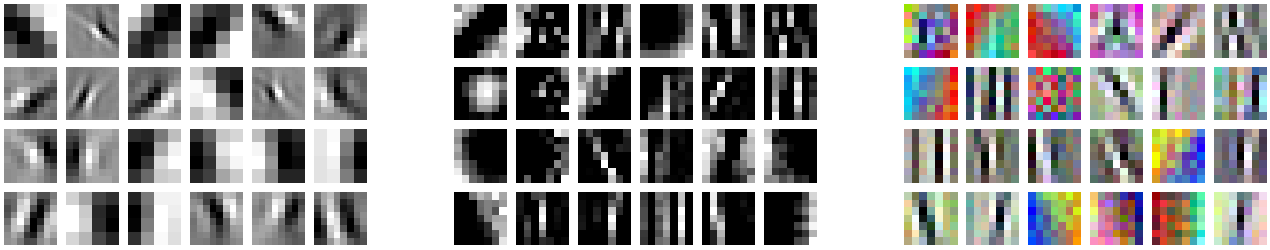

Figure 5: *Left*: 24 pre-selected Convolutional Restricted Boltzmann Machine (CRBM) learned filters: 10 filters of size 4×4 pixels, 11 filters of size 8×8 pixels, and 3 filters of size 16×16 pixels. *Middle*: 24 pre-selected grayscale-converted *OverFeat* first-layer filters. *Right*: 24 pre-selected RGB *OverFeat* first-layer filters. All *OverFeat* filters are of size 7×7 pixels. The pre-selection method is described in the main text.

3.1.2 Multi-scale Generative Convolutional Filters

Objects in a scene (e.g. cars, people, cyclists, trees, benches, *etc.*) can have a variety of sizes and exhibit spatial patterns (e.g. oriented edges) at different scales. To obtain strong detections for different types of objects, a convolutional filter bank should include filters able to activate on such patterns. To this end, Serre *et al.*[38] used a set of 64 fixed, multi-scale Gabor filters in the first layer of their biologically-inspired HMAX object-recognition network. These Gabor filters spanned 16 scales (filter sizes of 7×7 to 37×37 pixels in steps of two pixels) and four orientations (0°, 45°, 90°, 135°). Motivated by their approach, we investigate the use of filters learned at different scales (see Figure 2) by training four separate CRBM models on the same set of 15 images using 24 convolutional filters with four different sizes (w × w): 4×4 pixels, 8×8 pixels, 16×16 pixels and 32×32 pixels.

We note that our approach differs from the technique of image pyramids[19] that is typically applied to detect image patterns at different scales. Using image pyramids, a fixed-size target pattern is convolved with multiple copies of the same image at reduced resolutions, and this is equivalent (but computationally more efficient) to convolving the image with copies of the target pattern at increasing scales. Figure 4 illustrates that a similar equivalence does not hold when *learning* the target pattern: features learned using 16×16 pixel filters on 960×540 pixel training images (such as the *middle* image in Figure 3) look very different to those learned using 8×8 pixel filters on training images at $2\times$ reduced resolution (such as the image on the *right* in Figure 3). The reason for this is that the smaller filter size constrains the learned patterns in such a way that they appear more similar to 8×8 pixel filters learned on the full resolution (960×540 pixel) training images.

3.1.3 Pre-selected Discriminative and Generative Convolutional Filters

In order to compare the multi-object tracking performance obtained with discriminative first-layer *OverFeat* filters or generative CRBM-learned filters, we pre-select 24 filters from each of the 96 RGB filters in Figure 1, the 96 greyscale filters in Figure 1, and the 96 filters in Figure 2. The filters selected are those that can best detect the ground truth annotated objects against their local background across the 15 *Neovision2* Tower data set images used to train the CRBMs. Before the greyscale filters are used to compute feature maps, the images are first pre-processed by converting to greyscale, whitening, mean subtraction and normalization by *rms*, as described previously. In the case of the RGB *OverFeat* filters, these are applied to the raw images by convolving each filter channel with its corresponding image color channel and then summing the three resulting feature maps pixel-wise. For each candidate filter, *Symmetrical Uncertainty* (SU),[39] which is normalized *Mutual Information*, is used to measure the separation between the feature response extracted from each ground truth annotation region (an oriented rectangular box) and the feature response extracted from its surrounding local background region. The SU for each filter is accumulated for every annotated object in the 15 frames, and the candidate filters are ranked according to their combined SU.

Using this method, the 24 top-ranked generative and discriminative (greyscale and RGB) filters are shown in Figure 5. While the discriminative first-layer filters from *OverFeat* all have the same size (7×7 pixels), the generative filters are chosen among four filter sizes, three of which are selected with the following abundances: 10 filters of size 4×4 pixels, 11 filters of size 8×8 pixels, and 3 filters of size 16×16 pixels.

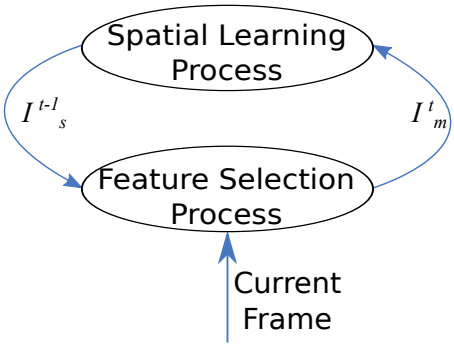

Figure 6: The feedback mechanism between online feature selection and spatial learning allows *CACTuS-FL* to autonomously focus attention to regions of locally correlated saliency.[42] I_S^{t-1} is the learned image of the object that is output from the tracking stage of the algorithm by the Shape Estimating Filter (SEF). The measured image, I_m^t, is the fused likelihood ratio map produced in the object detection stage of the algorithm.

3.2 Multi-object Tracking

This section describes the multi-object tracking system used to evaluate the impact of the discriminative and generative learned filter banks. Our system is the Competitive Attentional Correlation Tracker using Shape and Feature Learning (*CACTuS-FL*).[1] Multi-object tracking algorithms provide estimates of the object trajectories by uniquely associating new detections with each existing trajectory. The system tracks in *CACTuS-FL* can self-initialize through a combination of track-before-detect and *adaptive tracking* (e.g. see[40,41]). Specifically, an adaptive hierarchical state model *learns* the object shape and motion,[5] and provides top-down guidance to future detections, while the detection stage provides bottom-up input to the model, as illustrated by Figure 6. This feedback mechanism enables a tracker to autonomously move towards regions of temporally consistent local saliency.[42] The unique identities (IDs) of multiple objects are preserved by operating multiple subtrackers in competition with each other across the scene.

3.2.1 Object Detection

The autonomous detection of individual objects follows the method of Collins *et al.*,[4] which frames the online selection of a feature subset (from a larger set given by a filter bank) as an online two-class (object/local background) classification problem. Each candidate convolutional filter n is used to compute a *feature map* $Z_n^t(i)$ of the image t based on its feature response at each pixel i.

The detection process extracts the class-conditioned feature response distributions from the object and local background regions. In the system of Collins *et al.*,[4] the position of these regions is given by the object tracking process, while their extent is pre-defined by the user using concentric rectangular bounding boxes. Our system instead uses the learned object image from the previous frame I_s^{t-1} and applies it as a pixel mask to extract the class-conditioned feature response distribution:

$$F^t(u) = \frac{\sum_i I_s^{t-1}(i)\,\delta\left(Z_n^t(i) - u\right)}{\Sigma_u}, \tag{1}$$

where u denotes the bin size for a range of feature response values, i is the pixel index, δ is the Dirac delta function and $\frac{1}{\Sigma_u}$ is a normalization constant. B^t is extracted in the same way using a pixel mask of $1 - I_s^{t-1}$ over a local image patch. The learned image I_s^{t-1} mask allows for a more precise extraction of the feature response distributions because it limits pollution from the surroundings.[42] This, in turn, leads to stronger detections, which serve as input to the spatial learning process, as illustrated in Figure 6.

A detection map is computed for each feature using the Likelihood Ratio $LR = F^t(u)/B^t(u)$ by propagating it back into its corresponding feature map: $LR\left(Z_n^t(i)\right)$, and normalizing. Online feature selection then consists of choosing a subset of N detection maps (chosen empirically to be 6) to be used for tracking. To select discriminant

features, we apply the principle of Maximum Marginal Diversity[43] (MMD), which holds for features obtained from bandpass filters and has been successfully applied to visual tracking.[44,45] This involves scoring each feature n according to the *Mutual Information* $MI(X_n^t; C)$ between the random variable X_n^t, which represents the feature response in both object and local background regions, and the random variable C, which represents the class label:

$$MI(X_n^t; C) = \sum_{i=0}^{1} \sum_{u \in X_n^t} p(X_n^t = u; C = i) log_2 \frac{p(X_n^t = u; C = i)}{p(X_n^t = u)p(C = i)} \ , \tag{2}$$

where $C = 0$ and $C = 1$ denote the local background and object class labels, respectively.

The LR maps with the top 6 scores are then summed pixel-wise (i) in a weighted average to yield a fused detection map that serves as input measurement I_m to the tracking algorithm:

$$I_m^t(i) = \sum_{n=1}^{N} w_n P\left(o \mid Z_n^t(i)\right) \ , \tag{3}$$

where $P\left(o \mid Z_n^t(i)\right) \propto LR\left(Z_n^t(i)\right)$ is a measured detection map and $w_n = MI(X_n^t; C) \times B$ is its weight. Here B is the Bhattacharyya coefficient:[46]

$$B = \sum_u \sqrt{F_m^t(u) F_s^{t-1}(u)} \ , \tag{4}$$

which rewards temporal consistency between the target region feature response F_m^t measured in the current frame and a target region feature response learned up to the previous frame F_s^{t-1}. The learned target region feature response distribution is updated at each frame:

$$F_s^t(u) = \frac{F_s^{t-1}(u) F_m^t(u)}{\Sigma_u} \ , \tag{5}$$

where $\frac{1}{\Sigma_u}$ is a normalization constant, so that F_s^t is the posterior learned target region distribution for the current frame.

The notation of the detection map in Eq. 3 represents the probability of o, the hypothesis that pixel i belongs to an object, given the feature map $Z_n^t(i)$.

3.2.2 Object Trajectory Estimation

Each subtracker, which is called a Shape Estimating Filter (SEF),[5] learns an object state model that includes a probabilistic representation of its shape. A SEF autonomously combines information from past frames with new measurements to recursively to estimate the position, velocity and shape of the object.

In order to automatically associate new measurements to multiple system tracks, *CACTuS-FL* operates multiple SEFs simultaneously using a competitive attentional framework designed to enforce the tracking of multiple objects.[1] Under this scheme, the SEFs track everything in the scene, including parts of the background or sources of clutter, so that new measurements are assigned to the system tracks that best describe those measurements.

4. EXPERIMENTAL EVALUATION

We investigate the application of the learned convolutional filter banks described in Section 3 to object detection by comparing the corresponding multi-object tracking performance of *CACTuS-FL* on the *Neovision2* Tower sequence 001. To reduce the computational burden, the data were downsampled by a factor of two, so that the images used for tracking have dimensions of 960×540 pixels. Furthermore, we have added unique object IDs by hand to the original ground truth data, which consists of oriented object bounding boxes and associated class labels (car, person or cyclist). The Tower sequence 001 is 871 frames long and contains 14 ground truths: a (stationary) car, six people (two stationary, four moving) and seven cyclists (all moving). Not all ground truth objects exist for the full duration of the sequence, with some exiting the scene, while others enter it.

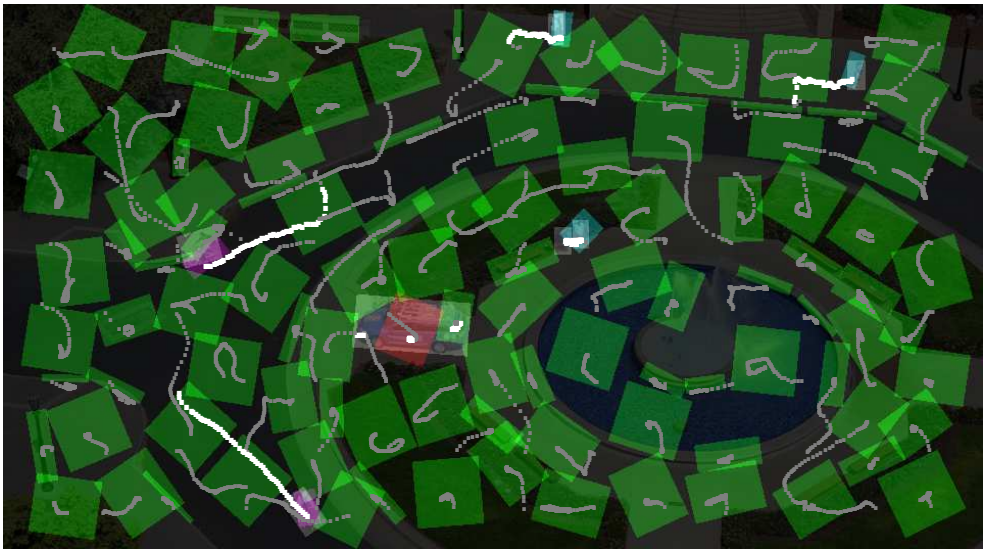

Figure 7: Evaluation of multi-object tracking up to frame 61 of *Neovision2*[37] Tower image sequence 001, showing *CACTuS-FL* system tracks (grey dots) and CLEAR MOT[48] metrics matches (white dots). The ground truth bounding boxes are displayed as grey shaded rectangles centred on each car, person or cyclist. The bounding boxes estimated by *CACTuS-FL* in the current frame, which are computed by parameterizing the object shape learned by each SEF, are colored green in the case of unmatched SEFs, whereas for those SEFs matched to ground truth objects they are colored according to a corresponding ground truth class label: red for cars, magenta for cyclists, cyan for people.

CACTuS-FL was evaluated on the Tower sequence 001 in order to test the seven banks of 24 convolutional filters in its detection stage. The banks learned by CRBMs are those using individual filter sizes (4×4 pixels, 8×8 pixels, 16×16 pixels, and 32×32 pixels) shown in Figure 2, and the so-called multi-scale bank of pre-selected filters shown in Figure 5. These filters are compared with the two banks of pre-selcted greyscale and RGB first-layer filters from *OverFeat* that are also shown in Figure 5. When greyscale filters are used to compute feature maps, the input images undergo the same image pre-processing used in training the CRBMs (greyscale conversion, whitening, mean subtraction and normalization by *rms*). When RGB (*OverFeat*) filters are used instead, each filter channel is convolved with its corresponding raw image color channel and the three output maps are summed pixel-wise into a single feature map.

The system is initialized in the first frame by positioning 112 SEFs at regular intervals in a rectangular (14×8 SEF) grid across the scene. At every subsequent frame, each SEF provides as output a learned image of the object that it is tracking. In order to compute the degree of spatial overlap (ranging from 0 to 1) between a learned image and the ground truth bounding boxes, learned images are parameterized using an ellipse of second order moments.[47] The 2σ length and width along the ellipse major and minor axis, respectively, are used to define an oriented bounding box, multiple instances of which are illustrated in Figure 7.

4.1 Multi-object Tracking Performance Metrics

We evaluate the multi-object tracking performance with the widely-used CLEAR MOT metrics[48] by following the implementation of Andriyenko *et al.*[49] This evaluation protocol uses the Munkres algoirthm[50] to fnd the optimum assignment between system tracks and ground truths in terms of their total spatial overlap. The uniquely associated bounding box pairs of system tracks and ground truths are labeled as *matches* when their degree of spatial overlap exceeds a user-defined threshold. Given that we postpone any decision on what constitutes a *false alarm* track, the metrics used in our comparative study are *Recall* and MOTP. *Recall* refers to the % of detected targets (or *matched* ground truths), while MOTP measures the target tracking precision by averaging the system track and ground truth bounding box overlap across all matches.

4.2 Multi-object Tracking Performance using Learned Filters

Table 1: *Recall* (% of detected targets) and MOTP (multi-object tracking precision) evaluated at a spatial overlap threshold of 0.2 for *CACTuS-FL* coupled with the listed banks of 24 learned convolutional filters. The metrics were computed using all ground truth objects in *Neovision2*[37] Tower sequence 001.

Metric	CRBM 4×4 pixels	CRBM 8×8 pixels	CRBM 16×16 pixels	CRBM 32×32 pixels	CRBM multi-scale	*OverFeat* greyscale	*OverFeat* RGB
Recall	36.31%	55.55%	**60.37**%	54.32%	55.12%	55.49%	53.04%
MOTP	38.84%	39.93%	**43.44**%	35.30%	37.99%	42.16%	36.03%

Table 2: The Area Under the Curve (AUC) for *Recall* (% of detected targets) and MOTP (multi-object tracking precision) computed between spatial overlap thresholds of $0.2-0.5$ for *CACTuS-FL* coupled with the listed banks of 24 learned convolutional filters. The metrics were computed using all ground truth objects in *Neovision2*[37] Tower sequence 001.

AUC	CRBM 4×4 pixels	CRBM 8×8 pixels	CRBM 16×16 pixels	CRBM 32×32 pixels	CRBM multi-scale	*OverFeat* greyscale	*OverFeat* RGB
Recall	220.48	290.81	**331.03**	265.88	285.24	306.01	295.96
MOTP	239.14	253.82	**255.91**	212.39	240.13	247.34	249.42

The learned convolutional filter banks are compared in terms of *Recall* and MOTP for spatial overlap thresholds of 0 to 0.5, in steps of 0.1. Figure 8 shows the *Recall* and MOTP for all ground truth objects, while Figures 9, 10 and 11 show the *Recall* and MOTP metrics computed using only cars, people or cyclists, respectively. We note that large spatial overlap thresholds tend to not only penalize a loss of track, but also those SEFs that only learn a small part of the object (e.g. the arm of a cyclist, a car part, etc.), which leads to a small bounding box and a lower *Recall*.

Considering the overall performance across all ground truth object types, Table 1 compares the metric results at a (typically used) spatial overlap threshold of 0.2. Alternatively, to evaluate the performance across all spatial overlap thresholds between $0 - 0.5$, Table 2 compares the Areas Under the Curves (AUCs) of Figure 8. Both Tables 1 and 2 indicate that the 16×16 pixel CRBM filters achieve the top performance in terms of both *Recall* and MOTP, whereas the 4×4 pixel CRBM filters yields the lowest *Recall* in both tables. The lowest MOTP in both tables is given by the the 32×32 pixel CRBM filters.

We next compare the performance of the learned filter banks for each object class in terms of the *Recall* and MOTP AUCs in Tables 3 and 4, respectively. The best performance for the car, both in terms of *Recall* and MOTP, is given by the *Overfeat* RGB filters. The 8×8 pixel CRBM filters provide the worst car *Recall* and MOTP. The people are best tracked, in terms of both *Recall* and MOTP, when using the 16×16 pixel CRBM filters, while the worst performance is obtained with 32×32 pixel CRBM filters. Finally, for cyclists, 32×32 pixel CRBM filters provide the best *Recall* and the worst MOTP. The best MOTP in this case is given by the *OverFeat* RGB filters, while the worst *Recall* comes from 4×4 pixel CRBM filters.

We note that while the second-largest (16×16 pixel) CRBM filters offer the best overall performance on this data set, the smallest (4×4 pixel) CRBM filters are consistently the worst performing in terms of *Recall*. This suggests that some spatial features useful in distinguishing the car, people or cyclists from the local background may be too large for the learned 4×4 pixel filters to activate on. It is also interesting that the multi-scale CRBM filters have neither the worst nor best overall performance, but that they remain competitive with the *OverFeat* greyscale and RGB filters, which were pre-selected using the same method. That the multi-scale CRBM filters perform wose than the 16×16 pixel CRBM filters may suggest that the method used to pre-select them is suboptimal. However, it may also be the case that the best scale at which to learn features for objects in this scene, including objects classed as clutter, is close to that provided by the 16×16 pixel CRBM filters. Under the competitive attentional framework of *CACTuS-FL*, this would enable some SEFs to better *explain-away* distracting background objects, which, in turn, would reduce number of lost tracks for all objects in the scene.

Table 3: The Area Under the Curve (AUC) for *Recall* (% of detected targets) computed between spatial overlap thresholds of $0.2 - 0.5$ for *CACTuS-FL* coupled with the listed banks of 24 learned convolutional filters. Here *Recall* was computed separately for each class (car, person, cyclist) of ground truth objects in *Neovision2*[37] Tower sequence 001.

Class	CRBM 4×4 pixels	CRBM 8×8 pixels	CRBM 16×16 pixels	CRBM 32×32 pixels	CRBM multi-scale	*OverFeat* greyscale	*OverFeat* RGB
Car	366.02	306.20	444.78	494.15	357.98	466.71	**513.32**
Person	167.49	276.40	**292.00**	166.70	264.65	262.61	270.30
Cyclist	265.95	323.63	354.44	**357.82**	295.53	327.89	247.94

Table 4: The Area Under the Curve (AUC) for MOTP (multi-object tracking precision) computed between spatial overlap thresholds of $0.2 - 0.5$ for *CACTuS-FL* coupled with the listed banks of 24 learned convolutional filters. Here MOTP was computed separately for each class (car, person, cyclist) of ground truth objects in *Neovision2*[37] Tower sequence 001.

Class	CRBM 4×4 pixels	CRBM 8×8 pixels	CRBM 16×16 pixels	CRBM 32×32 pixels	CRBM multi-scale	*OverFeat* greyscale	*OverFeat* RGB
Car	213.85	96.90	204.53	218.84	204.39	227.82	**236.42**
Person	261.56	260.68	**269.89**	91.79	248.46	251.39	246.27
Cyclist	245.74	260.07	272.77	231.90	252.62	263.23	**275.92**

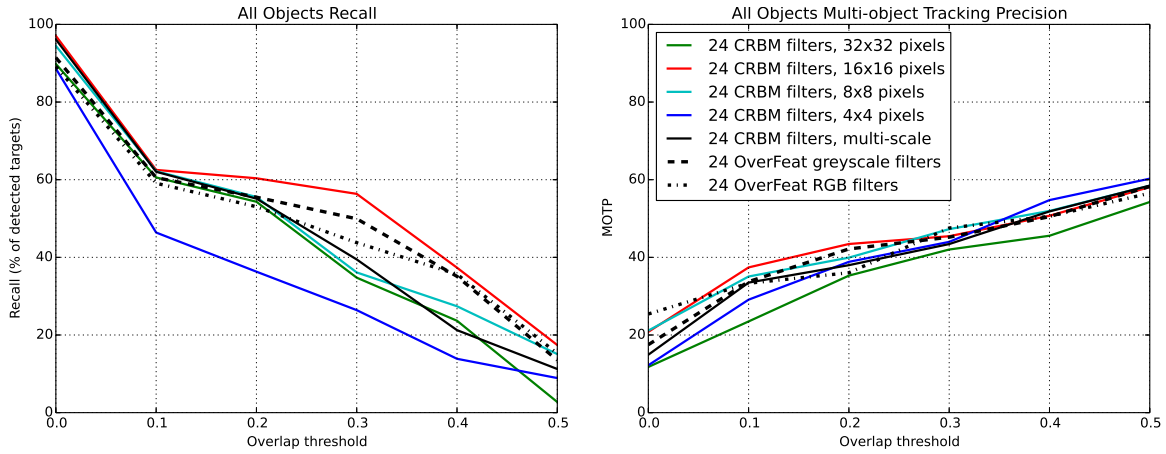

Figure 8: *Recall* and MOTP as functions of the spatial overlap threshold for all ground truth objects in *Neovision2* Tower sequence 001.

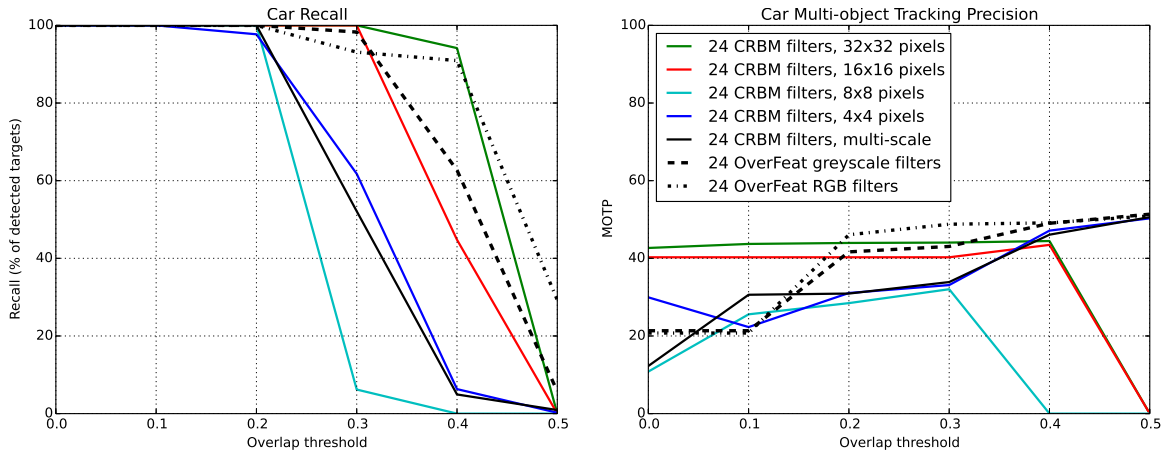

Figure 9: *Recall* and MOTP as functions of the spatial overlap threshold for the car in *Neovision2* Tower sequence 001.

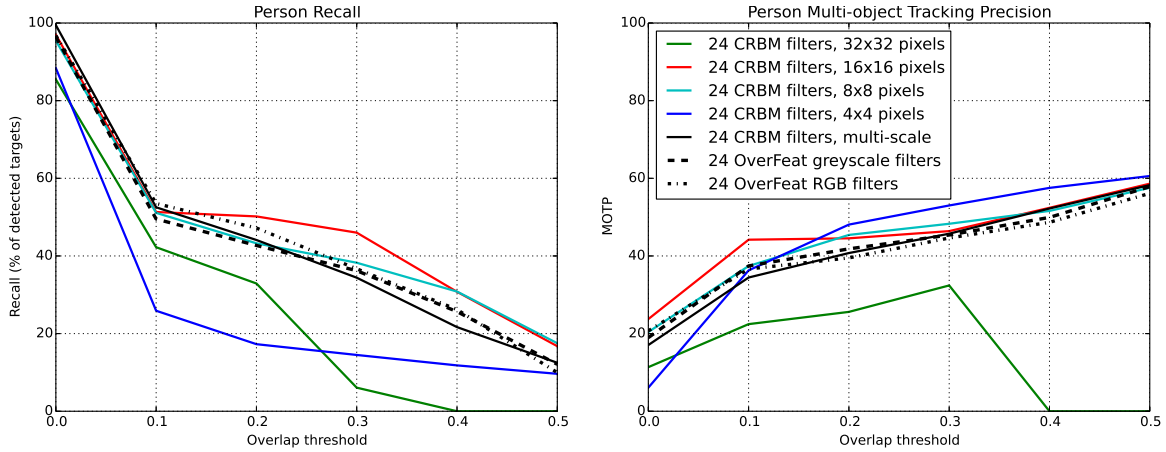

Figure 10: *Recall* and MOTP as functions of the spatial overlap threshold for all people in *Neovision2* Tower sequence 001.

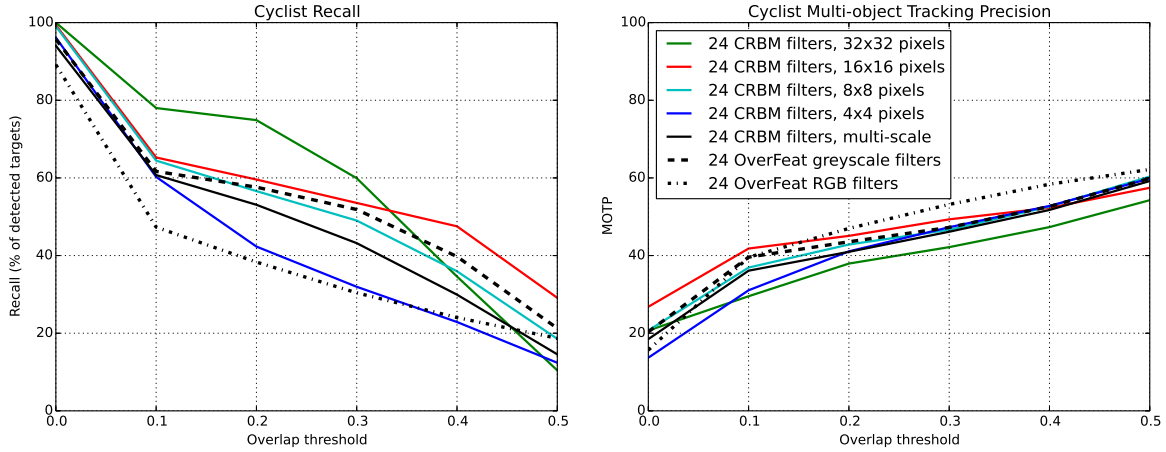

Figure 11: *Recall* and MOTP as functions of the spatial overlap threshold for all cyclists in *Neovision2* Tower sequence 001.

5. CONCLUSION

In this paper we have investigated the use of learned convolutional filters as generic object detectors for multi-object visual tracking. This work is motivated by recent experimental evidence, which suggests that filters learned by CNNs in their first layer are better suited to object detection than those in their deeper convolutional layers. We have compared RGB and greyscale discriminative filters from the first layer of the *OverFeat* CNN with greyscale generative filters learned at four separate spatial scales using a CRBM. We find that the largest impact on the multi-object tracking performance is due to the choice of filter size, rather than the manner in which the filters were learned or whether they are greyscale or RGB. Our study suggests that choosing a sufficiently large filter size is important if the filter bank is to have the ability to learn discriminant image features across a range of scales.

ACKNOWLEDGMENTS

The authors would like to thank Dr Adam Gatt for suggestions that helped us to improve this paper.

REFERENCES

[1] Wong, S., Gatt, A., Kearney, D., Milton, A., and Stamatescu, V., "A competitive attentional approach to mitigating model drift in adaptive visual tracking," in [*The 29th International Conference on Image and Vision Computing New Zealand (IVCNZ'14)*], 1–6, ACM (2014).

[2] Kristan, M. et al., "The Visual Object Tracking VOT2014 challenge results," in [*Proceedings, European Conference on Computer Vision (ECCV) Visual Object Tracking Challenge Workshop*], (2014).

[3] Kristan, M. et al., "The Visual Object Tracking VOT2015 challenge results," in [*The IEEE International Conference on Computer Vision (ICCV) Workshops*], (2015).

[4] Collins, R., Liu, Y., and Leordeanu, M., "Online selection of discriminative tracking features," *IEEE Trans. on Pattern Analysis and Machine Intelligence* **27**(10), 1631–1643 (2005).

[5] Wong, S. and Kearney, D., "Relating image, shape, position, and velocity in visual tracking," in [*Proc. SPIE 7338, Acquisition, Tracking, Pointing, and Laser Systems Technologies XXIII*], SPIE (2009).

[6] Baker, T. and Strens, M., "Representation of uncertainty in spatial target tracking," in [*1998. Proceedings. Fourteenth International Conference on Pattern Recognition*], **2**, 1339–1342, IEEE (1998).

[7] Itti, L., Koch, C., and E., N., "A model of saliency-based visual attention for rapid scene analysis," *IEEE Trans. on Pattern Analysis and Machine Intelligence* **20**, 1254–1259 (1998).

[8] Dalal, N. and Triggs, B., "Histograms of oriented gradients for human detection," in [*Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*], 886–893 (2005).

[9] Yin, Z. and Collins, R., "Moving Object Localization in Thermal Imagery by Forward-backward MHI," in [*Computer Vision and Pattern Recognition Workshop, 2006*], 133–133 (2006).

[10] Hou, X. and Zhang, L., "Saliency Detection: A Spectral Residual Approach," in [*Conference on Computer Vision and Pattern Recognition (CVPR'07), IEEE Computer Society*], 1–8, IEEE (2007).

[11] LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P., "Gradient-based learning applied to document recognition," *Proceedings of the IEEE* **86**(11), 2278–2324 (1998).

[12] Razavian, A., Azizpour, H., Sullivan, J., and Carlsson, S., "CNN features off-the-shelf: An astounding baseline for recognition," in [*IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*], 512–519 (2014).

[13] Girshick, R., Donahue, J., Darrell, T., and Malik, J., "Rich feature hierarchies for accurate object detection and semantic segmentation," in [*IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*], 580–587 (2014).

[14] Danelljan, M., Häger, G., Khan, F., and Felsberg, M., "Convolutional features for correlation filter based visual tracking," in [*The IEEE International Conference on Computer Vision (ICCV) Workshops*], 621–629 (2015).

[15] Ma, C., Huang, J.-B., Yang, X., and Yang, M.-H., "Hierarchical convolutional features for visual tracking," in [*Proceedings of the IEEE International Conference on Computer Vision*], 3074–3082 (2015).

[16] Cimpoi, M., Maji, S., and Vedaldi, A., "Deep filter banks for texture recognition and segmentation," in [*IEEE Conference on Computer Vision and Pattern Recognition*], (2015).

[17] Sermanet, P., Eigen, D., Zhang, X., Mathieu, M., Fergus, R., and LeCun, Y., "Overfeat: Integrated recognition, localization and detection using convolutional networks," in [*International Conference on Learning Representations (ICLR 2014)*], (2014).

[18] Lee, H., Grosse, R., Ranganath, R., and Ng, A., "Convolutional deep belief networksfor scalable unsupervised learning of hierarchical representations," in [*Proceedings of the 26th Annual International Conference on Machine Learning*], *ICML '09*, 609–616 (2009).

[19] Adelson, E. H., Anderson, C. H., Bergen, J. R., Burt, P. J., and Ogden, J. M., "1984, Pyramid methods in image processing," *RCA Engineer* **29**(6), 33–41 (1984).

[20] Pan, S. J. and Yang, Q., "A survey on transfer learning," *IEEE Transactions on Knowledge and Data Engineering* **22**(10), 1345–1359 (2010).

[21] Yang, J., Vela, P. A., Shi, Z., and Teizer, J., "Probabilistic multiple people tracking through complex situations," in [*In Performance Evaluation of Tracking and Surveillance (PETS) workshop at CVPR 2009*], 79–86 (2009).

[22] Breitenstein, M. D., Reichlin, F., Leibe, B., Koller-Meier, E., and Van Gool, L., "Online multiperson tracking-by-detection from a single, uncalibrated camera," *IEEE Trans. on Pattern Analysis and Machine Intelligence* **33**(9), 1820–1833 (2011).

[23] Wu, Z., Zhang, J., and Betke, M., "Online motion agreement tracking," in [*Proceeding of the 24th British Machine Vision Conference (BMVC)*], (2013).

[24] Bae, S. H. and Yoon, K., "Robust online multi-object tracking based on tracklet confidence and online discriminative appearance learning," in [*IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*], 1218–1225 (2014).

[25] Possegger, H., Mauthner, T., Roth, P., and Bischof, H., "Occlusion geodesics for online multi-object tracking," in [*IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*], 1306–1313 (2014).

[26] Deng, J., Dong, W., Socher, R., Li, L., Li, K., and Li, F., "ImageNet: A large-scale hierarchical image database," in [*Proceedings IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*], (2009).

[27] Krizhevsky, A., Sutskever, I., and Hinton, G. E., "ImageNet classification with deep convolutional neural networks," in [*Advances in neural information processing systems*], 1097–1105 (2012).

[28] Simonyan, K. and Zisserman, A., "Very deep convolutional networks for large-scale image recognition," *CoRR* **abs/1409.1556** (2014).

[29] Wang, Q., Chen, F., Yang, J., Xu, W., and Yang, M.-H., "Transferring visual prior for online object tracking," *IEEE Transactions on Image Processing* **21**(7), 3296–3305 (2012).

[30] Jin, J., Dundar, A., Bates, J., Farabet, C., and Culurciello, E., "Tracking with deep neural networks," in [*47th Annual Conference on Information Sciences and Systems (CISS)*], 1–5 (2013).

[31] Wang, N. and Yeung, D.-Y., "Learning a deep compact image representation for visual tracking," in [*Proceedings of Twenty-Seventh Annual Conference on Neural Information Processing Systems (NIPS)*], (2013).

[32] Wang, N., Li, S., Gupta, A., and Yeung, D.-Y., "Tansferring rich feature hierarchies for robust visual tracking," *arXiv preprint arXiv:1501.04587* (2015).

[33] Nam, H. and Han, B., "Learning multi-domain convolutional neural networks for visual tracking," *CoRR* **abs/1510.07945** (2015).

[34] Wang, L., Liu, T., Wang, G., Chan, K., and Q., Y., "Video tracking using learned hierarchical features," *IEEE Transactions on Image Processing* **24**(4), 1424–1435 (2015).

[35] Hu, D., Zhou, X., and Wu, J., "Visual tracking based on convolutional deep belief network," in [*Advanced Parallel Processing Technologies - 11th International Symposium, APPT, Proceedings*], 103–115 (2015).

[36] Olshausen, B. A. and Field, D. J., "Sparse coding with an overcomplete basis set: A strategy employed by v1?," *Vision Research* **37**(23), 3311–3325 (1997).

[37] "DARPA Neovision2." http://ilab.usc.edu/neo2/dataset/ (2013). Accessed: 2014-06-04.

[38] Serre, T., Wolf, L., Bileschi, S., Riesenhuber, M., and Poggio, T., "Robust object recognition with cortex-like mechanisms," *IEEE Trans. on Pattern Analysis and Machine Intelligence* **29**(3), 411–426 (2007).

[39] Press, W., Flannery, B., Teukolsky, S., and Vetterling, W., [*Numerical recipes in C*], Cambridge University Press, Cambridge (1988).

[40] Comaniciu, D., Ramesh, V., and Meer, P., "Kernel-based Object Tracking," *IEEE Trans. on Pattern Analysis and Machine Intelligence* **25**(5), 564–577 (2003).

[41] Wong, S., "Advanced correlation tracking of objects in cluttered imagery," in [*SPIE Acquisition, Tracking, and Pointing XIX*], **5810**, 158–169, SPIE (2005).

[42] Gatt, A., Wong, S., and Kearney, D., "Combining online feature selection with adaptive shape estimation," in [*25th International Conference of Image and Vision Computing New Zealand (IVCNZ), 2010*], 1–8, IEEE (2010).

[43] Vasconcelos, N., "Feature selection by maximum marginal diversity: optimality and implications for visual recognition.," in [*Conference on Computer Vision and Pattern Recognition (CVPR'03), IEEE Computer Society*], 762–769, IEEE (2003).

[44] Mahadevan, V. and Vasconcelos, N., "Biologically inspired object tracking using center-surround saliency mechanisms," *IEEE Trans. on Pattern Analysis and Machine Intelligence* **35**(3), 541–554 (2013).

[45] Stamatescu, V., Wong, S., Kearney, D., Lee, I., and Milton, A., "Mutual information for enhanced feature selection in visual tracking," *Proc. SPIE, Automatic Target Recognition XXV* **9476**, 947603–947603–11 (2015).

[46] Bhattacharyya, A., "On a measure of divergence between two statistical populations defined by their probability distributions," *Bulletin of the Calcutta Mathematical Society* **35**, 99–109 (1943).

[47] M. Hillas, A., "Cerenkov light images of EAS produced by primary gamma rays and by nuclei," in [*Proceedings of the 19th International Cosmic Ray Conference*], **3**, 445–448 (1985).

[48] Bernardin, K. and Stiefelhagen, R., "Evaluating multiple object tracking performance: The clear mot metrics," *EURASIP Journal on Image and Video Processing* **2008**, 1–10 (2008).

[49] Andriyenko, A., Schindler, K., and Roth, S., "Discrete-continuous optimization for multi-target tracking," in [*IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2012)*], 1926–1933 (2012).

[50] Munkres, J., "Algorithms for the assignment and transportation problems," *Journal of the Society of Industrial and Applied Mathematics* **5**(1), 32–38 (1957).