Track Everything: Limiting Prior Knowledge in Online Multi-Object Recognition

Sebastien C. Wong, *Senior Member, IEEE*, Victor Stamatescu, *Member, IEEE*, Adam Gatt, *Member, IEEE*, David Kearney, Ivan Lee, *Senior Member, IEEE*, and Mark D. McDonnell, *Senior Member, IEEE*

***Abstract*— This paper addresses the problem of online tracking and classification of multiple objects in an image sequence. Our proposed solution is to first track all objects in the scene without relying on object-specific prior knowledge, which in other systems can take the form of hand-crafted features or user-based track initialization. We then classify the tracked objects with a fast-learning image classifier, that is based on a shallow convolutional neural network architecture and demonstrate that object recognition improves when this is combined with object state information from the tracking algorithm. We argue that by transferring the use of prior knowledge from the detection and tracking stages to the classification stage, we can design a robust, general purpose object recognition system with the ability to detect and track a variety of object types. We describe our biologically inspired implementation, which adaptively learns the shape and motion of tracked objects, and apply it to the Neovision2 Tower benchmark data set, which contains multiple object types. An experimental evaluation demonstrates that our approach is competitive with the state-of-the-art video object recognition systems that do make use of object-specific prior knowledge in detection and tracking, while providing additional practical advantages by virtue of its generality.**

***Index Terms*— Object recognition, image classification, visual tracking, multi-object tracking.**

I. Introduction

We REPORT on the design of an automated vision system that can accurately locate and recognize multiple types of objects. The goal of online object recognition systems is to continuously detect and correctly classify the objects in a scene as they undergo changes in motion or appearance. Furthermore, the system should be robust to distracting or occluding clutter. Our proposed solution to these challenges is an adaptive multiple object tracking (MOT) algorithm that tracks all objects in the scene and defers any decisions on *what is an object of interest* to a separate classification stage. Object recognition then involves combining these class predictions, with state information given by object tracking. This approach emulates the separate *what* and *where* processing streams in primate vision [1], and allows the tracking process to be performed without any reliance on object-specific prior knowledge.

An important practical consideration in the design of online object recognition systems is the finite amount of labeled and annotated data available for training. When scarce, this can degrade classification performance due to overfitting and reduce the detection probability of highly tuned object detectors. Even when larger data sets are available, these may be biased in such a way that their image statistics do not accurately reflect the data encountered by the system at run time [2]. In the case of classifier-based object recognition [3] and detection [4], the use of *features*, which are higher-level representations of an object than the raw image, can mitigate these problems by providing a degree of invariance across different data sets. In the case of tracking and object detection algorithms, the same set of challenges can be addressed by making the tracker and detector designs less domain-specific. In our system this is achieved through the use of *adaptive tracking* (e.g., [5], [6]) and by employing a *track-before-detect* [7] approach that delays the requirement for object specific prior knowledge from detection until recognition.

We note that there exist commercial and security video analysis applications in which the user may not possess specific knowledge about new, previously unseen objects. For example, the user may not have access to information on the appearance of a set of target objects, but may still wish to track these targets in order to accumulate a domain-specific data set. Moreover, it may be impractical for the user to initialize the system on multiple targets, especially when more objects are expected to come into view, or are stationary for long periods. Therefore, in applications where the system requirements are initially not well defined, a useful first step is for the system to autonomously detect and track all (moving and stationary) objects, including those that may, at first, not be considered objects of interest.

Given these aims and real-world requirements, we present a novel approach to online object recognition centered on the idea of tracking all salient objects in the scene. We argue that this "track everything" approach can be realized by limiting the explicit use of prior knowledge, and demonstrate that this can be implemented by simultaneously learning both feature

Manuscript received September 27, 2016; revised February 19, 2017 and March 30, 2017; accepted April 8, 2017. Date of publication April 24, 2017; date of current version July 18, 2017. This work was support by the Defence Science and Technology Group, Australian Department of Defence, Commonwealth of Australia. The associate editor coordinating the review of this manuscript and approving it for publication was Mr. Pierre-Marc Jodoin. *(Corresponding authors: Sebastien C. Wong; Victor Stamatescu.)*

S. C. Wong is with the Defence Science and Technology Group, Edinburgh, SA, Australia (e-mail: sebastien@computer.org).

V. Stamatescu, D. Kearney, I. Lee, and M. D. McDonnell are with the Computational Learning Systems Laboratory, School of Information Technology and Mathematical Sciences, University of South Australia, Mawson Lakes, SA, Australia (e-mail: victor.stamatescu@unisa.edu.au).

A. Gatt is with the Australian Department of Defence, RAAF Base Edinburgh, Edinburgh, SA, Australia.

This paper has supplementary downloadable material available at http://ieeexplore.ieee.org., provided by the author. The material includes additional tables not included within the paper itself. The total size of the file is 35.1 kB. Contact sebastien@computer.org for further questions about this work.

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIP.2017.2696744

and spatial information about each object and assigning new measurements to system tracks. This argument is supported by the following contributions:

- a novel object shape learning algorithm, the Shape Estimating Filter (SEF), and its multi-object counterpart, the Competitive Attentional Correlation Tracker using Shape (CACTuS) [8];
- the integration of a feature learning (FL) algorithm with a shape learning algorithm [9];
- CACTuS-FL: the first algorithm to automatically detect and track multiple objects in a video sequence without object-specific prior knowledge [10];
- an online object recognition system that employs an ensemble of single hidden layer feedforward networks (SLFNs) to combine state information from the multi-object tracking algorithm (CACTuS-FL) with the output from an image classifier, the Shallow Convolutional Neural Network (S-CNN).

The rest of this paper is organized as follows. Key recent advances in the areas of multi-object tracking, image classification and object recognition systems are outlined in Section II. An overview of our system is provided in Section III, and this is expanded upon in Sections IV to VI. We demonstrate and examine the efficacy of our approach using Neovision2 benchmark data in Section VII. Finally, Section VIII concludes the paper with a summary of our findings.

II. Related Work

We review related works in the areas of online multi-object detection and tracking, object recognition, and benchmarks for evaluating such systems.

A. Online Detection and Tracking

Recent state-of-the-art online multi-object trackers (e.g. [11]–[15]) follow the *tracking-by-detection* approach, where objects of interest are detected independently in each frame and then uniquely associated with system tracks from the previous frame. The term *online* implies that the underlying algorithm may only use information collected up to the current frame. The aforementioned examples rely on specialised people detectors, with the exception of Urban Tracker [15], which uses background subtraction to detect all types of traffic under the assumption that only moving objects are of interest. This assumption of motion can also be used to form tracklets [16], elementary trajectory fragments, which can clustered together (usually in an off-line manner) to form complete tracks. Although tracking-by-detection algorithms are state-of-the-art, one limitation stems from noisy or missed detections, which can lead to incomplete system tracks. New systems generally aim to mitigate this problem through more reliable object detector design and/or better data association techniques. For example, Breitenstein *et al.* [12] handled occlusions by coupling detection confidence maps with an association scheme based on online-learned classifiers. Bae & Yoon [14] used tracklet confidence to resolve unreliable detections, while their data association stage was based on online discriminative appearance learning. Unlike the aforementioned examples, our system relies instead on the track-before-detect paradigm [7], which is less prone to missing weak detections. Under this approach, the tracking process guides the detection process in order to correlate detections over multiple frames.

B. Recognition

Our approach to object recognition is motivated by the success of deep learning for image classification tasks (see [17] for a recent review). This typically involves training deep (multi-layered) hierarchical models such as Deep Belief Networks (DBNs) [18] and Convolutional Neural Networks (CNNs) [19]. By training complex models with large amounts of data CNNs have set new image classification benchmarks in recent years through models such as AlexNet [20], OverFeat [21] and VGGNet [22]. Rather than relying on such deep architectures, however, our system performs object recognition using a Shallow CNN [23] that limits learning to a single layer. It has been shown to achieve competitive results on standard image classification data sets [24] while being fast to train (when compared with standard deep learning approaches) and maintaining low implementation complexity (few tuneable metaparameters).

C. Benchmark Data

The third key ingredient to our system is domain-specific image sequence data with sufficient object class labeled examples to allow the supervised training of S-CNNs. As mentioned previously, most public multi-object tracking data sets, including those collected for the recent MOT Challenge [25], contain only a single (pedestrian) target class. This focus on people tracking is highlighted by the latest data release, MOT16 [26], in which ground truth object classes are grouped into three broad categories: Target (pedestrian, cyclist, skater), Ambiguous (lying/sitting person, reflection, distractor), Other (car, motorbike, occluder, bicycle). An image sequence data set that does contain multiple object types has been provided by the DARPA Neovision2 [27] program. This data set was collected to enable training and evaluation of Neuromorphic Vision algorithms [28]–[31], which are a class of object recognition algorithms motivated by the emergence of bio-inspired vision sensors [32] and processing hardware (e.g., [33]).

D. Prior Knowledge

As previously discussed, in a tracking-by-detection approach [11]–[14] object specific prior knowledge is embedded into the detector model. Another common prior assumption is that only moving objects are of interest, leading to detection through background subtraction [15], or track formation through tracklets [16]. These assumptions limit tracking to only a specific set of objects, or only moving objects. Furthermore, offline trackers not only make use of prior knowledge of objects, but also incorporate knowledge about future frames, and thus can not run on streaming video. For object recognition using a CNN [22], [24] prior knowledge

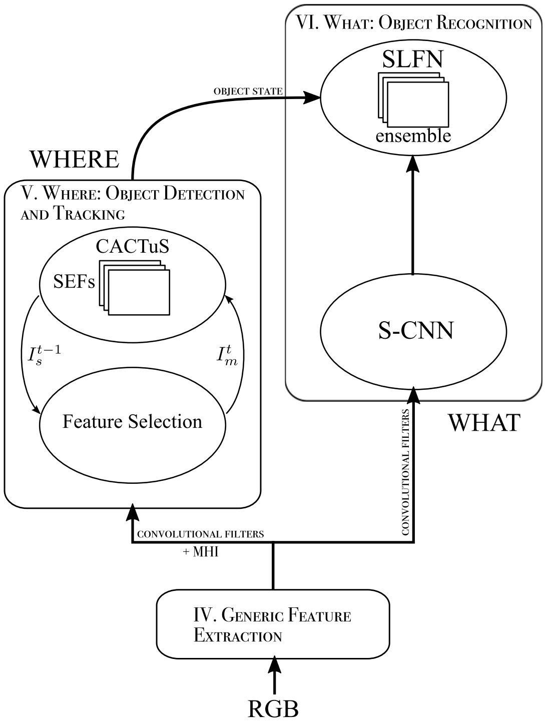
VI. WHAT: OBJECT RECOGNITION
SLFN
ensemble
OBJECT STATE
WHERE
V. WHERE: OBJECT DETECTION AND TRACKING
CACTuS
SEFs
I_s^{t-1}
I_m^t
Feature Selection
S-CNN
CONVOLUTIONAL FILTERS
WHAT
CONVOLUTIONAL FILTERS + MHI
IV. GENERIC FEATURE EXTRACTION
RGB

Fig. 1. Overview of our system for online object recognition comprising the *where* (CACTuS-FL) and *what* (S-CNN and SLFN ensemble) processing streams. The SLFN also combines object state information from the *where* stream.

is strongly embedded into these models through the large training data sets. Thus, there is sufficient scope within the literature to investigate an online system design that transfers prior knowledge from detection and tracking into recognition.

III. Overview

This section provides an overview of our online object recognition system, shown in Figure 1, as well the notation used in this paper.

A. Road Map

Section IV describes the generic feature extraction stage that is used by the *what* and *where* processing streams. The *where* processing stream (Section V) seeks to locate salient objects in the scene and guide the attention of the *what* processing stream (Section VI) to these objects. The *where* stream is handled by the autonomous multi-object tracking algorithm CACTuS-FL [10]. The *what* processing stream relies on a S-CNN architecture [23] that is followed by an ensemble of SLFNs [24], which combines the S-CNN output with object state information from the *where* processing stream. The S-CNN and individual SLFNs are trained offline and then deployed in the online classification of image regions (or patches) associated with system tracks.

B. Notation

Probability mass functions (PMFs) are denoted by capital letters. The subscripts p, m, & s are used to denote predicted, measured and posterior PMFs respectively, while the subscript 0 denotes a constant prior. The superscripts t and $t-1$ denote the current and previous time frames respectively. For brevity, equations that operate only on the current frame do not include superscript t. The notation for normalizing across all bins $\boldsymbol{u}$ of a histogram to form a PMF is abbreviated to $\frac{1}{\Sigma_{\boldsymbol{u}}}$ to avoid additional indexing variables.

IV. Generic Feature Extraction

Good features are those which provide a response that discriminates the object(s) of interest and is invariant to changes in the scene. Here we desire a set of common features that are good for both detection and recognition. Furthermore, for our track everything approach every candidate object (including clutter and stationary objects) should be tracked, and is therefore of interest.

Our tracker, CACTuS-FL, can operate on any set of features, including hand-crafted features [10], however, recent experimental evidence demonstrates that convolutional filters learned by CNNs can produce good features for online visual tracking, enhancing state-of-the-art performance [34], [35]. Furthermore, while motion provides a strong visual cue to the presence of salient objects, which can form an image feature [36] or constrain appearance models [37], this type of cue can not, by itself, detect stationary objects.

For object recognition, the orderless pooling of CNN filter banks can also provide state-of-the-art performance [38], despite earlier evidence to the contrary [39].

Thus, we choose a motion history image (MHI) feature [36], as moving (as well as stationary) objects are of interest, and a biologically inspired convolutional filter bank [40] that is learned in a generative manner to encapsulate the entire scene.

A. Motion History Image

The MHI [36] combines object movement information over an image sub-sequence. To meet the requirement of online tracking we avoid the backward MHI and implement only the forward MHI. This candidate feature is obtained from frame differences between the current image and historical images (through a Markov chain), which highlights the cumulative object motion with a gradient trail that fades away.

B. Convolutional Filters

The 24 convolutional filters, shown in Figure 2, were learned in an unsupervised manner from the first frames of Neovision2 Tower training image sequences 010 – 024 by using a Convolutional Restricted Boltzmann Machine (CRBM) [41]. Each greyscale filter has dimensions of 16 × 16 pixels, which was chosen empirically [40]. In training the generative CRBM model, RGB input images were first downsampled by a factor of two (to a size of 960×540 pixels) to match the resolution of input images used in the online object recognition system. The training images were pre-processed by converting to greyscale, applying the whitening function used by Olshausen & Field [42], subtracting the image mean and normalizing the result by its root mean square (*rms*), as illustrated in Figure 3.

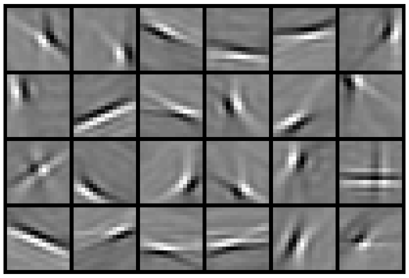

Fig. 2. Bank of 24 generative filters of size 16 × 16 pixels learned using a Convolutional Restricted Boltzmann Machine (CRBM) [41]. The unsupervised training was carried out using the first frames of Neovision2 Tower training sequences. All training image were first converted to greyscale and pre-processed (see main text for details).

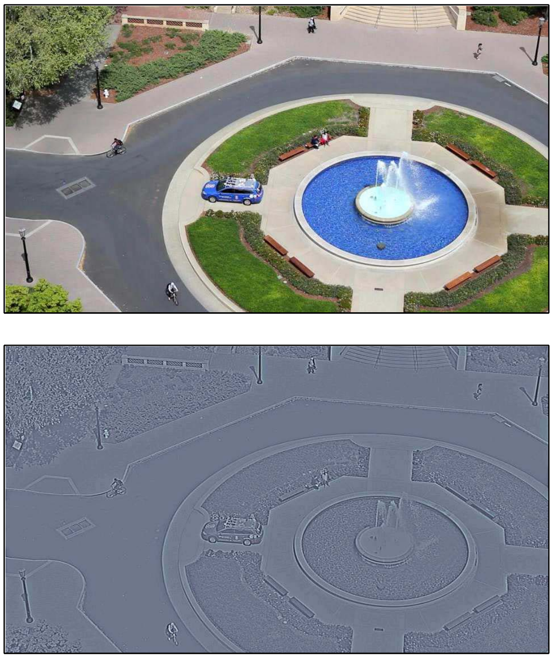

Fig. 3. Sample RGB (top) and pre-processed (bottom) input image, showing frame 61 from Neovision2 [27] Tower image sequence 001, which was first downsampled to a size of 960 × 540 pixels. Image pre-processing involves the application of the whitening function used by Olshausen & Field [42], subtraction of the image mean and normalization of the result by its root mean square (*rms*).

The whitening function applies a combined whitening and low-pass filter with frequency response of the form $fe^{-(f/f_0)^4}$, where f_0 is a cutoff frequency of 200 cycles/image. During the online application of these filters, each new input image also undergoes these pre-processing steps.

V. Where: Object Detection and Tracking

Multiple object tracking algorithms are required to maintain temporally consistent trajectories (state information) for all objects and to uniquely associate new observations with each trajectory. An additional requirement in our design is that tracks are able to self-initialize by automatically converging onto regions of temporally consistent and spatially correlated local saliency. To this end, we couple the track-before-detect paradigm with an adaptive tracking approach (e.g. [5], [6]), so that a state model, which recursively learns both object shape and motion, is able to guide future detections. The unique identities of multiple objects are preserved by correctly associating multiple sub-trackers with new observations. This is accomplished by operating these sub-trackers in competition with one another across the scene.

A. Feature Selection

We first address the problem of autonomous single object detection. Typical object detectors in visual tracking use application-specific knowledge such as hard-coding a fixed set of features that describe a particular object or type of object. By contrast, this paper follows the adaptive method proposed by Collins *et al.* [43], which frames the online selection of a subset of features (from a larger set) as an evolving "object versus local background" two-class classification problem. This *discriminant tracking* approach is analogous to the center-surround mechanisms for attention and saliency that are found in biological vision [44] and enable automatic track initiation.

Every candidate feature $n \in 1, \dots, 25$ (the MHI feature and the 24 convolutional features from Section IV) is used to compute a *feature map* $Z_n^t(\boldsymbol{i})$, which is a representation of the image at frame t in terms of the feature response at each pixel position $\boldsymbol{i}$. Following [43], discriminative features are selected based on the separation of their class-conditioned feature response distributions $F_n^t(u)$ and $B_n^t(u)$, which are 1D histograms extracted for each feature from the object foreground and local background regions, respectively. Here $u \in 1, \dots, 64$ is an index into a histogram of feature response values. In order to extract the object feature response distribution we use the learned object image from the previous frame I_{s}^{t-1}, defined by Eqn. (23), as a pixel weighting mask:

$$F_n^t(u) = \frac{\sum_i I_{\mathrm{s}}^{t-1}(\boldsymbol{i})\, \delta\left(Z_n^t(\boldsymbol{i}) - u\right)}{\Sigma_u}, \tag{1}$$

where δ is the Kronecker delta function. The local background feature response distribution B_n^t is extracted in a similar way, using a weighting mask $1 - I_{\mathrm{s}}^{t-1}$ over an appropriately sized local image patch. Using the learned image I_{s}^{t-1} to precisely identify object pixels leads to a more precise extraction of the feature response distributions than with a bounding box (as used in [43]), reducing background pollution in the feature learning process [9]. This, in turn, provides stronger detections for the tracking process. This feedback between tracking and feature selection is illustrated in Figure 1.

A detection map $\hat{L}_n(\boldsymbol{i})$ is computed for each feature by back-projecting its Likelihood Ratio $L_n(u) = F_n^t(u)/B_n^t(u)$ into its feature map and normalizing: $\hat{L}_n(\boldsymbol{i}) = L_n\left(u = Z_n^t(\boldsymbol{i})\right) / \max(L_n\left(u = Z_n^t(\boldsymbol{i})\right))$, see [43] for the original formulation and [9] for an illustrated example. Online feature selection then involves choosing the most discriminable set of N detection maps, with $N = 6$ chosen empirically, as similar values $(4 - 8)$ yielded comparable tracking performance. By considering the feature response in each pixel of a local image region (i.e. object, local background, or both) as a discrete random variable z_n^t, we use Maximum Marginal Diversity (MMD) [45] to approximate the *infomax space*:

the subset of N features that maximizes its own *mutual information* with the class label random variable $\boldsymbol{c}$. When applied to feature selection in discriminant tracking [44], [46] MMD involves scoring each feature by its mutual information $\mathcal{I}(z_n^t;\boldsymbol{c})$ with the object ($\boldsymbol{c}=1$) and local background ($\boldsymbol{c}=0$) class labels:

$$\mathcal{I}(z_n^t;\boldsymbol{c}) = \sum_{c=0}^{1} p(\boldsymbol{c}=c)\mathcal{R}[p(z_n^t=u|\boldsymbol{c}=c)||p(z_n^t=u)]\ , \tag{2}$$

where $\mathcal{R}[p(u)||q(u)] = \sum_{u\in U} p(u)log_2\frac{p(u)}{q(u)}$ is the Kullback-Leibler divergence between two distributions p and q. Here the class-conditioned feature response distributions $p(z_n^t=u|\boldsymbol{c}=1)$ and $p(z_n^t=u|\boldsymbol{c}=0)$ are given by $F_n^t(u)$ and $B_n^t(u)$, respectively, while $p(z_n^t=u)$ corresponds to the combined object and local background regions.

The most discriminative detection maps are selected by choosing the N highest scores given by Eqn. (2), and these are summed pixel-wise in a weighted average to produce a fused detection map I_{m}^t that serves as input to the tracking algorithm:

$$I_{\mathrm{m}}^t(\boldsymbol{i}) = \sum_{n=1}^{N} w_n \hat{L}_n(\boldsymbol{i})\,. \tag{3}$$

The weights in Eqn. (3) are given by $w_n = \mathrm{I}(z_n^t;\boldsymbol{c})\times B$, where the *similarity score* B is the Bhattacharyya coefficient [47]:

$$B = \sum_u \sqrt{F_{n,\mathrm{m}}^t(u)\,F_{n,\mathrm{s}}^{t-1}(u)}, \tag{4}$$

which rewards temporal consistency between the object feature response $F_{n,\mathrm{m}}^t$ measured in the current frame according to Eqn. (1) and an object feature response learned up to the previous frame $F_{n,\mathrm{s}}^{t-1}$. The learned posterior feature response $F_{n,\mathrm{s}}^t$ is updated at each frame by

$$F_{n,\mathrm{s}}^t(u) = \frac{F_{n,\mathrm{s}}^{t-1}(u)\,F_{n,\mathrm{m}}^t(u)}{\Sigma_u}. \tag{5}$$

B. Shape Estimating Filters

We next address the problem of adaptively learning an object state model, which includes a probabilistic representation of its shape. The proposed solution is a single object tracker called the Shape Estimating Filter (SEF) [8], which combines spatiotemporal information from past frames with new measurements to recursively estimate the object position, velocity and shape. A SEF autonomously correlates recurring saliency from each new fused detection map into shape and trajectory estimates.

Assuming that only a single object is present in an image, the 2D PMF $I(\boldsymbol{i})$ is used to describe the probability that a given pixel $\boldsymbol{i}=(i_1,i_2)$ belongs to that object. The PMF $I(\boldsymbol{i})$ can then be factored into 2D PMFs for shape $S(\boldsymbol{j})$ and position $X(\boldsymbol{x})$. Here $X(\boldsymbol{x})$ represents the probability that the object center of mass has position $\boldsymbol{x}=(x_1,x_2)$, while $S(\boldsymbol{j})$ is proportional to the probability that the pixel $\boldsymbol{j}=(j_1,j_2)$ is part of the object. The vectors $\boldsymbol{i}$, $\boldsymbol{j}$ and $\boldsymbol{x}$ are considered 2D random variables operating on the set of integers. The relationship between image, position and shape random variables is given by $\boldsymbol{i}=\boldsymbol{x}+\boldsymbol{j}$, which can be expressed as $\boldsymbol{x}=\boldsymbol{i}-\boldsymbol{j}$, or as $\boldsymbol{j}=\boldsymbol{i}-\boldsymbol{x}$. This relationship allows the shape of an object to be decoupled from its position in the image.

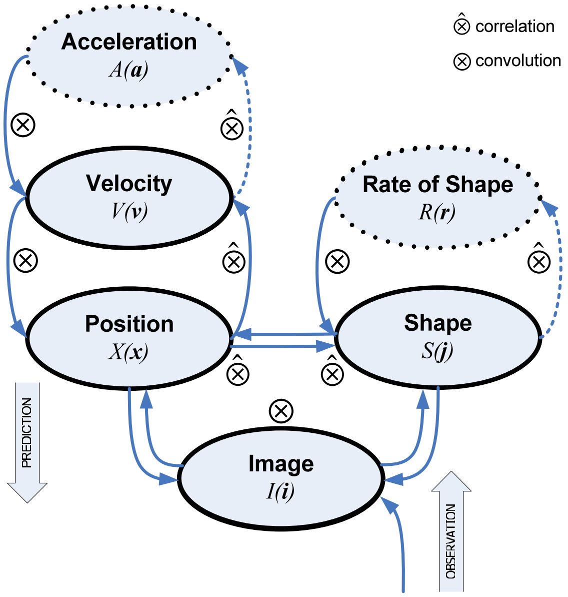

Fig. 4. The hierarchical state model of the Shape Estimating Filter (SEF) [8]. Predictions are propagated from the top-down and new observations from the bottom-up. Predictions and observations are combined at each layer to provide an approximate Bayesian update of the state model.

In order to describe the object motion across a sequence of images in an adaptive manner, 2D random variables are used to model acceleration $\boldsymbol{a}$ and velocity $\boldsymbol{v}$. These variables are described by the PMFs $A(\boldsymbol{a})$ and $V(\boldsymbol{v})$, respectively. Assuming a simplified Euler motion (non-rotational point-mass) for the object and that $\Delta t = t-(t-1) = 1$ leads to the following relationships: $\boldsymbol{v}^t = \boldsymbol{v}^{t-1}+\boldsymbol{a}^t$ and $\boldsymbol{x}^t = \boldsymbol{x}^{t-1}+\boldsymbol{v}^t$. Rearranging, this gives: $\boldsymbol{v}^t = \boldsymbol{x}^t - \boldsymbol{x}^{t-1}$ and $\boldsymbol{a}^t = \boldsymbol{v}^t - \boldsymbol{v}^{t-1}$.

To handle deformable objects, the 2D PMF $R(\boldsymbol{r})$ is defined as the change in shape from one frame to the next, which is described by the random variable relationship $\boldsymbol{r} = \boldsymbol{j}^t - \boldsymbol{j}^{t-1}$.

These random variable relationships are used to build the SEF algorithm, using the operations of convolution $\otimes$ and cross-correlation $\hat{\otimes}$, as illustrated in Figure 4. The SEF state-space hierarchy provides a framework for combining top-down predictions with bottom-up sensory measurements through a Bayesian update process.

Predictions are made by traversing down the state model hierarchy (starting in the top left of Figure 4) according to:

$$\boldsymbol{v}^t = \boldsymbol{v}^{t-1}+\boldsymbol{a} \Rightarrow V_{\mathrm{p}}^t = V_{\mathrm{s}}^{t-1}\otimes A_0, \tag{6}$$

$$\boldsymbol{x}^t = \boldsymbol{x}^{t-1}+\boldsymbol{v}^t \Rightarrow X_{\mathrm{p}}^t = X_{\mathrm{s}}^{t-1}\otimes V_{\mathrm{p}}^t, \tag{7}$$

$$\boldsymbol{j}^t = \boldsymbol{j}^{t-1}+\boldsymbol{r} \Rightarrow S_{\mathrm{p}}^t = S_{\mathrm{s}}^{t-1}\otimes R_0, \tag{8}$$

$$\boldsymbol{i}^t = \boldsymbol{j}^t+\boldsymbol{x}^t \Rightarrow I_{\mathrm{p}}^t = S_{\mathrm{p}}^t\otimes X_{\mathrm{p}}^t, \tag{9}$$

where A_0 and R_0 are 2D Gaussian priors.

Given I^t_{m}, measurements are made by traversing up the state model hierarchy (starting at the bottom of Figure 4) according to:

$$\boldsymbol{x}^t = \boldsymbol{i}^t - \boldsymbol{j}^t \Rightarrow X^t_{\mathrm{m}} = I^t_{\mathrm{m}} \hat{\otimes} S^t_{\mathrm{p}}, \tag{10}$$

$$\boldsymbol{v}^t = \boldsymbol{x}^t - \boldsymbol{x}^{t-1} \Rightarrow V^t_{\mathrm{m}} = X^t_{\mathrm{m}} \hat{\otimes} X^{t-1}_{\mathrm{s}}, \tag{11}$$

$$\boldsymbol{j}^t = \boldsymbol{i}^t - \boldsymbol{x}^t \Rightarrow S^t_{\mathrm{m}} = I^t_{\mathrm{m}} \hat{\otimes} X^t_{\mathrm{s}}. \tag{12}$$

An approximate Bayesian update scheme is used to combine top-down predictions with bottom-up observations. The posterior PMFs of position, velocity and shape are described by:

$$X^t_{\mathrm{s}}(\boldsymbol{x}) = \frac{X^t_{\mathrm{m}}(\boldsymbol{x})\, X^t_{\mathrm{p}}(\boldsymbol{x})}{\Sigma_{\boldsymbol{x}}}, \tag{13}$$

$$V^t_{\mathrm{s}}(\boldsymbol{v}) = \frac{V^t_{\mathrm{m}}(\boldsymbol{v})\, V^t_{\mathrm{p}}(\boldsymbol{v})}{\Sigma_{\boldsymbol{v}}}, \tag{14}$$

$$S^t_{\mathrm{s}}(\boldsymbol{j}) = \frac{S^t_{\mathrm{m}}(\boldsymbol{j})\, S^t_{\mathrm{p}}(\boldsymbol{j})}{\Sigma_{\boldsymbol{j}}}. \tag{15}$$

C. Competitive Attention Correlation Tracking Using Shape

Finally, we address the problem of automatically associating new measurements to multiple system tracks. The proposed solution, which extends the work of Strens and Gregory [48], operates multiple SEFs simultaneously in a competitive attentional framework designed to enforce the tracking of multiple objects. Under this scheme, the SEFs track everything in the scene, including parts of the background or sources of clutter, so that every new measurement is assigned to the SEF that best describes that measurement [10].

For each frame t, the multi-object tracking algorithm operates $k = 1, .., K$ individual SEFs. The bottom-up input of each SEF k is modulated by an association term $\beta^k(\boldsymbol{i})$, so that Eqn. (3) becomes

$$I^k_{\mathrm{m}}(\boldsymbol{i}) = \beta^k(\boldsymbol{i}) \sum_{n=1}^{N} \omega_n \hat{L}_n(\boldsymbol{i}). \tag{16}$$

As shown in Figure 5, top-down modulation provides each SEF with a spatial area of attention to collect new measurements. The term $\beta^k(\boldsymbol{i})$ is computed from learned predictions about the expected image:

$$\beta^k(\boldsymbol{i}) = \frac{I^k_{\mathrm{p}}(\boldsymbol{i})}{\sum_{j=1}^{K} I^j_{\mathrm{p}}(\boldsymbol{i})}. \tag{17}$$

This selective attentional mechanism modifies the bottom-up input to each SEF, enabling individual SEFs to selectively ignore pixels that are strongly claimed by another SEF, where $0 \leq \beta^k(\boldsymbol{i}) \leq 1$ describes the strength of the claim of pixel at location $\boldsymbol{i}$ by SEF k.

By assuming a 2D Gaussian prior shape S_0, an additional attentional mechanism is introduced by replacing Eqn. (10) with

$$X^k_{\mathrm{m}}(\boldsymbol{x}) = I^k_{\mathrm{m}}(\boldsymbol{i})\, \hat{\otimes} (S^k_{\mathrm{p}}(\boldsymbol{j})\, S_0(\boldsymbol{j})). \tag{18}$$

This introduces a self-centering capability to the system [49], which reduces the problem of *model drift* [50] that affects correlation trackers [6].

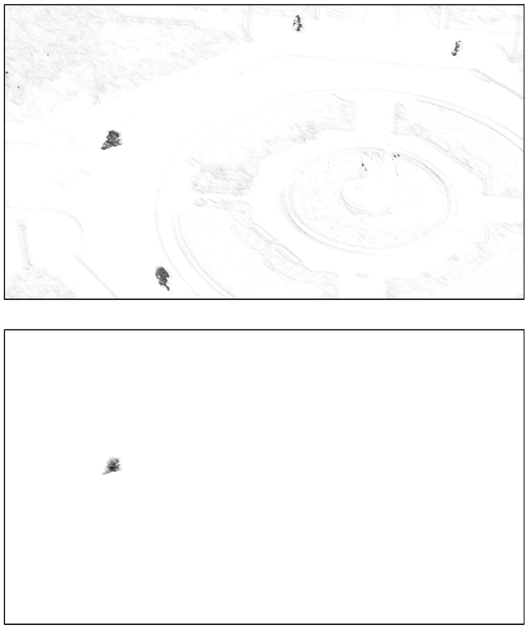

Fig. 5. The selective attentional mechanism of CACTuS-FL, for SEF $k = 33$, which is tracking a cyclist, in frame 61 of Neovision2 Tower image sequence 001. The $I^k_{\mathrm{m}}(\boldsymbol{i})$ fused detection map from Eqn. (3) (top) is modulated by the spatial area of attention $\beta^k(\boldsymbol{i})$ to form the bottom-up input for the SEF (bottom).

In order to encourage SEFs to track multiple objects, Eqn. (13) is modified by a *winner-take-more* competitive mechanism [48]. Under this scheme, which has the inherent assumption that different objects tend to occupy different positions, K separate SEFs compete over position $\boldsymbol{x}$ to track every object in the scene. Each SEF k competes against all SEFs for its own share of the total association probability $\sum_{l=1}^{K} C^l(\boldsymbol{x}) = 1$ at each position $\boldsymbol{x}$. The individual association probability C^k, which is shown for a single SEF in Figure 6, is computed using the predicted position $X^k_{\mathrm{p}}(\boldsymbol{x})$ according to

$$C^k(\boldsymbol{x}) = \frac{X^k_{\mathrm{p}}(\boldsymbol{x})}{\sum_{l=1}^{K} X^l_{\mathrm{p}}(\boldsymbol{x})}. \tag{19}$$

The update of position $X^k_{\mathrm{s}}(\boldsymbol{x})$ for each SEF k in Eqn. (13) is then modified to include this spatial attention modulation for each SEF

$$X^k_{\mathrm{s}}(\boldsymbol{x}) = \frac{X^k_{\mathrm{m}}(\boldsymbol{x})\, X^k_{\mathrm{p}}(\boldsymbol{x})\, C^k(\boldsymbol{x})}{\Sigma_{\boldsymbol{x}}}. \tag{20}$$

An example of $X^k_{\mathrm{s}}(\boldsymbol{x})$ is shown for a single SEF in Figure 6. This mechanism enables the SEF that best describes the position state estimate for a particular object to converge on a region corresponding to that object and exclude other SEFs from that region. This competition encourages SEFs to track different objects, rather than all SEFs converging on the most salient object in the scene.

The shape of the object is observed using the relationship $\boldsymbol{j} = \boldsymbol{i} - \boldsymbol{x}$. First, the best estimate of the object location in the current fused detection map $I^k_{\mathrm{m}}(\boldsymbol{i})$ is extracted from the posterior position PMF $X^k_{\mathrm{s}}(\boldsymbol{x})$ according to $X^k_{\mathrm{smax}}(\boldsymbol{x}) = \delta\left(\mathrm{argmax}_1\left(\mathrm{X}^k_{\mathrm{s}}(\boldsymbol{x})\right) - \boldsymbol{x}\right)$, where argmax_1 returns

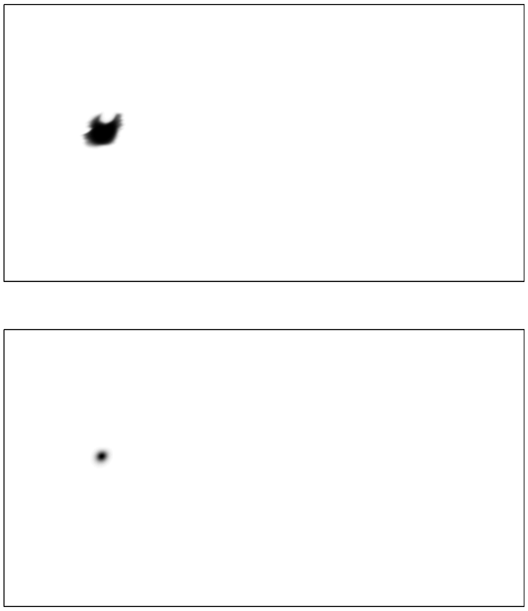

Fig. 6. The association probability $C^k(\boldsymbol{x})$ (top) and the posterior position PMF $X_\mathrm{s}^k(\boldsymbol{x})$ (bottom) for SEF $k = 33$, which is tracking a cyclist, in frame 61 of Neovision2 Tower image sequence 001.

one maximum. Next, the PMF $X_\mathrm{smax}^k(\boldsymbol{x})$ is used to extract the observed shape $S_\mathrm{m}^k(\boldsymbol{j})$ from $I_\mathrm{m}^k(\boldsymbol{i})$ using:

$$S_\mathrm{m}^k(\boldsymbol{j}) = I_\mathrm{m}^k(\boldsymbol{i})\,\hat{\otimes} X_\mathrm{smax}^k(\boldsymbol{x})\,. \tag{21}$$

The process used to update shape has been adapted from [6] as a way to mitigate model drift. First, the degree of match ρ is computed as the L^2 normalized cross-correlation at $\boldsymbol{j} = (0, 0)$ of the measured and predicted shapes, $\rho = \hat{S_\mathrm{m}}((0, 0))\ \hat{\otimes}\ \hat{S}_\mathrm{p}((0, 0))$, where $0 \le \rho \le 1$ is a scalar.

Next the parameter α is computed as $\alpha(\rho, \lambda) = H(\rho-\lambda)\rho^2$, where $H(x)$ is the unit step function and the threshold λ acts as the vigilance parameter [51] to ensure that very poor observations are not introduced into memory, see [6] for details.

This controls the degree by which the posterior shape $S_\mathrm{s}^k(\boldsymbol{j})$ is influenced by new observations $S_\mathrm{m}^k(\boldsymbol{j})$, or prior expectations $S_\mathrm{p}^k(\boldsymbol{j})$, and thus Eqn. (15) is replaced with

$$S_\mathrm{s}^k(\boldsymbol{j}) = (S_\mathrm{m}^k(\boldsymbol{j}))^\alpha (S_\mathrm{p}^k(\boldsymbol{j}))^{(1-\alpha)}. \tag{22}$$

A high degree of match results in a large update of the shape $S_\mathrm{s}^k(\boldsymbol{j})$, while a low degree of match leads to a small update. The resulting posterior shape is shown for a single SEF in Figure 7.

Rather than combining the predicted and measured images, the posterior image $I_\mathrm{s}^k(\boldsymbol{i})$, is computed according to a maximum *a posteriori* approach based on the shape and position:

$$I_\mathrm{s}^k(\boldsymbol{i}) = S_\mathrm{s}^k(\boldsymbol{j}) \otimes X_\mathrm{smax}^k(\boldsymbol{x})\,. \tag{23}$$

The posterior image, which is shown for a single SEF in Figure 7, then provides top-down guidance for new detections according to Eqn. (1) in the object detection stage.

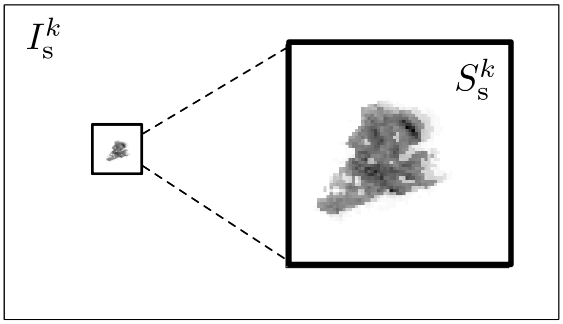

Fig. 7. The posterior image $I_\mathrm{s}^k(\boldsymbol{i})$, for SEF $k = 33$, which is tracking a cyclist, in frame 61 of Neovision2 Tower image sequence 001. The inset shows the corresponding posterior shape $S_\mathrm{s}^k(\boldsymbol{j})$.

D. Tracking Output

Each SEF k outputs the posterior image I_s^k of the object that it is tracking. This learned image is then parameterized by calculating its ellipse of second order moments [52]. The 2σ length and width along the ellipse major and minor axis, respectively, are used to define an oriented output bounding box for each object, in every frame.

VI. What: Object Recognition

This section describes the S-CNN and SLFN ensemble classification algorithms, detailing their supervised offline training and application to online object recognition.

While a variety of image classifiers could act as the *what* processing stream, S-CNNs and SLFNs, in which only the output layer weights are learned, have the advantage of being fast to train (on the order of minutes on standard PCs) and hence are well suited to tasks that require frequent domain-specific re-training.

A. Shallow Convolutional Neural Network

1) S-CNN Offline Training: Here we summarise our application-specific S-CNN implementation, while an in depth description of the algorithm may be found in [23]. The network architecture consists of five layers: an input image pixel layer, three hidden unit layers, and an output layer. Only the weights that project to the final layer are learned. The S-CNN can be divided into two conceptual stages: a convolutional filtering and pooling stage formed by the first two hidden layers, which extract translation and scale invariant features, and a classification stage consisting of the third hidden layer and the output layer.

Stage 1 (Convolutional Filtering and Pooling): Each domain-specific S-CNN is trained on a single batch of image patches of size 61×61 pixels. The bank of 24 visual processing filters shown in Figure 2, which serve as generic object detectors in the *where* processing stream, are reused here as the first layer of convolutional filters. Following [23], the first hidden layer units are obtained by applying a termwise nonlinear function $g_1(u) = u^2$. The first hidden layer activations are average-pooled and down-sampled by applying a uniform low pass filter with a pooling size of 18 × 18 pixels and stride of 6 pixels. Finally a termwise nonlinear function of form $g_2(u) = u^{0.25}$ is applied to obtain the image features.

Stage 2 (Classificationv): The features from the second hidden layer are concatenated and linearly projected onto 12000 hidden units using a fully-connected set of real-valued input weights, which is set only once during training following the method of [53]. Applying the termwise squaring function $g_1(u)$ to every mapped feature yields the third hidden layer activations. Output labels can then be predicted by linearly mapping these activations using a set of fully-connected output weights obtained as in [23], and described further below.

In order to train the S-CNN, pre-processed (see Section III) 61 × 61 pixel image patches are extracted from its training image sequences. Using the Neovision2 Tower training videos, this involves extracting image patches from 15 image sequences ($010 - 024$) based on the positions of the ground truth bounding boxes. To simulate the effect of object tracking during training, the centre of each patch includes positional Gaussian random jitter about the object ground truth location, with a standard deviation of 10 pixels in both the x and y axis directions. In each patch, the pixels outside a central circular spatial attention region of radius 30 pixels are set to 0. Additional patches are randomly extracted from background regions in each training image to provide training examples for a background Clutter class. The training examples are then randomly shuffled and the class abundances are balanced so that the number of training examples is uniformly spread among four classes: Car, Person, Cyclist and Clutter.

Given that the convolutional filters, the pooling parameters and the classifier input weights are fixed, the offline training algorithm only involves finding the set of optimal output weights. These are obtained by forming a set of linear equations from a single batch of training class labels and output layer activations and solving for the output weights using least squares regression as in [23].

2) S-CNN Online Object Recognition: In online processing, raw pixel image patches, which are centered on the position of each SEF, are presented as input to the trained S-CNN in the form of an input vector $\mathbf{x}_{\text{test}}$. Following the matrix notation of [23], the S-CNN output for each patch is the predicted label vector $\mathbf{y}_{\text{test}}$ whose length corresponds to the number of classes:

$$\mathbf{y}_{\text{test}} = \mathbf{W}_{\text{out}}\ g_1(\mathbf{W}_{\text{in}}\ g_2(\mathbf{W}_{\text{Pool}}\ g_1(\mathbf{W}_{\text{Filter}}\mathbf{x}_{\text{test}}))), \tag{24}$$

where the convolution matrices $\mathbf{W}_{\text{Filter}}$ and $\mathbf{W}_{\text{Pool}}$ apply convolutional filtering and pooling, respectively, the matrix $\mathbf{W}_{\text{in}}$ corresponds to the fully-connected input weights, and the matrix $\mathbf{W}_{\text{out}}$ corresponds to the fully-connected output weights. If the S-CNN were used on its own, without applying the SLFN, the predicted class would be given by the index of the maximum value in $\mathbf{y}_{\text{test}}$.

B. Single Hidden Layer Feedforward Network Ensemble

1) SLFN Offline Training: We next train SLFNs to predict the ground truth class label associated with each SEF by combining object state (*where* stream) information and the corresponding S-CNN (*what* stream) output unit activations. To reduce the potential for over fitting, an ensemble [54] of seven small SLFNs are trained separately. Each SLFN employs the same type of architecture as the S-CNN classification stage. The input features of the first six SLFNs are linearly mapped onto 320 hidden units using a fixed set of fully-connected input weights that are set randomly only once in training [24]. Using the same approach, the seventh SLFN instead maps the vector form of the 71 × 71 pixel posterior shape (e.g. see Figure 7) onto a layer of 12800 hidden units. In all SLFN instances, a termwise logistic sigmoid function $g(u) = 1/(1 + \exp(-u))$ is applied to each hidden unit, and these activations are mapped to the output units using an optimal set of fully-connected output weights that is learned during training. As was done for the S-CNN, the optimal output weights for each SLFN are obtained in one shot using least squares regression.

The training procedure for the first six SLFNs relies on a set of 10 features, comprising the *softmax* of the S-CNN output vector from Eqn. (24) (6 features), and state variables in the form of predicted object bounding box width, length and absolute inclination angle (about the x-axis), as well as the energy of the posterior position PMF: $\sum_{\boldsymbol{x}} (X_{\text{s}}^{k}(\boldsymbol{x}))^2$, which measures the degree to which a SEF has collapsed (or latched) onto its object. Before training the SLFN, each state variable is pre-processed by subtracting the training sample mean and then normalizing by the *rms* of the entire mean-subtracted training sample, and these parameters are saved and also used in online pre-processing. Six SLFNs are then trained using a 65 dimension input feature vector that is formed by multiplying pairs of features, for all unique pairwise combinations plus the individual unpaired features themselves.

In order to accumulate training examples, CACTuS-FL and the S-CNN are applied the Neovision2 Tower training sequences 001, 010, 013, 014, and 017, for which we added unique object IDs by hand to the original ground truth data. This allows optimal associations to be made between SEF bounding boxes and ground truth bounding boxes using the Munkres algorithm [55]. This mapping procedure is used to assign true class labels to each tracked object, which produces the required set of training labels. The first six SLFNs are trained by applying a bagging technique that randomly divides the data among six separate sets. In the case of the posterior shape based (seventh) SLFN, all of the training data is used in a single batch.

2) SLFN Online Object Recognition: During online processing, given the vector $\mathbf{f}_{\text{test}}^{c}$ of input features appropriate for each trained SLFN $c = 1, \ldots, 7$, the output unit vector $\mathbf{y}'^{c}_{\text{test}}$ is given by:

$$\mathbf{y}'^{c}_{\text{test}} = \mathbf{W}'^{c}_{\text{out}}\ g(\mathbf{W}'^{\ c}_{\text{in}}\ \mathbf{f}_{\text{test}}{}^{c}), \tag{25}$$

where the matrices $\mathbf{W}'^{\ c}_{\text{in}}$ and $\mathbf{W}'^{c}_{\text{out}}$ correspond to each of SLFN input and trained output weights, respectively. Finally, a *softmax* function is applied to each output vector, and the SLFNs are used in an ensemble by combining their output through an element-wise sum:

$$\mathbf{y}'_{\text{ensemble}} = \sum_{c=1}^{7} \text{softmax}(\mathbf{y}'^{c}_{\text{test}}). \tag{26}$$

The class predicted by the online object recognition system is given by the index of the maximum value in $\mathbf{y}'_{\text{ensemble}}$.

VII. Experimental Evaluation

This section describes the data used in our experiments together with a summary of previous evaluations of the main system components. The section also details our experimental parameters, highlighting any use of prior knowledge, as well as explaining the performance evaluation metrics. The system performance is then compared against existing online object recognition benchmark results [28], while the impact of the main components (CACTuS-FL, S-CNN, SLFN ensemble) on its performance is also investigated.

A. Previous Experiments

We summarise previous experimental results for key components of our online object recognition system: generic feature extraction, CACTuS-FL and the S-CNN, using separate visual tracking and image classification benchmarks.

1) Generic Feature Extraction: The choice of convolutional filter bank and individual filter size were made based on experiments [40] using the Neovision2 Tower training sequence 001, where the multi-object tracking performance for all object classes was evaluated in terms of the best Recall (60.37%) and tracking precision MOTP (43.44%).

2) Where–CACTuS-FL: CACTuS-FL was evaluated using 8 videos from the VOT2013 single object tracking benchmark [56]. In these experiments [10] the robustness of the tracker was measured by the number of tracking failures. CACTuS-FL incurred 4 tracking failures, as compared to the well known TLD algorithm [57] that had 39 tracking failures, and the state-of-the-art LGT algorithm [58] that had 2.75 tracking failures. A qualitative evaluation on multi-object tracking using soccer videos was also presented.

3) What–S-CNN: The S-CNN was previously evaluated [23] on the MNIST [59], NORB [60], SVHN [61] and CIFAR-10 [62] benchmark data sets, achieving image classification error rates of 0.37%, 2.21%, 3.96% and 24.14%, respectively. In the case of MNIST and NORB, this represents state-of-art image classification accuracy if excluding techniques that perform training set data augmentation [63]. Furthermore, the experiments showed that S-CNNs are robust in the sense that the same network metaparameters can be applied across different data sets to yield similar performance to that obtained by tuning the metaparameters for each data set.

B. Online Object Recognition Experiments

While the key aspects of our online object recognition system have been tested separately, testing the integrated system requires a MOT data set with multiple target classes. As outlined in Section II, existing MOT datasets only exercise tracking of a single class, and often provide pre-computed detections [25], [26]. By contrast, we require a multi-object, multi-class benchmark and this is provided by Neovision2 [27]. This set of challenging image sequences, captured under varying environmental conditions, contains numerous targets, including stationary objects, which can undergo occlusions by neighbouring objects or background clutter.

1) Benchmark Data: The Neovision2 Tower data set consists of 50 training and 50 test videos captured from an elevated camera. In both Tower training and test sets the camera is rotated by 90° after the first 24 videos, and, given that this changes the ground sample distance (pixel/m), we limit our study to videos $001 - 024$ in both the training and test sets.

Each image sequence was recorded at 29.97 frames/s and has 871 annotated frames, with ground truth data consisting of a class label and oriented bounding box coordinates for each object of interest. Five target object classes are present in the Tower data domain (Car, Truck, Bus, Person, Cyclist) and, through random sampling of the background, we include a sixth Clutter class in order to identify SEFs that are tracking background objects. Due to the scarcity of Truck and Bus training examples, however, we avoid training and testing on the (few) videos that do contain these object types, which leaves the following four classes: Car, Person, Cyclist, Clutter. Following these criteria and also simply excluding any video found to have clearly incorrect ground truth annotations, we select 12 Neovision2 Tower test set videos: 001, 002, 009, 010, 012, 013, 017, 018, 019, 021, 022, 023. This set of videos, which contains 82139 ground truth objects across 10452 image frames, was tested only once.

C. Experimental Parameters

1) Prior Knowledge: While the majority of architectural decisions and run time parameter settings for our system were chosen empirically based on previous experiments [6], [10], [23], some were tuned for the Neovision2 Tower training data set. These system parameters constitute domain-specific prior knowledge and are listed in Table I, which outlines the reason behind each choice.

2) System Initialization: CACTuS-FL is initialized in the first frame of an image sequence by positioning the SEFs at regular intervals in a 14×8 rectangular grid across the scene. The position, shape and velocity PMFs for each SEF are initialized using isotropic 2D Gaussian distributions.

D. Performance Evaluation Metrics

The Neovision2 object recognition performance metrics [64] are based on the degree of spatial overlap $d_{t,i,k}$ between each ground truth bounding box region $\mathbf{r}_{t,i}^{GT}$ and every candidate bounding box region $\mathbf{r}_{t,k}^{SEF}$ output by the k^{th} SEF:

$$d_{t,i,k} = \frac{\mathbf{r}_{t,i}^{GT} \cap \mathbf{r}_{t,k}^{SEF}}{\mathbf{r}_{t,i}^{GT} \cup \mathbf{r}_{t,k}^{SEF}}, \tag{27}$$

where t refers to the image frame and i is the ground truth index.

To evaluate the online object recognition performance we use the publicly available Neovision2 evaluation tool [64]. This uses the Munkres algorithm [55] to find optimal SEF to ground truth bounding box associations in each frame for a spatial overlap threshold of $T_d = 0.2$. For each image sequence s, the system performance in detecting each target object class o (i.e. Car, Person, Cyclist) is measured

TABLE I
PRIOR KNOWLEDGE

Parameter	Symbol	Value	Justification
Size of learned CRBM convolutional filter		16 × 16 pixels	Tuned for tracking performance of all objects in the scene [40].
Size of posterior shape		71 × 71 pixels	Chosen by eye to ensure that $S_s^k(\boldsymbol{j})$ is large enough to encompass and collapse on any person, cyclist or car.
Total number of SEFs	K	112	Chosen to encourage competition between SEFs over the entire scene.
Size of second order moment ellipse		2σ	Chosen so that the bounding boxes of collapsed SEFs encompass their object, but do not extend too far beyond this.
Size of image patch		61 × 61 pixels	Chosen so that the S-CNN input captures a large fraction of cars, but limits the extent of the background around people or cyclists.

using the Normalized Multiple Object Thresholded Detection Accuracy (NMOTDA):

$$\text{NMOTDA}_{s,o} = 1 - \frac{\sum_t (\text{FN}_{t,o} + \text{FP}_{t,o})}{\sum_t \text{GT}_{t,o}}, \qquad (28)$$

where in each frame t, $\text{GT}_{t,o}$, $\text{FN}_{t,o}$ and $\text{FN}_{t,o}$ are the number of ground truth objects, false negatives, and false positives, respectively, of object class o. NMOTDA is reported as a number in the range $(-\infty, 1]$. The $\text{NMOTDA}_{s,o}$ scores are then aggregated across all image sequences to yield the Weighted Normalized Multiple Object Thresholded Detection Accuracy (WNMOTDA):

$$\text{WNMOTDA}_o = \frac{\sum_s \text{NMOTDA}_{s,o} \times \text{GT}_{s,o}}{\sum_s \text{GT}_{s,o}}, \qquad (29)$$

where the weight $\text{GT}_{s,o}$ is the total number of ground truth objects belonging to class o that are present in image sequence s. Average NMOTDA and Average WNMOTDA are also calculated for all object types according to Eqn. (28) and Eqn. (29), respectively, by ignoring the object class label o.

In sequences for which we have added ground truth object IDs, such as 001, we also apply the CLEAR MOT multi-object tracking metrics [65]. Following the implementation of [66], the optimal mapping between SEFs and ground truths is found across all frames in terms of the total spatial overlap. The associated ground truth and SEF pairs are then identified as *matches* $j \equiv (i, k)$ when $d_{t,j}$ exceeds a user-defined threshold T_d, which can be varied between 0 and 1. Figure 8 illustrates some examples of matched SEF/ground truth pairs for $T_d = 0.2$. This procedure is used to assign ground truth class labels to SEFs for the purpose of generating SLFN training data.

E. Results

Table II lists the training and validation classification accuracies obtained by applying the S-CNN to image patches extracted around clutter and randomly jittered ground truth object positions. The un-jittered validation set accuracies obtained here on training video 001 are comparable to the range of accuracies (96.77% − 100%) obtained by a deep CNN [31] on Neovision2 Tower data. The validation results in Table II indicate that the classification accuracy of the S-CNN degrades considerably, especially for the Person class, when random position jitter is applied to the image patches, despite the fact that the same approach was used for the training patches.

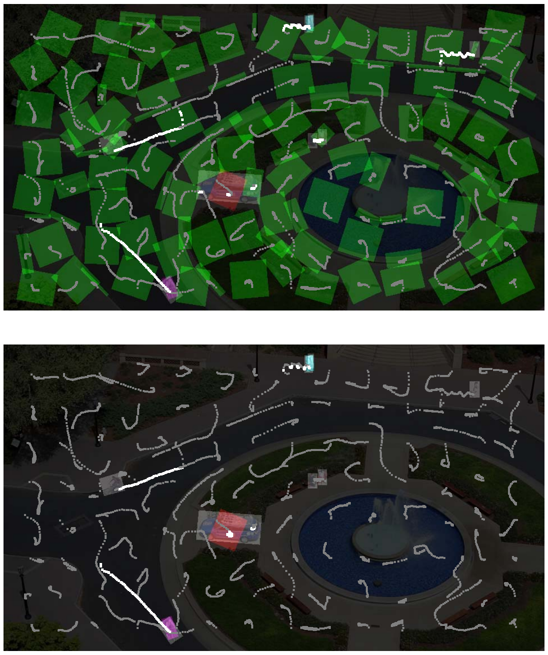

Fig. 8. Neovision2 Tower image sequence 001 frame 61 showing SEF output from the current frame and CLEAR MOT [65] tracks up to and including this frame. CACTuS-FL SEF tracks are shown as grey dots and those identified as SEF and ground truth matches are shown as white dots. Ground truth bounding boxes are indicated by shaded grey rectangles, which are centered on every car, person or cyclist in the scene. Bounding boxes estimated by CACTuS-FL in the current frame, which are computed by parameterizing the object shape learned by each SEF, are shown in green, red, magenta and cyan for SEFs classified as Clutter, Car, Cyclist and Person, respectively. The top plot shows all bounding boxes, while the bottom plot shows only those bounding boxes that have not been classified as Clutter.

TABLE II
S-CNN CLASSIFICATION ACCURACY FOR TOWER TRAINING (010 − 024) AND VALIDATION (001) SEQUENCE IMAGE PATCHES

Data set	Training (010 − 024) with jitter	Validation (001) without jitter	Validation (001) with jitter
Car	99.89%	100.00%	99.89%
Person	96.76%	95.45%	78.42%
Cyclist	95.52%	99.46%	96.07%
Clutter	99.07%	97.61%	97.67%

In order to gain some intuition into the impact of tracking and classification accuracy on NMOTDA, we attempt to decouple the two effects in Figure 9, which shows validation results from Tower training sequence 001. Starting with perfect

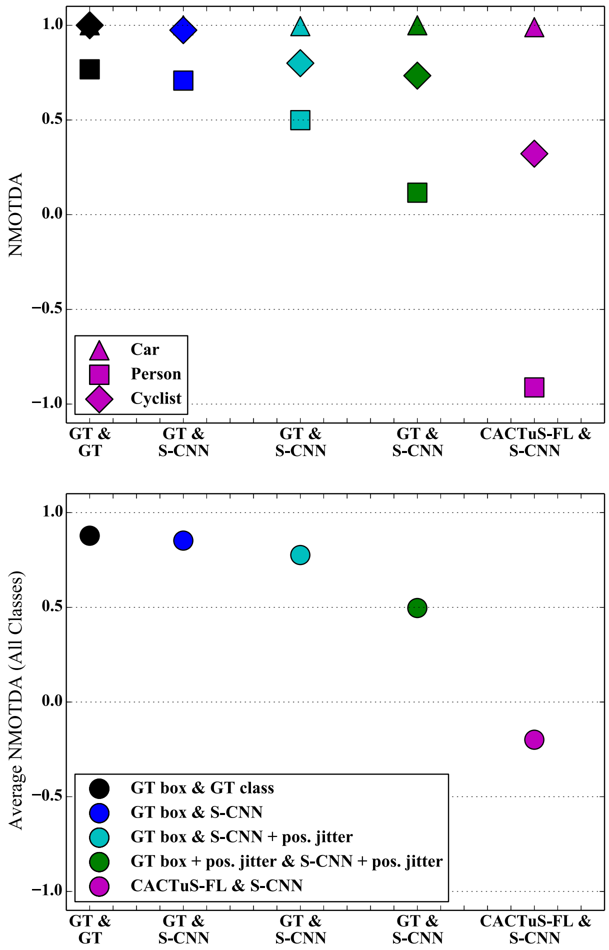

Fig. 9. NMOTDA scores for Neovision2 Tower training sequence 001, used here for validation. The top plot shows NMOTDA for each class, the bottom plot shows the Average NMOTDA scores for all classes. Black markers correspond to taking the ground truth position and class label data as system outputs (e.g. perfect object tracking and classification). Blue markers correspond to perfect tracking and S-CNN based classification. Cyan markers correspond to perfect tracking, but here the S-CNN has random jitter applied to its input image patch positions. Green markers show the case when position jitter is also applied to the ground truth bounding box to simulate the effect of imperfect tracking. In all cases mentioned thus far, the number of SEFs operating in a frame is equal to the number of ground truths in that frame. Magenta markers correspond to tracking using CACTuS-FL and classification using the S-CNN, and in this case 112 SEFs operate in each frame, with the vast majority tracking background clutter objects.

tracking and classification, NMOTDA is made progressively worse by first classifying using the S-CNN, by next adding position jitter in its input images patches, and finally by also adding position jitter to the bounding boxes. Aside from object tracking accuracy, a second key aspect is that none of these four simulated tests incorporate clutter-tracking SEFs, which would provide additional false positives. CACTuS-FL and the S-CNN have the lowest score in Figure 9 for this very reason: operating 112 SEFs across the scene means that the vast majority of SEFs track clutter sources. The S-CNN on its own, with a typical Clutter class accuracy of ∼ 97.6% (see Table II), would then yield ∼ 2.5 false positives per frame and thus reduce the NMOTDA score.

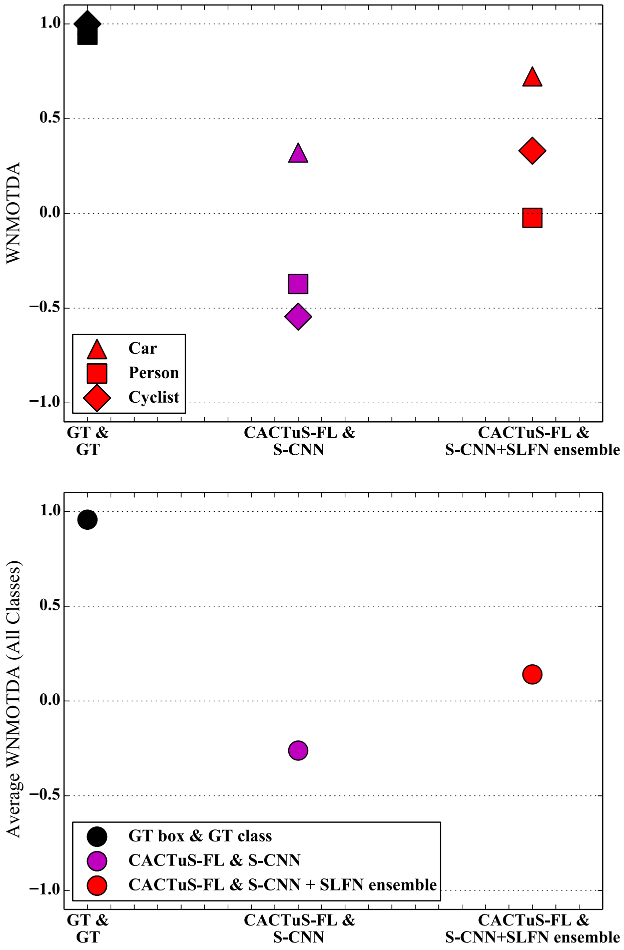

Fig. 10. Neovision2 Tower test WNMOTDA scores, computed across 12 videos. Black markers indicate perfect tracking and classification, magenta markers indicate results from CACTuS-FL & S-CNN, and red markers indicate results from CACTuS-FL & S-CNN + SLFN ensemble. The top plot shows the individual class WNMOTDA, while the bottom plot shows the Average WNMOTDA for all classes.

This inherent challenge posed by *tracking everything* motivates the need for a SLFN ensemble. The Tower test results in Figure 10 illustrate this point, where the inclusion of the SLFN ensemble greatly improves both the overall and class-wise performance. The marked improvement is due to a large reduction in false positives while the number of false negatives tends to remain about the same. Together, the S-CNN and SLFN ensemble fulfil the dual roles of (1) object detection: rejecting Clutter objects while retaining target objects, and (2) object recognition: correctly classifying the target objects (Car, Person, Cyclist), as illustrated by Figure 8.

Table III compares the total numbers of detections, false negatives, and true positives with the total numbers of ground truth objects in our Tower test set of 12 videos. This indicates, for instance, that when considering all objects classes together, the total Recall is ∼ 41%, while the number of false positives per frame is ∼ 2.09. In Figure 11 we compare WNMOTDA with data points that we have extracted from the figures in [28]. Here Teams A, B and C rely on Neuromorphic Vision algorithms, whereas those denoted as Baseline are the results of a computer vision algorithm. Our system is competitive with the state-of-the-art [30] (Team A) in terms of the detection score (Average WNMOTDA), which demonstrates the efficacy of our track everything approach. We also achieve the top scores for Cars and Cyclists, although it should be noted that this is on a reduced 12 video test set.

TABLE III

TOWER TEST SET RESULTS ACROSS ALL 10452 FRAMES: GROUND TRUTHS, DETECTIONS, FALSE NEGATIVES, FALSE POSITIVES

	$\sum_t \mathrm{GT}_t$	$\sum_t \mathrm{Det}_t$	$\sum_t \mathrm{FN}_t$	$\sum_t \mathrm{FP}_t$
All	82139	55271	48746	21878
Car	10452	11499	923	1970
Person	61699	35730	44558	18589
Cyclist	9988	8085	4294	2391

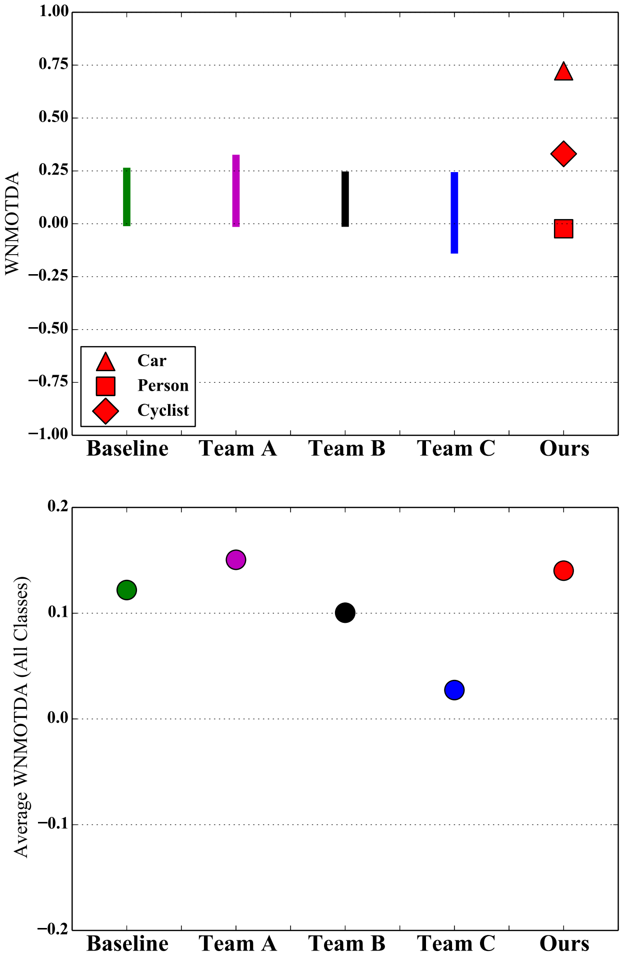

Fig. 11. WNMOTDA published by other teams [28] (Baseline and Teams A−C) and WNMOTDA obtained with our system (CACTuS-FL & S-CNN + SLFN ensemble) across 12 Neovision2 Tower test sequences (in red). The top plot shows our WNMOTDA scores for individual classes (red symbols), while the vertical coloured bands indicate the approximate range of individual class scores obtained by the competing teams. The bottom plot shows the WNMOTDA detection scores, which are obtained by treating all objects (Car, Person, Cyclist) as a single class. Our approach achieves state-of-the-art performance for Car and Cyclist, and comparable performance for Person. The NMOTDA score for Person is reduced in cases when a single SEF tracks a group of people walking together, see text for details.

F. Discussion

1) Prior Knowledge: We have shown that accurate online object recognition can be implemented by using a general-purpose multi-object tracking system that is able to detect and track all salient objects. For this to work, the use of object specific knowledge should be avoided. We have identified in Table I five sources of domain-specific prior knowledge used by CACTuS-FL: the size of convolutional filters, the SEF shape size, the total number of SEFs, the scale of the second order moment ellipse used to define bounding boxes, and the image patch size. However, none of these parameters were tuned for specific object classes, and therefore do not constitute object specific prior knowledge.

Team A [30] achieved state-of-the-art performance using an approach similar to ours, where salient objects are detected and prior knowledge is mostly embedded into the object classifier. The saliency mechanism consists of fusing multiple saliency channels that are created from several individual feature response maps. However, prior knowledge is embedded into some of these saliency channels using the Targeted Contrast Enhancement (TCE) algorithm to create feature response maps that allow them to "easily detect objects with [specified] colors, … e.g. finding all red cars on the road." Another point of difference is that Team A do not perform tracking, only detection and classification. Instead they embed motion processing as another saliency channel, which detects pixels that appear to be moving in comparison to a (stationary or registered) background scene.

The primary difference between our approach and traditional tracking-by-detection approaches is that prior knowledge of the objects of interest is removed from detection and tracking, and only used for recognition.

2) Advantages: The advantage is that all objects are tracked and 'explained away', including sources of clutter. This handling of distracting and occluding clutter improves tracking robustness [10]. For instance, when a person (target) walks behind a lamppost (clutter), the SEF tracking the lamppost learns that it is not moving and the competitive attentional mechanisms in CACTuS-FL allow the SEF tracking the person to ignore the observations from the lamppost.

3) Limitations: One limitation in our current approach is that the tracking system does not know what the extent of a single object is; it simply associates a consistent set of observations (in shape, position and velocity) with a single SEF. For example, people walking together in a group (thus having the same position and velocity) can be efficiently described in the state-space of a single SEF, and thus be considered a single object. This occurs in the Neovision2 Tower test data set video 023. Here a single SEF tracks a crowd of people and the classifier labels the track a 'Person'. However, the bounding box of the crowd is larger than the ground-truth box of any individual person, thus failing the spatial overlap requirement $d_{t,j} > T_d$ from Eqn. (27). This results in both one false positive for the SEF tracking the crowd and many false negatives for the individual people within the crowd, and thus a poor NMOTA score of −0.26 for the video (see supplemental material). This video is a key contributor to the low WNMOTA for the Person class in

Figure 11. Furthermore, without this video the overall Average WNMOTA score would be 0.17 rather than its present value of 0.14.

4) Integrating What and Where: In our architecture low level processing is performed with a common set of convolutional filters (see Figure 2), resulting in a shared set of features for the separate *what* and *where* processing streams. The *what* processing stream is performed by the S-CNN, while the *where* processing stream is performed using CACTuS-FL. By parameterizing elements of the CACTuS-FL state information, it is possible to efficiently re-integrate the *what* and *where* processing stream, using the SLFN ensemble. The benefit of the integration is a gain in Average WNMOTDA of 0.4 as shown in Figure 10. This improvement in recognition performance may provide insight into the function of neurons that integrate both the *what* and *where* processing streams in the primate visual cortex [67]. Knowing *where* an object is (tracking) may help recognise *what* an object is (classification).

VIII. Conclusion

We have presented a system for online object recognition that can autonomously locate and recognize multiple types of objects using biologically inspired *what* and *where* processing streams. Our overall approach may be characterized as a shift of the use of object-specific prior knowledge out of the *where* stream and into the *what* stream. This enables the *where* stream, which is implemented as a general purpose multi-object tracking algorithm, to locate every salient object in the scene, including sources of occluding or distracting clutter. Online recognition of localized objects is then handled by re-integration of the *what* and *where* processing streams. This takes the form of a SLFN ensemble that combines object-tracking state information with class label estimate information from the S-CNN to provide robust object recognition outputs, the performance of which is comparable to the state-of-the-art.

References

[1] F. A. Wilson, S. P. Scalaidhe, and P. S. Goldman-Rakic, “Dissociation of object and spatial processing domains in primate prefrontal cortex,” *Science*, vol. 260, no. 5116, pp. 1955–1958, 1993.

[2] A. Torralba and A. A. Efros, “Unbiased look at dataset bias,” in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2011, pp. 1521–1528.

[3] A. S. Razavian, H. Azizpour, J. Sullivan, and S. Carlsson, “CNN features off-the-shelf: An astounding baseline for recognition,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2014, pp. 512–519.

[4] R. Girshick, J. Donahue, T. Darrell, and J. Malik, “Rich feature hierarchies for accurate object detection and semantic segmentation,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2014, pp. 580–587.

[5] D. Comaniciu, V. Ramesh, and P. Meer, “Kernel-based object tracking,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 25, no. 5, pp. 564–577, May 2003.

[6] S. Wong, “Advanced correlation tracking of objects in cluttered imagery,” *Proc. SPIE*, vol. 5810, pp. 158–169, Jun. 2005.

[7] T. Baker and M. Strens, “Representation of uncertainty in spatial target tracking,” in *Proc. 14th Int. Conf. Pattern Recognit.*, vol. 2. Aug. 1998, pp. 1339–1342.

[8] S. Wong and D. Kearney, “Relating image, shape, position, and velocity in visual tracking,” *Proc. SPIE*, vol. 7338, p. 73380B, May 2009.

[9] A. Gatt, S. Wong, and D. Kearney, “Combining online feature selection with adaptive shape estimation,” in *Proc. 25th Int. Conf. Image Vis. Comput. New Zealand (IVCNZ)*, Nov. 2010, pp. 1–8.

[10] S. Wong, A. Gatt, D. Kearney, A. Milton, and V. Stamatescu, “A competitive attentional approach to mitigating model drift in adaptive visual tracking,” in *Proc. 29th Int. Conf. Image Vis. Comput. New Zealand (IVCNZ)*, Nov. 2014, pp. 1–6.

[11] J. Yang, P. A. Vela, Z. Shi, and J. Teizer, “Probabilistic multiple people tracking through complex situations,” in *Proc. Perform. Eval. Tracking Surveill. (PETS)*, 2009, pp. 79–86.

[12] M. D. Breitenstein, F. Reichlin, B. Leibe, E. Koller-Meier, and L. Van Gool, “Online multiperson tracking-by-detection from a single, uncalibrated camera,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 9, pp. 1820–1833, Sep. 2011.

[13] Z. Wu, J. Zhang, and M. Betke, “Online motion agreement tracking,” in *Proc. 24th Brit. Mach. Vis. Conf. (BMVC)*, 2013, p. 7.

[14] S.-H. Bae and K.-J. Yoon, “Robust online multi-object tracking based on tracklet confidence and online discriminative appearance learning,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2014, pp. 1218–1225.

[15] J.-P. Jodoin, G.-A. Bilodeau, and N. Saunier, “Urban tracker: Multiple object tracking in urban mixed traffic,” in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, Mar. 2014, pp. 885–892.

[16] O. Arandjelović, “Contextually learnt detection of unusual motion-based behaviour in crowded public spaces,” in *Computer and Information Sciences II*. London, U.K.: Springer, 2011, pp. 403–410.

[17] J. Schmidhuber, “Deep learning in neural networks: An overview,” *Neural Netw.*, vol. 61, pp. 85–117, Jan. 2015.

[18] G. E. Hinton, S. Osindero, and Y.-W. Teh, “A fast learning algorithm for deep belief nets,” *Neural Comput.*, vol. 18, no. 7, pp. 1527–1554, 2006.

[19] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, “Gradient-based learning applied to document recognition,” *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.

[20] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet classification with deep convolutional neural networks,” in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.

[21] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun, “OverFeat: Integrated recognition, localization and detection using convolutional networks,” presented at the Int. Conf. Learn. Represent. (ICLR), 2014. [Online]. Available: https://sites.google.com/site/representationlearning2014/conference-proceedings

[22] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” presented at the ICLR, 2015.

[23] M. D. McDonnell and T. Vladusich, “Enhanced image classification with a fast-learning shallow convolutional neural network,” in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2015, pp. 1–7.

[24] M. McDonnell, M. D. Tissera, T. Vladusich, A. van Schaik, and J. Tapson, “Fast, simple and accurate handwritten digit classification by training shallow neural network classifiers with the ‘extreme learning machine’ algorithm,” *PLoS ONE*, vol. 10, no. 8, pp. e0134254-1–e0134254-20, 2015.

[25] L. Leal-Taixé, A. Milan, I. Reid, S. Roth, and K. Schindler. (2015). “Motchallenge 2015: Towards a benchmark for multi-target tracking.” [Online]. Available: https://arxiv.org/abs/1504.01942

[26] A. Milan, L. Leal-Taixé, I. Reid, S. Roth, and K. Schindler. (2016). “MOT16: A benchmark for multi-object tracking.” [Online]. Available: https://arxiv.org/abs/1603.00831

[27] (2013). *DARPA Neovision2*, accessed on Jun. 4, 2014. [Online]. Available: http://ilab.usc.edu/neo2/dataset/

[28] R. Kasturi *et al.*, “Performance evaluation of neuromorphic-vision object recognition algorithms,” in *Proc. 22nd Int. Conf. Pattern Recognit. (ICPR)*, 2014, pp. 2401–2406.

[29] D. M. Paiton *et al.*, “Combining multiple visual processing streams for locating and classifying objects in video,” in *Proc. IEEE Southwest Symp. Image Anal. Interpretation (SSIAI)*, Apr. 2012, pp. 49–52.

[30] D. Khosla, Y. Chen, and K. Kim, “A neuromorphic system for video object recognition,” *Frontiers Comput. Neurosci.*, vol. 8, p. 147, Nov. 2014.

[31] Y. Cao, Y. Chen, and D. Khosla, “Spiking deep convolutional neural networks for energy-efficient object recognition,” *Int. J. Comput. Vis.*, vol. 113, no. 1, pp. 54–66, 2015.

[32] C. Posch, T. Serrano-Gotarredona, B. Linares-Barranco, and T. Delbruck, “Retinomorphic event-based vision sensors: Bioinspired cameras with spiking output,” *Proc. IEEE*, vol. 102, no. 10, pp. 1470–1484, Oct. 2014.

[33] P. A. Merolla *et al.*, “A million spiking-neuron integrated circuit with a scalable communication network and interface,” *Science*, vol. 345, no. 6197, pp. 668–673, Aug. 2014.

[34] M. Danelljan, G. Häger, F. S. Khan, and M. Felsberg, "Convolutional features for correlation filter based visual tracking," in *Proc. IEEE Int. Conf. Comput. Vis. Workshop (ICCV)*, Dec. 2015, pp. 621–629.

[35] C. Ma, J.-B. Huang, X. Yang, and M.-H. Yang, "Hierarchical convolutional features for visual tracking," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2015, pp. 3074–3082.

[36] Z. Yin and R. Collins, "Moving object localization in thermal imagery by forward-backward MHI," in *Proc. Comput. Vis. Pattern Recognit. Workshop*, 2006, p. 133.

[37] R. Martin and O. Arandjelović, "Multiple-object tracking in cluttered and crowded public spaces," in *Proc. Int. Symp. Vis. Comput.*, 2010, pp. 89–98.

[38] M. Cimpoi, S. Maji, and A. Vedaldi, "Deep filter banks for texture recognition and segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 3828–3836.

[39] M. Varma and A. Zisserman, "Texture classification: Are filter banks necessary?" in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, vol. 2. Jun. 2003, pp. 691–698.

[40] V. Stamatescu, S. Wong, M. D. McDonnell, and D. Kearney, "Learned filters for object detection in multi-object visual tracking," *Proc. SPIE*, vol. 9844, p. 98440F, May 2016.

[41] H. Lee, R. Grosse, R. Ranganath, and A. Y. Ng, "Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations," in *Proc. 26th Annu. Int. Conf. Mach. Learn. (ICML)*, 2009, pp. 609–616.

[42] B. A. Olshausen and D. J. Field, "Sparse coding with an overcomplete basis set: A strategy employed by V1?" *Vis. Res.*, vol. 37, no. 23, pp. 3311–3325, 1997.

[43] R. T. Collins, Y. Liu, and M. Leordeanu, "Online selection of discriminative tracking features," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 10, pp. 1631–1643, Oct. 2005.

[44] V. Mahadevan and N. Vasconcelos, "Biologically inspired object tracking using center-surround saliency mechanisms," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 3, pp. 541–554, Mar. 2013.

[45] N. Vasconcelos, "Feature selection by maximum marginal diversity: Optimality and implications for visual recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2003, pp. 762–769.

[46] V. Stamatescu, S. Wong, D. Kearney, I. Lee, and A. Milton, "Mutual information for enhanced feature selection in visual tracking," *Proc. SPIE*, vol. 9476, pp. 947603-1–947603-11, May 2015.

[47] A. Bhattacharyya, "On a measure of divergence between two statistical populations defined by their probability distributions," *Bull. Calcutta Math. Soc.*, vol. 35, no. 1, pp. 99–109, 1943.

[48] M. J. A. Strens and I. N. Gregory, "Tracking in cluttered images," *Image Vis. Comput.*, vol. 21, no. 10, pp. 891–911, 2003.

[49] J.-S. Cho and B.-J. Yun, "Selective-attention correlation measure for precision video tracking," *IEICE-Trans. Inf. Syst.*, vol. E88-D, no. 5, pp. 1041–1049, 2005.

[50] I. Matthews, T. Ishikawa, and S. Baker, "The template update problem," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 6, pp. 810–815, Jun. 2004.

[51] S. Grossberg, "Adaptive pattern classification and universal recoding: I. Parallel development and coding of neural feature detectors," *Biol. Cybern.*, vol. 23, no. 3, pp. 121–134, 1976.

[52] A. M. Hillas, "Cerenkov light images of EAS produced by primary gamma rays and by nuclei," in *Proc. 19th Int. Cosmic Ray Conf.*, vol. 3. 1985, pp. 445–448.

[53] W. Zhu, J. Miao, and L. Qing, "Constrained extreme learning machine: A novel highly discriminative random feedforward neural network," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2014, pp. 800–807.

[54] G. Hinton, O. Vinyals, and J. Dean. (2015). "Distilling the knowledge in a neural network." [Online]. Available: https://arxiv.org/abs/1503.02531

[55] J. Munkres, "Algorithms for the assignment and transportation problems," *J. Soc. Ind. Appl. Math.*, vol. 5, no. 1, pp. 32–38, 1957.

[56] M. Kristan *et al.*, "The visual object tracking VOT2013 challenge results," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV) Workshops*, Jun. 2013, pp. 98–111.

[57] Z. Kalal, K. Mikolajczyk, and J. Matas, "Tracking-learning-detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 7, pp. 1409–1422, Jul. 2012.

[58] L. Cehovin, M. Kristan, and A. Leonardis, "Robust visual tracking using an adaptive coupled-layer visual model," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 4, pp. 941–953, Apr. 2013.

[59] Y. LeCun, C. Cortes, and C. J. C. Burges. (1998). *The MNIST Database of Handwritten Digits*, accessed on Jul. 2015. [Online]. Available: http://yann.lecun.com/exdb/mnist/

[60] Y. LeCun, F. J. Huang, and L. Bottou, "Learning methods for generic object recognition with invariance to pose and lighting," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, vol. 2. Jun. 2004, pp. 97–104.

[61] Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu, and A. Y. Ng. (2011). *NIPS Workshop on Deep Learning and Unsupervised Feature Learning*. [Online]. Available: http://ufldl.stanford.edu/housenumbers

[62] A. Krizhevsky and G. Hinton, "Learning multiple layers of features from tiny images," M.S. thesis, Dept. CS, Univ. Toronto, Toronto, ON, Canada, 2009. [Online]. Available: http://www.cs.toronto.edu/~kriz/cifar.html

[63] S. C. Wong, A. Gatt, V. Stamatescu, and M. D. McDonnell, "Understanding data augmentation for classification: When to warp?" in *Proc. Int. Conf. Digit. Image Comput., Techn. Appl. (DICTA)*, 2016, pp. 1–6.

[64] R. Ekambaram, D. Goldgof, and R. Kasturi. (2012). *Neovision2 Performance Evaluation Protocol*, accessed on Oct. 23, 2015. [Online]. Available: http://ilab.usc.edu/neo2/dataset/neovision2-performance-evaluation-protocol.pdf

[65] K. Bernardin and R. Stiefelhagen, "Evaluating multiple object tracking performance: The CLEAR MOT metrics," *J. Image Video Process.*, vol. 2008, pp. 1–10, Feb. 2008.

[66] A. Andriyenko, K. Schindler, and S. Roth, "Discrete-continuous optimization for multi-target tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2012, pp. 1926–1933.

[67] S. C. Rao, G. Rainer, and E. K. Miller, "Integration of what and where in the primate prefrontal cortex," *Science*, vol. 276, no. 5313, pp. 821–824, 1997.

Sebastien C. Wong (SM'16) received the bachelor's degree (Hons.) in computer systems engineering from Curtin University, the master's degree in electronic systems engineering and the Ph.D. degree in computer science from the University of South Australia, and the Graduate Diploma degree in scientific leadership from the University of Melbourne. He joined the Defence Science and Technology Group, Australia, in 1999, where he currently holds the science team leader image processing position. He is also the Director of Machine Learning with Consilium Technology, and an Adjunct Associate Professor with the University of South Australia. He has authored over 20 peer-reviewed papers. His research interests include machine learning algorithms and parallel processing architectures for autonomous vision systems. He received the Chief of Air force Gold Commendation in 2008 for his work on missile approach warning algorithms and the DSTO Achievement Award in 2012 for his work on Hostile Fire Indication algorithms. He is the past Chair of the IEEE South Australia Section Computer Society Chapter.

Victor Stamatescu (M'14) received the B.Sc. degree (Hons) majoring in physics and the Ph.D. degree in astrophysics from The University of Adelaide in 2004 and 2010, respectively. From 2010 to 2013, he was a Post-Doctoral Research Scientist with the Institute of High Energy Physics, Barcelona, Spain. From 2013 to 2014, he was a Research Fellow with the High Energy Astrophysics Group, The University of Adelaide. Since 2014, he has been a Research Fellow with the School of Information Technology and Mathematical Sciences, University of South Australia. He has authored 79 peer-reviewed journal and conference papers. His current research interests are visual tracking, image classification, and machine learning.

Adam Gatt (M'15) received the bachelor's degree (Hons.) in information technology in 2008, and the Ph.D. in computer science from the University of South Australia in 2013. He is currently a Software Developer with the Australian Defence Force. His research interests involve visual tracking, feature learning, and tracker evaluation. He received the Joyner Scholarship and the UniSA Vice Chancellor and President's Scholarship in 2009. He is a member of the IEEE Computer Society and is the past Chair for the South Australia Chapter.

David Kearney is currently an Associate Professor of Computer Science with the University of South Australia. His research has focused on high performance parallel computing using reconfigurable hardware based on field programmable gate arrays, leading to over 90 refereed publications. His research outputs include the hardware join java language and the ReconfigME operating system for reconfigurable computing. He has interests in applications related to high speed parallel image processing, simulation, and tracking. He has in recent years taken an interest in forms of parallel computing inspired by biology, including membrane computing and neural networks.

Ivan Lee (SM'15) received the B.Eng., M.Com., M.E.R., and Ph.D. degrees from The University of Sydney, Australia. He was a Software Development Engineer with Cisco Systems, a Software Engineer with Remotek Corporation, and an Assistant Professor with Ryerson University. Since 2008, he has been a Senior Lecturer with the University of South Australia. His research interests include multimedia systems, medical imaging, data analytics, and computational economics.

Mark D. McDonnell (SM'11) received the B.E. degree in electronic engineering, the B.Sc. degree (Hons.) in applied mathematics, and the Ph.D. degree in electronic engineering from The University of Adelaide, Australia, in 1998, 2001, and 2006, respectively. In 2007, he joined the University of South Australia, where he is currently an Associate Professor and a Principal Investigator of the Computational Learning Systems Laboratory. He has authored over 100 papers, including several review articles, and a book *Stochastic Resonance*, (Cambridge University Press). He has served as a Guest Editor of the PROCEEDINGS OF THE IEEE and the *Frontiers in Computational Neuroscience*. His research interests lie in the intersection between machine learning and neurobiological learning, with a specific focus on the influence of random noise on learning. His contributions to this area have been recognized by the Award of an Australian Research Fellowship from the Australian Research Council in 2010, and a South Australian Tall Poppy Award for Science in 2008. He has served as the Vice-President and the Secretary of the IEEE South Australia Section Joint Communications and Signal Processing Chapter.