# Online Descriptor Enhancement via Self-Labelling Triplets for Visual Data Association

Yorai Shaoul, Katherine Liu, Kyel Ok, and Nicholas Roy

***Abstract*— We propose a self-supervised method for incrementally refining visual descriptors to improve performance in the task of object-level visual data association. Our method optimizes deep descriptor generators online, by fine-tuning a widely available network pre-trained for image classification. We show that earlier layers in the network outperform later-stage layers for the data association task while also allowing for a 94% reduction in the number of parameters, enabling the online optimization. We show that choosing positive examples separated by large temporal distances and negative examples close in the descriptor space improves the quality of the learned descriptors for the multi-object tracking task. Finally, we demonstrate a MOTA score of 21.25% on the 2D-MOT-2015 dataset using visual information alone, outperforming methods that incorporate motion information.**

## I. Introduction

We are interested in matching visual object detections across temporally separated frames – a fundamental capability for a wide range of applications in robotics and computer vision such as object tracking-by-detection and object-level simultaneous localization and mapping [1], [2].

Although supervised learning methods [3], [4] have recently outperformed hand-engineered descriptors when attempting to adapt robustly to new data-association problems [5], [6], they can be difficult to train, and perform inconsistently. Relying on the existence of massive labeled datasets containing domain-specific training samples, supervised learning of descriptors or affinity metrics may fall short when deployed to novel environments [7]. Contradictory results in the literature also shed light on the inconsistencies plaguing descriptors generated with pre-trained models [8].

In contrast, self-supervised learning methods aim to reduce or eliminate annotation requirements and improve solutions online [7], [9]. To this end, these methods harness the temporal structure of video sequences to collect and annotate positive pairs of image-patches (i.e., subsets of frame pixels) containing the same object in real time, and compile those with negative samples (patches of different objects) into training datasets. The reliance solely on visual information removes the need for relevant annotated data, which may be difficult to obtain for novel environments.

However, due to the computational overhead of online training, one of the challenges of online self-supervised learning for data association is generating training data which are compact but informative. Although the importance of finding informative negative samples is generally acknowledged, many existing approaches rely on simple heuristics such as randomly sampling training examples from candidates [11], [12]. More sophisticated approaches consider image-space properties such as bounding box overlaps [13].

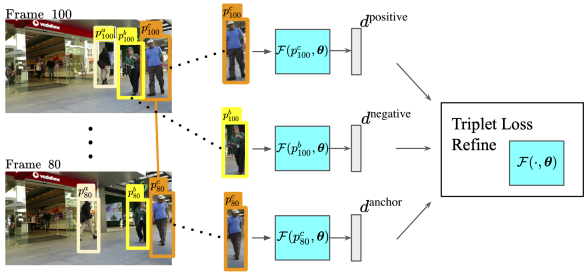


Fig. 1: Our proposed approach self-supervises label generation to incrementally optimize a deep descriptor generator (cyan). To construct a triplet when frame 100 is received, we choose a positive object ($p^c_{100}$), a temporally distant anchor instance of the same object ($p^c_{80}$), and a negative example from the same frame that is closest in the current descriptor space ($p^b_{100}$). When enough patch-triplets are aggregated, they are used to train a descriptor-generator as a batch. The visuals are frames 80 to 100 of the sequence ADL-Rundle-6 included in [10].

In addition, existing approaches tend to still be reliant on pre-trained models, which can limit performance if assumed to be static, or be difficult to obtain if they require offline training. For example, many approaches for tractable online self-supervised visual data association methods have focused on learning lightweight affinity metrics (such as logistic regression) between pairs of patch descriptors [11], [13]–[15]. Without a mechanism for updating the descriptor generation model, such approaches are upper bounded by the representational power of their pre-trained models, and may struggle to extend to novel scenarios. Other methods learn descriptors online, but require labelled detection pairs to pre-train custom descriptor networks before performing online refinement, which may be difficult to obtain for arbitrary new environments [12].

In this work, we propose an online, self-supervised[1] framework for refining deep descriptor models by self-labeling challenging object-triplets in real time. We leverage networks pre-trained for image classification, a task for which training data is abundantly available [16], to provide an initial de-

All authors are with the Computer Science and Artificial Intelligence Laboratory, Massachusetts Institute of Technology in Cambridge, USA. {yorai, katliu,kyelok,nickroy}@mit.edu

This research was sponsored by the MIT Quest for Intelligence and the Army Research Laboratory. It was accomplished under Cooperative Agreement Number W911NF-17-2-0181. Their support is gratefully acknowledged.

[1]To be consistent with prior work [13], we use the term self-supervised in that there is no external labeling process for our data. This kind of learning is more properly termed weakly-supervised learning in that a supervised learner algorithm is used with potentially noisy labels automatically derived from the data.

scriptor space from which to self-supervise the generation of the labels necessary for training the same models to the challenging task of intra-class object disambiguation in novel domains. We call our approach DELTA, for Descriptor Enhancement via Labelling Triplets Attentively.

We exploit the descriptor similarity between detected image patches in consecutive frames, facilitated by their strong visual affinity, to find temporally distant appearances of the same object for positive reinforcement. We further leverage the descriptor space to select difficult negative samples that currently appear to be most similar to the positive example.

We demonstrate the advantages of our method in the context of object tracking-by-detection by evaluating an incrementally refined network through several multiple-object-tracking (MOT) benchmarks. Our online approach learns the descriptors, rather than an affinity metric, and experimentally shows improved tracking performance when trained not only by similar objects separated temporally, but also by negative samples near in the descriptor space. We focus on improving descriptor learning based on visual characteristics alone, observing that our approach can complement methods that consider motion models [11], [17] or global (rather than incremental) information [18]. Our empirical analysis of a convolutional neural network previously trained for image classification enables a 94% reduction in model parameters with an improvement in descriptor performance and makes our algorithm tractable for online optimization. Our method outperforms other approaches that utilize object motion models in terms of multiple object tracking accuracy (MOTA), despite using only incremental visual information.

In the following sections, we formulate the self-supervised online descriptor optimization problem, and discuss our triplet cosine loss as well as the procedure for generating training labels. We describe how our algorithm runs in parallel to a traditional frame-to-frame object tracker, incrementally updating the descriptor generation model. Finally, we report results on the challenging 2D-MOT-2015 tracking dataset, and show that we achieve improved MOTA performance despite drastic computational savings and using only visual information.

## II. Data Association and Tracking Problem Overview

We are interested in the tracking-by-detection problem in dynamic video sequences. There, each observed frame includes bounding-box detections of objects (such as vehicles, pedestrians, cyclists, etc., as illustrated in Fig. 1). Some bounding boxes may also be erroneous detections of the background. We would like to associate each object detection to a previously tracked object, or create new tracks if prior tracked objects are not available. In our framework, we focus on incrementally refining the data-association component of a simple object tracker. Sub-Sections II-A and II-B formulate the data-association and object tracking problems. Finally, Section III details our self-supervised approach for incrementally improving data-association performance online.

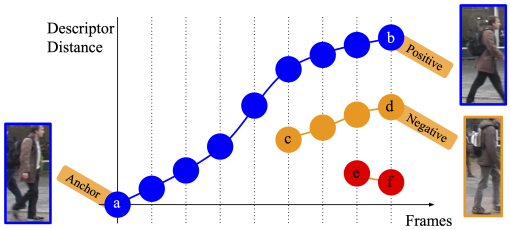


Fig. 2: Illustration of triplet selection, where tracked objects are distinguished by color and similarity to anchor ($a$) by vertical distance. To build a challenging triplet with a positive sample at $b$ for the blue object, we choose a negative example that is near in the descriptor space ($d$) and an anchor example that is distant temporally ($a$). By choosing $d$ over $f$ for the negative example we generate a more informative nuanced label, as $d$ and $b$ are relatively close in the descriptor space. Object patches extracted from the sequence TUD-Campus in 2D-MOT-2015 dataset [10].

Our self-supervised descriptor-learning method runs parallel to the object tracker, interacting only via the deep descriptor model. Fig. 3 illustrates this modular separation, which allows any object tracker using visual descriptors to refine these online using our method. We consider a simple online tracking algorithm to generate the online tracking results, leaving more sophisticated methods for future work.

### A. Visual Data Association Problem

Given consecutive image frames of a scene, and bounding-box detections for objects in the frame, we extract the image contents in the boxed sections, i.e., "patches". Let $p_t^i$ denote the patch extracted from bounding box $i$ in the frame seen at time $t$. We declare two temporally separated patches $p_{t-1}^i, p_t^j$ as positive match if these are images of the same object instance. To this end, we embed image patches $p_t^i \in \mathbb{R}^{h_t^i \times w_t^i \times 3}$ (of height and width $h_t^i, w_t^i \in \mathbb{N}$) in a lower dimensional descriptor vector $\mathbf{d}_t^i \in \mathbb{R}^n$.

Descriptor vectors for patches (resized to $h \times w$) are computed by a mapping $\mathcal{F} : \mathbb{R}^{h \times w \times 3} \to \mathbb{R}^n$ parameterized by $\boldsymbol{\theta}$, i.e. patch $p_t^i$ is mapped to descriptor $\mathbf{d}_t^i = \mathcal{F}(p_t^i, \boldsymbol{\theta})$. We define the similarity between object patches to be the distance between the descriptors associated with them, as computed via a distance metric. The similarity between $\mathbf{d}_{t-1}^i$ and another descriptor $\mathbf{d}_t^j$ is given by a distance function $\mathcal{D}(\mathbf{d}_{t-1}^i, \mathbf{d}_t^j)$, where $\mathcal{D} : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$. For patches $p_t^i, p_{t-1}^{i'}$ of the same object, and $p_t^j$ of a different object, we would like the distance function and descriptor model to yield

$$\mathcal{D}(\mathbf{d}_{t-1}^i, \mathbf{d}_t^{i'}) < \mathcal{D}(\mathbf{d}_{t-1}^i, \mathbf{d}_t^j). \tag{1}$$

Given an appropriate descriptor space, Equation 1 intuitively allows for discrimination between similar and dissimilar patch pairs through distance values – the smaller the distance between patch descriptors is, the more likely they are to correspond to the same object.

The descriptor-generation function $\mathcal{F}$ must capture the highly complex mapping between the raw pixel data to the descriptor space, allowing $\mathcal{D}$ to produce meaningful distance values to discriminate between similar and dissimilar object patches. It is easy to see that approaches that keep $\mathcal{F}$ fixed,

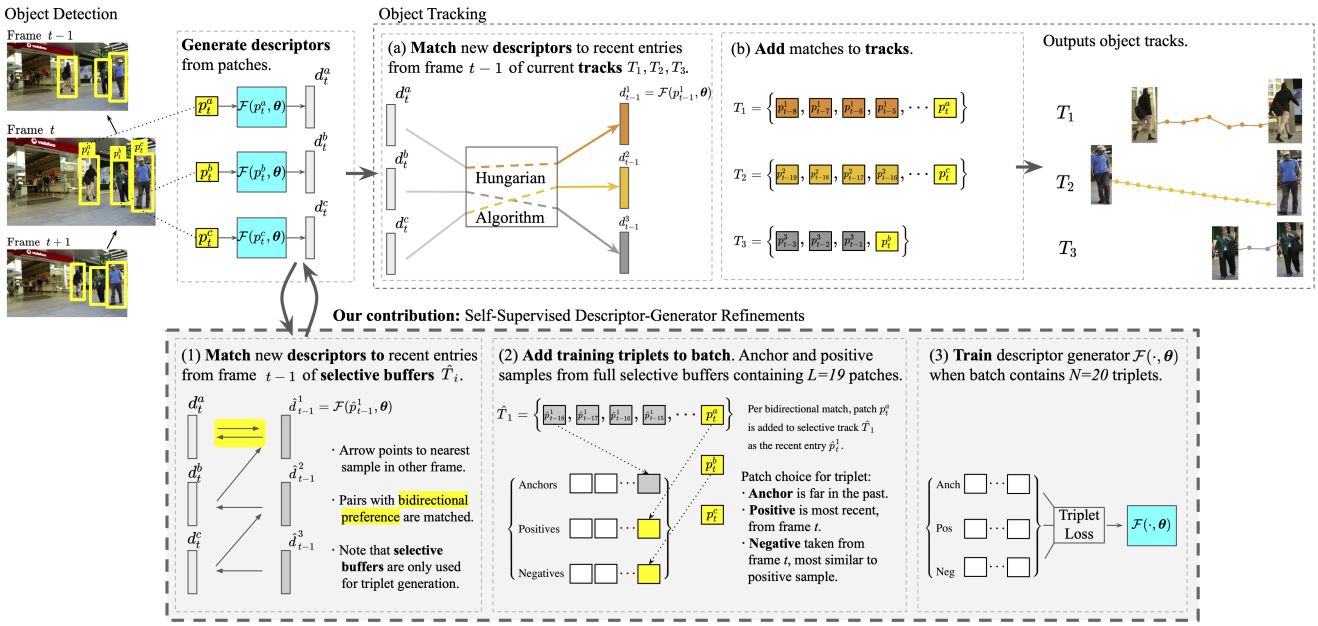


Fig. 3: Our proposed system is composed of a descriptor generator, object tracking module, and self-supervised learning pipeline. The descriptor generator $\mathcal{F}(\cdot, \boldsymbol{\theta})$ converts object measurements to descriptors, and the two processes (tracking and descriptor refinement) run in parallel and interact only through the learned descriptor generator. The self-supervised method keeps selective buffers $\hat{T}_j$, which require that bi-directional preference be satisfied, for the purposes of dataset construction. The object tracker is less particular, and matches all new patches to existing tracks $T_j$ to provide the best possible estimates for all detections.

and learn an affinity metric $\mathcal{D}$, may fall short when $\mathcal{F}$ produces descriptors that cannot be disambiguated under any affinity metric. Therefore, we choose to learn the parameters $\boldsymbol{\theta}$ of the descriptor-generation function.

One method of achieving this complex mapping is by learning the parameters $\boldsymbol{\theta}$ from a labeled training dataset $\mathcal{S}$ where pairs of image patches are labeled as positive (both patches are observations of the same object) or negative pairs (patches of different objects). Given $\mathcal{S}$, a descriptor-generating function $\mathcal{F}$ could be trained to minimize the distance $\mathcal{D}(\mathbf{d}^i, \mathbf{d}^{i'})$ for similar object patches $p^i, p^{i'}$ by using a fixed or learned distance metric $\mathcal{D}$. However, in practice it is difficult to build such datasets. In this work we therefore label relevant training samples in real time.

### B. Object Tracking-By-Detection Problem

For video sequences, where each frame includes noisy bounding box object detections, the object tracking task is to associate each valid bounded image patch, i.e., showing at least part of an object, with a unique identity representing that object. In our framework, we call each unique identifier a "track", and formulate it as a set of object patches $T_i = \{\ldots, p_{t-2}^i, p_{t-1}^i\}$ all of the object $i$ before time $t$.

We attempt to match every detected frame-patch to a track using visual similarity alone, without relying on predictive motion models. To this end, the tracking problem reduces to a data association task – matching observed image patches to existing tracks. Under the assumption that the distance $\mathcal{D}$ is smaller for descriptor pairs of the same object than for different objects, we formulate this problem as an optimization, aiming to choose the least-distance assignment between input patches $p_t^i$ and tracks $T_j$. Let the binary decision variables $x_{i,j}$ take the value 1 when input patch descriptor $\mathbf{d}_t^i = \mathcal{F}(p_t^i, \boldsymbol{\theta})$ matched to the most recent entry of track $T_j$, i.e. $p_{t-1}^j$, and 0 otherwise, our objective is

$$\min \mathcal{D}(\mathbf{d}_t^i, \mathcal{F}(p_{t-1}^j, \boldsymbol{\theta})) \cdot x_{i,j}, \tag{2}$$

under the constraint that as many input patches as possible are matched to tracks. If there exist more detections than tracks, unassigned inputs are each assigned a new tracks. We solve this optimization with the Hungarian algorithm [19]. Our specific implementation details are in Section V.

## III. Online Self-Supervision

In this section we describe our choice of loss function and distance metric for refining descriptors online, detail our method for choosing difficult positive and negative training samples for online training, and discuss our descriptor-generating model. As detailed in Fig. 3, we label patch-triplets in a self-supervised framework and use those to train our descriptor-generating model. Fig. 2 illustrates our use of two sources of information for online dataset construction: time and visual appearance.

### A. Descriptor Refinement with Cosine Triplet Loss

In order to learn a complex mapping $\mathcal{F}$ between object patch pixels to descriptors, a Siamese [21] set of neural networks is often used [11]. Two or more identical descriptor-generators learn to produce different outputs by training on image-patches labeled as similar or dissimilar. Loss functions, such as contrastive loss [22] for training pairs or triplet-loss [14] for training triplets, combine descriptors to a single loss value. Given a descriptor distance metric $\mathcal{D}$, both loss functions aim to minimize the distance between

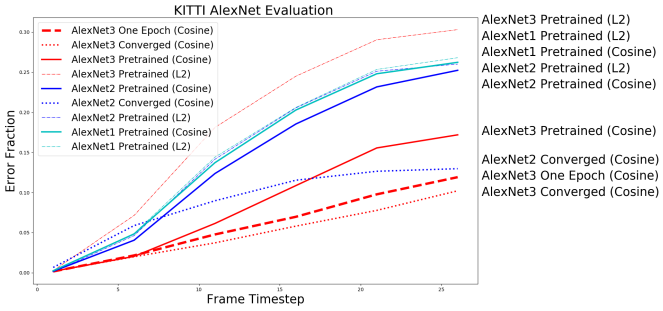


Fig. 4: Supervised descriptor evaluation on KITTI [20] dataset. We evaluate descriptors extracted from the last max-pooling layer within AlexNet (AlexNet3), and the two fully connected layers that follow it (AlexNet2, AlexNet1). Given two frames at time steps $t$ and $t+\Delta$, for all similar patches $p_t^i, p_{t+\Delta}^i$ with dissimilar patches $p_{t+\Delta}^j$, we declare an error if $\mathcal{D}_{\cos}(\mathbf{d}_t^i, \mathbf{d}_{t+\Delta}^j) \leq \mathcal{D}_{\cos}(\mathbf{d}_t^i, \mathbf{d}_{t+\Delta}^i)$. We observe overall worse performance as $\Delta$ grows. "AlexNet3 One Epoch" was trained on the training samples once, as opposed to repeating the training to convergence, and shows quick learning of effective descriptors. Training was performed using ground truth detections in sequences 08-20 divided to 1740 batches of 20 triplets where the positive and anchor samples were 15 and 20 frames apart. The evaluation above was done on the remaining sequences in 80460 comparisons. Our learning rates were $10^{-4}$ for "AlexNet3 One Epoch" and $10^{-10}$ for the converged models.

similar descriptors and to maximize the distance between dissimilar pairs. We choose to use the triplet loss function since it has an inherent balance of positive and negative samples, and found it easier in practice to optimize than a contrastive loss. Additionally, the cosine triplet loss has proven useful in supervised learning contexts [23]. Given $\mathbf{d}_{\text{anchor}}, \mathbf{d}_{\text{positive}}$ for descriptors of the same object and $\mathbf{d}_{\text{negative}}$ for a descriptor of a different object, the triplet loss is

$$L_{\text{triplet}}(\mathbf{d}_{\text{anchor}}, \mathbf{d}_{\text{positive}}, \mathbf{d}_{\text{negative}}) = \max\{0, \mathcal{D}(\mathbf{d}_{\text{anchor}}, \mathbf{d}_{\text{positive}}) + m - \mathcal{D}(\mathbf{d}_{\text{anchor}}, \mathbf{d}_{\text{negative}})\} \tag{3}$$

where $m \in \mathbb{R}$ is a margin parameter marking sufficient dissimilarity between negative pairs.

Although the distance metric $\mathcal{D}$ could be learned [4], we elect to use the fixed cosine distance metric (Equation 4), given that our work focuses on learning descriptors.

$$\mathcal{D}_{\cos}(\mathbf{d}^i, \mathbf{d}^j) = 1 - \frac{\mathbf{d}^i \cdot \mathbf{d}^j}{\|\mathbf{d}^i\| \|\mathbf{d}^j\|}. \tag{4}$$

As depicted in Fig. 4, we experimentally verified the results from [23], showing that the cosine distance $\mathcal{D}_{\cos}$ metric outperformed Euclidean distance in the supervised task of frame-to-frame data association.

### B. Positive Sample Collection

To construct our triplets, we seek to find pairs of image patches of the same object. We seek positive pairs that are expected to be difficult to classify as similar, to provide a good training signal to our model, i.e., we would like to find $p_{\text{anchor}}$ and $p_{\text{positive}}$ such that the initial distance $\mathcal{D}(\mathbf{d}_{\text{anchor}}, \mathbf{d}_{\text{positive}})$ is large.

We collect these difficult pairs by choosing temporally distant image patches. We harness the high frame-rate of video sequences to track objects whose appearance does not vary dramatically between consecutive frames – a property keeping their descriptors close in the initial embedding space. As shown in Fig. 3-(1), we create the bounded-length "selective" patch buffers $\hat{T}_j$ independently from the object tracker by performing bidirectional matching between descriptors $\mathbf{d}_t^i$ of new object detections and the descriptors for the recent entries of selective buffers $\hat{T}_j$, namely $\hat{\mathbf{d}}_{t-1}^j = \mathcal{F}(\hat{p}_{t-1}^j, \boldsymbol{\theta})$. Among $Z$ frame detection patches, we consider associating a patch $p_t^i$ to any of the $M$ selective buffers $\hat{T}_j$, making it the buffer's most recent entry $\hat{p}_t^j$, if

$$\textbf{AND} \begin{cases} \underset{m \in \{1,\dots M\}}{\arg\min}\ \mathcal{D}(\mathbf{d}_t^i, \hat{\mathbf{d}}_{t-1}^m) = j \\ \underset{z \in \{1,\dots Z\}}{\arg\min}\ \mathcal{D}(\hat{\mathbf{d}}_{t-1}^j, \mathbf{d}_t^z) = i \end{cases} \tag{5}$$

Unlike the Hungarian algorithm, which matches as many detections to tracks as possible when solving Equation 2, this criterion is designed to be more selective. Buffers not extended are deleted and unassigned detection patches join new empty buffers. When a buffer $\hat{T}_j$ contains $L$ patches thought to be of consecutive observations of the same object, we choose $p_{\text{anchor}}$, $p_{\text{positive}}$ to be the temporally distant pair $\hat{p}_{t-L+1}^j$, $\hat{p}_t^j$. Full buffers accept new entries and discard old ones to maintain a size of $L$. We use these selective buffers, which often include easily tracked objects, to improve the descriptors used to match more challenging objects.

### C. Negative Sample Collection

To increase the utility of our limited training samples, we find negative samples that are difficult to disambiguate from the positive sample using the descriptor distance metric $\mathcal{D}$. We want a $p_{\text{negative}}$ such that $\mathcal{D}(\mathbf{d}_{\text{positive}}, \mathbf{d}_{\text{negative}})$ is currently small, so that we can improve the descriptors to create more discriminative distance between the positive and negative classes. As shown in Fig. 3-(2) we consider patches detected in frame $t$, the frame from which the positive sample is drawn from, as possible negative sample candidates. We choose the detection patch $p_t^i$ whose descriptor $\mathcal{F}(p_t^i, \boldsymbol{\theta})$ is most similar to the positive patch descriptor $\mathbf{d}_{\text{positive}}$,

$$p_{\text{negative}} = \underset{p_t^i \in \text{frame } t}{\arg\min}\ \mathcal{D}(\mathbf{d}_{\text{positive}}, \mathcal{F}(p_t^i, \boldsymbol{\theta})). \tag{6}$$

### D. Online Optimization

As shown in Fig. 3, we generate training triplets in a process parallel to online tracking. After accumulating $N$ training examples, we train the descriptor model $\mathcal{F}(\cdot, \boldsymbol{\theta})$ for a single epoch, optimizing Equation 7 by substituting in Equations 3, 4, i.e.,

$$\underset{\boldsymbol{\theta}}{\arg\min}\ \frac{1}{N} \sum_i^N L_{triplet}(\mathcal{F}(p_{\text{anchor}}^i, \boldsymbol{\theta}), \mathcal{F}(p_{\text{positive}}^i, \boldsymbol{\theta}), \mathcal{F}(p_{\text{negative}}^i, \boldsymbol{\theta})). \tag{7}$$

We discard each batch after optimization. Our parameter choices are detailed in Section V-A.

## IV. Adaptive Descriptor Generators

CNNs with demonstrated performance in image classification competitions such as [16] are commonly used for image patch matching applications [23], [24]. Although these networks [25], [26] are pre-trained for classification, rather than instance-level disambiguation, previous works have shown that the utility of such networks to generate reasonable descriptor spaces [13]. We propose leveraging the initial suitability of these networks for refining descriptor generation online, and therefore seek network architectures suitable for real-time use and fast adaptation for online refinement. Similar to previous work, we consider the activation values within one of the network's hidden layers as a descriptor for the image patch input to the network, and utilise a distance metric $\mathcal{D}$ to determine similarity between descriptors.

Our experiments in Section V use the output of the last max-pooling layer within AlexNet [25] as our descriptor-generator, and call the subset of AlexNet up to this layer AlexNet3, given that it is missing three fully-connected layers. This network includes only 6% of the parameters originally used in AlexNet, and compared to VGG-16 [26] as it is used in [13] (we name it "VGG-16-2", since it misses two layers), AlexNet3 is smaller by 3136%. We include an evaluation of VGG-16-2 as a descriptor generator in Section V as well.

Fig. 4 details our supervised experimental evaluation of AlexNet in the context of visual data-association across long time steps in videos, and shows that it initially performs about 35% better relatively to other evaluated hidden layers, when pretrained on Imagenet [16] only – without online refinements. These results suggest that earlier layers in this classification network may maintain more detail required to disambiguate intra-class object instances. Additionally, the figure demonstrates AlexNet3 pretrained on Imagenet and tested on KITTI can reduce its error rates by additional 35% after a single supervised training epoch on ground truth bounding boxes from a reserved subset of KITTI [20]. AlexNet3's ability to adapt to new data quickly makes it a favorable choice for a self-supervised descriptor-generator.

## V. Experiments

We evaluated the performance of our proposed method for online descriptor enhancement via self-labelling triplets attentively (DELTA) in the context of tracking-by-detection, as shown in Table I. We evaluate tracking performance in the dynamic video sequences of the popular 2D-MOT-2015 dataset [10], as it provides the temporal structure necessary for constructing dataset triplets online. In the following subsections we discuss the evaluation dataset, the implementation details of our experimental framework, and our performance as compared to baseline methods.

### A. Implementation Details

In our evaluation we used the Hungarian algorithm with the cosine distance $\mathcal{D}_{\cos}$ to match detected patches in frames to existing tracks. We chose AlexNet3 (see Section IV) as our descriptor-generator $\mathcal{F}$, and refined the model online with DELTA. We implemented our approach in Python, with the network implementations provided by PyTorch [28] and the Hungarian algorithm by SciPy [29]. Extracted patches were resized to $227 \times 227$ pixels and normalized to the PyTorch input standards for ImageNet models. All timing metrics were collected on a single-threaded 2.2 GHz Intel i7.

We aggregated batches of 20 triplets for each training cycle, where we used a learning rate of $3.28 \times 10^{-5}$ and margin $m$ of $0.3$. Additional parameters were obtained via a black-box optimization [30] over the *Train* set assuming AlexNet3 as the descriptor-generator. Matches with cost larger than a threshold $0.59$ were discarded. Tracks $T_i$ not extended for more than one frame were de-registered and not extended further. If upon de-registration a track's length was less than 12 frames, it was erased and not used in evaluation. The temporal distance $L$ between anchor and positive samples was 19 frames. All results reported for DELTA, for both on the *Train* and *Test* datasets and both AlexNet3 and VGG-16-2, use the same parameter values.

### B. Evaluation Dataset

We tested DELTA using the 2D-MOT-2015 multiple-object-tracking (MOT) benchmark [10]. This dataset provides a number of video sequences along with bounding-box detections of the objects their frames as generated by the object detector [31]. This challenging benchmark includes videos where the camera is static and dynamic, videos where objects are often occluded, and generally imperfect bounding box detections which include partial detections of objects (i.e., only a leg of a person) or of background.

We quantify the performance in the tracking task, as it is formulated in Section II-B, using the widely accepted CLEAR-MOT metrics [32]. There, the multiple object tracking accuracy (MOTA $\uparrow$) metric quantifies tracking performance comprehensively with the relationship

$$\text{MOTA} = 1 - \frac{\sum_t (\text{FP}_t + \text{FN}_t + \text{IDs}_t)}{\sum_t \text{GT}_t}. \tag{8}$$

At frame t, $\text{FP}_t$ is the number of false positives (patches not showing an object) included in tracks, $\text{FN}_t$ is the number of false negatives (correct detections not included in tracks), $\text{IDs}_t$ is the number of times a true object is assigned a different identity, and $\text{GT}_t$ is the number of ground-truth objects. The multiple object tracking precision (MOTP $\uparrow$) metric is also included, and represents the spatial misalignment of reported tracks and true tracks.

The dataset was divided to *Train* and *Test* sets. We ran our proposed incremental approach on each sequence in the dataset, resetting the network to its pre-trained state after the sequence terminated and before the next was evaluated. The *Train* set can be evaluated locally and was used for choosing values via black-box optimization for the fixed tracker parameters specified in subsection V-A. We did not use ground-truth bounding boxes for any part of our method.

### C. Evaluation Methods

We primarily compare DELTA to other works where visual data association is evaluated by tracking performance.

| **Set** | Sequence | MOTA↑ | MOTP ↑ | IDs↓ | FP↓ | FN↓ |
|---|---|---|---|---|---|---|
| *Train* set results AlexNet3 | DELTA + AlexNet3 (**ours**) | 27.8% | 71.2% | 480 | 6097 | 22228 |
| | DELTA Easy-Positives | 20.5% | 71.9% | 843 | 10349 | 20533 |
| | DELTA Random-Negatives | 26.9% | 72.1% | 516 | 6507 | 22146 |
| | AlexNet3 Pretrained | 20.5% | 71.9% | 843 | 10349 | 20533 |
| *Train set* results VGG-16-2 | VGG-16-2 Pretrained | 19.0% | 71.9% | 1032 | 10974 | 20337 |
| | DELTA + VGG-16-2 | 26.1% | 72.2% | 551 | 6932 | 22008 |
| *Test* set results for strictly visual methods | DELTA + AlexNet3 (**ours**) | **21.25%** | 70.95% | **1231** | **8597** | **38557** |
| | ALEX-TRAC [13] | 17.0% | **71.2%** | 1859 | 9233 | 39933 |
| Additional *Test* set results | TC_SIAMESE [11] | 20.2% | 71.1% | 294 | 6127 | 42596 |
| | TBD [27] | 15.9% | 70.9% | 1939 | 14943 | 34777 |
| | TC_ODAL [12] | 15.1% | 70.5% | 637 | 12970 | 38538 |
| | LDCT [15] | 4.7% | 71.7% | 12348 | 14066 | 32156 |

TABLE I: 2D-MOT-2015 Results. When compared to other trackers emphasizing data-association, and in particular to ALEX_TRAC as it is concerned with strictly visual data association, we achieve better multiple object tracking accuracy (MOTA), a metric which comprehensively combines several statistics as shown in Equation 8. We report the overall number of false-positives (FP), false-negatives(FN), and identity-switches(IDs), and list the multiple object tracking precision (MOTP ↑) metric. We write ↑ next to metrics where larger value is better, and ↓ where smaller value is preferred.

Most similar to our work is the self-supervised tracker ALEX_TRAC [13], where only visual information is used to train an affinity metric online. Similar to our method, they also generate descriptors with a subset of a CNN pretrained on Imagenet [16] (i.e., VGG-16-2) and use the Hungarian algorithm as the main data association module for object tracking. The most prominent difference between the two methods is DELTA learns to refine our descriptor-generating network online while ALEX_TRAC learns an affinity metric.

We also include results for trackers that make use of motion models in addition to visual matching. TC_ODAL [12] trains a descriptor-generator offline and refines it online, and TC_SIAMESE [11] learns a distance metric offline. Both require video datasets with relevant objects and ground-truth detections for training. The self-supervised method LDCT [15] learns tracker parameters in a Latent Structual SVM framework. Finally, we also include the general tracking method TBD [27], which utilizes the Hungarian algorithm.

To assess the efficacy of our proposed method for generating difficult positive and negative examples, we evaluated DELTA with one frame-step between anchor and positive samples (denoted "DELTA Easy-Positives") – making the collected positive samples more visually similar to the anchors. We tested a version of our algorithm where negative samples are randomly chosen "DELTA Random-Negatives". Finally, we applied DELTA to refine a different descriptor generator VGG-16-2 ("DELTA + VGG-16-2"), and included the performance achieved with Imagenet pre-training alone, without online optimization, at "AlexNet3 Pretrained" and "VGG-16-2 Pretrained", for reference.

### *D. Experimental Results*

Among all evaluated methods, DELTA with AlexNet3 achieves the highest *Test* set MOTA score of $21.25\%$, despite not making use of motion models or any spatial information. We outperform ALEX_TRAC despite using a substantially smaller network, showing that our approach effectively leverages data collected online to optimize descriptors.

We show that finding hard positive and hard negative examples is useful for optimizing descriptor performance. Table I shows that choosing positive samples from shorter selective buffers ("DELTA Easy-Positives") led to low loss values during the online training process which had practically no effect on the network, achieving a similar score to that achieved without online learning ("AlexNet3 Pretrained"). In addition, allowing negative samples to be randomly chosen ("DELTA Random-Negatives") also results in a degradation in performance, confirming the utility of selecting negative examples that are near in the descriptor space to the anchor.

We also demonstrate the applicability of our online training procedure and network selection; we show that DELTA improves MOTA scores for both "DELTA + AlexNet3 (**ours**)" and "DELTA + VGG-16-2" from their respective pre-trained performances ("AlexNet3 Pretrained", "VGG-16-2 Pretrained"). However, "DELTA + VGG-16-2" averages a processing rate of 0.1 frames per second, while our analysis in Section IV enables our approach to process 5.5 frames per second on average, further emphasizing the effectiveness of DELTA in refining small descriptor generators.

We note that the MOTP score, which quantifies spatial deviation of object tracks from their true positions, of all evaluated methods is relatively similar, within a margin of $0.7\%$. The consistency may arise because all methods make use of the same bounding box detections provided by the dataset. Additionally, Table I shows that methods relying on motion models can achieve better IDs scores – an expected result given that motion predictions are more robust to occlusions – shedding light on the potential our method has for solving the visual data association task in trackers leveraging motion models in future work.

## VI. Conclusion

We have presented a novel method for incrementally refining object-descriptor-generators online without the need of labeled data or prior domain knowledge. We have shown that refining descriptors online can help improve visual data association performance, and demonstrated that our approach can be applied to object tracking. By achieving object tracking accuracy better than existing methods, we believe that our self-supervised method can be used to solve the visual data association in object trackers that make use of sophisticated motion models.

## References


[1] S. Thrun and W.-K. Yeap, "Simultaneous localization and mapping," in *Robotics and Cognitive Approaches to Spatial Mapping*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 13–41, ISBN: 978-3-540-75388-9. DOI: 10.1007/978-3-540-75388-9_3. [Online]. Available: https://doi.org/10.1007/978-3-540-75388-9_3.

[2] K. Ok, K. Liu, K. Frey, J. P. How, and N. Roy, "Robust object-based slam for high-speed autonomous navigation," in *2019 International Conference on Robotics and Automation (ICRA)*, 2019, pp. 669–675.

[3] A. Holliday and G. Dudek, "Long-distance loop closure using general object landmarks," *CoRR*, vol. abs/1710.10466, 2017. arXiv: 1710.10466. [Online]. Available: http://arxiv.org/abs/1710.10466.

[4] S. Zagoruyko and N. Komodakis, "Learning to compare image patches via convolutional neural networks," *CoRR*, vol. abs/1504.03641, 2015. arXiv: 1504.03641. [Online]. Available: http://arxiv.org/abs/1504.03641.

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

[6] H. Bay, T. Tuytelaars, and L. Van Gool, "Surf: Speeded up robust features," in *Computer Vision – ECCV 2006*, A. Leonardis, H. Bischof, and A. Pinz, Eds., Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 404–417, ISBN: 978-3-540-33833-8.

[7] P. Fischer, A. Dosovitskiy, and T. Brox, "Descriptor matching with convolutional neural networks: A comparison to sift," *arXiv preprint arXiv:1405.5769*, 2014.

[8] V. Balntas, K. Lenc, A. Vedaldi, and K. Mikolajczyk, "Hpatches: A benchmark and evaluation of handcrafted and learned local descriptors," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 5173–5182.

[9] X. Wang and A. Gupta, "Unsupervised learning of visual representations using videos," *CoRR*, vol. abs/1505.00687, 2015. arXiv: 1505.00687. [Online]. Available: http://arxiv.org/abs/1505.00687.

[10] L. Leal-Taixé, A. Milan, I. D. Reid, S. Roth, and K. Schindler, "Motchallenge 2015: Towards a benchmark for multi-target tracking," *CoRR*, vol. abs/1504.01942, 2015. arXiv: 1504.01942. [Online]. Available: http://arxiv.org/abs/1504.01942.

[11] Y. Yoon, Y. Song, K. Yoon, and M. Jeon, "Online multi-object tracking using selective deep appearance matching," in *2018 IEEE International Conference on Consumer Electronics - Asia (ICCE-Asia)*, 2018, pp. 206–212.

[12] S.-H. Bae and K.-J. Yoon, "Robust online multi-object tracking based on tracklet confidence and online discriminative appearance learning," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 1218–1225.

[13] A. Bewley, L. Ott, F. Ramos, and B. Upcroft, "Alextrac: Affinity learning by exploring temporal reinforcement within association chains," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, 2016, pp. 2212–2218.

[14] K. Q. Weinberger and L. K. Saul, "Distance metric learning for large margin nearest neighbor classification.," *Journal of Machine Learning Research*, vol. 10, no. 2, 2009.

[15] F. Solera, S. Calderara, and R. Cucchiara, "Learning to divide and conquer for online multi-target tracking," *CoRR*, vol. abs/1509.03956, 2015. arXiv: 1509.03956. [Online]. Available: http://arxiv.org/abs/1509.03956.

[16] J. Deng, W. Dong, R. Socher, L. Li, Kai Li, and Li Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, 2009, pp. 248–255.

[17] A. Milan, S. Roth, and K. Schindler, "Continuous energy minimization for multitarget tracking," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 1, pp. 58–72, 2014.

[18] B. Yang and R. Nevatia, "An online learned crf model for multi-target tracking," in *2012 IEEE Conference on Computer Vision and Pattern Recognition*, 2012, pp. 2034–2041. DOI: 10.1109/CVPR.2012.6247907.

[19] H. W. Kuhn, "The hungarian method for the assignment problem," *Naval research logistics quarterly*, vol. 2, no. 1-2, pp. 83–97, 1955.

[20] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[21] J. Bromley, I. Guyon, Y. LeCun, E. Säckinger, and R. Shah, "Signature verification using a" siamese" time delay neural network," in *Advances in neural information processing systems*, 1994, pp. 737–744.

[22] S. Chopra, R. Hadsell, and Y. LeCun, "Learning a similarity metric discriminatively, with application to face verification," in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, IEEE, vol. 1, 2005, pp. 539–546.

[23] K Kavitha and B. T. Rao, "Evaluation of distance measures for feature based image registration using alexnet," *arXiv preprint arXiv:1907.12921*, 2019.

[24] S. Yuan, X. Yu, and A. Majid, "Robust face tracking using siamese-vgg with pre-training and fine-tuning," in *2019 4th International Conference on Control and Robotics Engineering (ICCRE)*, 2019, pp. 170–174.

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

[26] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[27] A. Geiger, M. Lauer, C. Wojek, C. Stiller, and R. Urtasun, "3d traffic scene understanding from movable platforms," *IEEE transactions on pattern analysis and machine intelligence*, vol. 36, no. 5, pp. 1012–1025, 2013.

[28] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "Pytorch: An imperative style, high-performance deep learning library," in *Advances in Neural Information Processing Systems 32*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d Alché-Buc, E. Fox, and R. Garnett, Eds., Curran Associates, Inc., 2019, pp. 8024–8035. [Online]. Available: http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf.

[29] P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, S. J. van der Walt, M. Brett, J. Wilson, K. J. Millman, N. Mayorov, A. R. J. Nelson, E. Jones, R. Kern, E. Larson, C. J. Carey, bibinitperiodI. Polat, Y. Feng, E. W. Moore, J. VanderPlas, D. Laxalde, J. Perktold, R. Cimrman, I. Henriksen, E. A. Quintero, C. R. Harris, A. M. Archibald, A. H. Ribeiro, F. Pedregosa, P. van Mulbregt, and SciPy 1.0 Contributors, "SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python," *Nature Methods*, vol. 17, pp. 261–272, 2020. DOI: 10.1038/s41592-019-0686-2.

[30] P. Knysh and Y. Korkolis, "Blackbox: A procedure for parallel optimization of expensive blackbox functions," *CoRR*, vol. abs/1605.00998, 2016. arXiv: 1605.00998. [Online]. Available: http://arxiv.org/abs/1605.00998.

[31] P. Dollár, R. Appel, S. Belongie, and P. Perona, "Fast feature pyramids for object detection," *IEEE transactions on pattern analysis and machine intelligence*, vol. 36, no. 8, pp. 1532–1545, 2014.

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