
Peer reviewed version

Link to published version (if available): 10.1109/IROS45743.2020.9341050

Link to publication record in Explore Bristol Research

PDF-document

This is the author accepted manuscript (AAM). The final published version (version of record) is available online via IEEE at https://ieeexplore.ieee.org/document/9341050 . Please refer to any applicable terms of use of the publisher.

University of Bristol - Explore Bristol Research

General rights

This document is made available in accordance with publisher policies. Please cite only the published version using the reference above. Full terms of use are available: http://www.bristol.ac.uk/red/research-policy/pure/user-guides/ebr-terms/
Centroids Triplet Network and Temporally-Consistent Embeddings for In-Situ Object Recognition

Miguel Lagunes-Fortiz¹,² Dima Damen² and Walterio Mayol-Cuevas²

Abstract—This work proposes learning to recognize objects from a small number of training examples collected and deployed in-situ. That is, from data collected where the objects are commonly placed or being used, perhaps after first encountering them, the learning algorithm immediately is able to recognize them again. We refer to this methodology as in-situ learning, and it opposes to the conventional methodology of using complex data acquisition mechanisms, such as rotating tables or synthetic data, to build a large-scale dataset for training convolutional neural networks (ConvNets). To learn in-situ, we propose a novel loss function that generates discriminative features for known and unseen objects, by utilizing a regularization term that reduces the distance between features and their manifold centroid. Additionally, we propose utilizing a regularization term that reduces the distance between discriminative features for known and unseen objects, by utilizing a temporal filter that is particularly useful to quickly react to appearing objects on the scene, which depending on the distance between neighboring video-frame features, it applies a weighted average between the current and the previous frame. Our framework achieves state-of-the-art accuracy for in-situ and on-the-fly learning, for the case of known objects achieves an average increase in accuracy of 3.01%, an increase of 3.3% for novel objects, and an average increase of 7.07% for the combined case, compared with the closest baseline. Utilizing the temporal filtering, led to a further increase in accuracy against nuisances of 7.32% for the known and novels objects case.

I. INTRODUCTION

As robots are used for performing tasks under unstructured and dynamic environments (Fig. 1), there is a need for them to have a high level of autonomy, this is, without requiring human intervention when they are performing their duties and when they face a novel problem to address at hand.

To increase the level of autonomy in robots, we propose an approach that gives them the capability of learning to recognize specific objects (instances) without the need of recurring to complex image-acquisition systems or generating synthetic examples. Instead, they utilize a small amount of training examples, i.e. low hundreds of training examples per instance, as contained in a few seconds of video data. Since our algorithms are meant to be used by robots and other autonomous platforms, we achieve real-time rates during deployment, e.g. in the magnitude of dozens of frames per second, and without requiring an enormous amount of computational resources such as multiple GPUs.

We aim for a scalable real-time recognition system that can process in-situ data, as we argue that is a more straightforward approach of data collection compared to using complex data-acquisition setups, such as rotating tables, for obtaining pictures depicting them with ideal imaging conditions (in-vitro images). Additionally, we aim for a robust recognition system that can deal with the commonly encountered changes in illumination, perspective, scale, backgrounds, and occlusions.

The main challenges associated with in-situ learning includes recognizing novel objects efficiently, considering the constrained computational resources on-board for most robotic platforms, as well as the unavoidable domain shift, present when training a model with data collected in one environment and deploying in unseen conditions.

Our strategy to address these challenges consists of designing a discriminative model that can associate images depicting the same object, even when they belong to different environments. To do so, we propose the Centroids Triplet Network (CTN) for minimizing the distance between embeddings from the same objects, but also minimizing the distance with respect to their centroids (prototypes), while maximizing the separation to the closest centroid from another object’s manifold. At inference, we propose utilizing the nearest centroids algorithm, to keep scalability and real-time performance at deployment. To deal with nuisances and ambiguous viewpoints, we propose the use of a temporal filter that enforces the temporal coherence that exists in video-data, which prevents sudden changes in the predictions between neighboring video-frames.

We evaluate our approach on datasets tailored for in-situ object recognition and compare it against state-of-the-art methods for learning discriminative features and generating
stable predictions in ConvNets.

II. LITERATURE REVIEW

In-situ recognition of objects, understanding this as objects depicted within their natural or common environments, as opposed to using in-vitro pictures, was first addressed by [1], [2]. In [1], the authors focus on achieving domain generalization from in-vitro to in-situ data, while [2] combines features from in-vitro datasets with features extracted in-situ to build a large-scale real-time recognition system. As limitations, [1] requires in-vitro examples to extract clean descriptors and concluding that collecting in-situ data for training would be an impractical practice. On the other hand, [2] requires that the class of the desired instance to be learned is present in the ImageNet dataset to build a robust classifier.

More recently, the Amazon Robotics Challenge 2017 presented the new requirement of learning novel objects efficiently by providing a set of unseen objects two hours before the competition. Since Amazon provided in-vitro images from such objects, winning teams [3], [4] proposed metric learning techniques to achieve domain adaptation between the images captured by the robot in-situ and the provided in-vitro images.

While [3], [4] were designed for domain adaptation between in-vitro and in-situ, they empirically demonstrated that a ConvNet can be used to learn new objects without having to retrain the model by performing the k-nearest neighbors search in the features space, where data points from the same object are close to each other and separated otherwise. [5] builds on the same idea of utilizing a discriminative ConvNet and simplifies the model into a single branch that uses a combination of Softmax and Triplet Loss for achieving state-of-the-art accuracy for learning objects on-the-fly. We follow this research direction of utilizing discriminative networks for learning new objects without the need for retraining, and therefore, we focus this literature review on supervised approaches that learn discriminative features by utilizing regularizers in the loss function.

Regardless of the architecture design, it is now broadly studied that the commonly used combination of Cross-Entropy Loss and the Softmax function in the last fully connected layer, a.k.a. Softmax Loss, does not explicitly optimize the feature embedding to enforce higher similarity for intra-class samples and diversity for inter-class samples [5], [6], [7], [8], [9], [10], [11], [12], [13].

In this regard, the main approach for learning discriminable features consists of combining the Softmax Loss with regularizers that enforce the intra-class clustering and inter-class separation. These regularizes can be divided into Euclidean regularizers and angular-margin loss functions.

State-of-the-art Euclidean regularizers include Center Loss [9], where the ConvNet learns centers and the clustering of data points around those ones; Triplet Center Loss [14], originally proposed for 3D object retrieval, proposes the incorporation of the Triplet Loss with Center Loss in order to enforce inter-class separability of clusters; Similarly, [5] combines the Triplet Loss with the Softmax Loss where the features for each task are separated into two different heads, in order to improve the accuracy for classifying novel objects. In all these approaches, a hyper parameter is used to balance the supervision signals.

On the other hand, CosFace [11], SphereFace [10] and ArcFace [12] posit as the state-of-the-art angular-margin approaches and build on the findings from Large-Margin Softmax (L-Softmax) [8], which proposes a margin in the cosine product between the weights w and features x in the fully connected layer used for classification. As their name suggests, these approaches come from the facial recognition community and consist of angular constraints applied into the cosine version of the Softmax Loss. Additionally, for performing person identification, authors utilize cosine similarity to compare a query feature, against the features in the database.

Producing stable predictions is also the focus of this work. Stability Training [7] and Single-Frame Regularization [13] are the closest approaches to our goal. In Stability Training, authors propose reducing the dissimilarity in the embeddings between an image X and a variant of it with a small perturbation $T(X) = X + \Delta X$, where the perturbation $\Delta X$ is described as per-pixel independent normal distributed noise $\Delta X \sim N(0, \Sigma)$, with $\Sigma = \sigma^2 I$. In [13], where the goal is to achieve consistency for image-to-image translation, the authors propose reducing the Euclidean between embeddings produced by an image X, where an affine transformation $T$ has been applied before and after the translation: $L_{trans-intra} = \| f(T(X)) - T(f(X)) \|_2$.

As studied by [13], methods for enforcing temporal consistency in image processing are mostly based on estimating dense motion, optical flow, or using recurrent neural networks. While these approaches have shown usefulness, they all suffer from one or more of the following problems: 1) Training a ConvNet and a RNN are commonly separated training stages since one requires shuffled examples and the other sequential ones. 2) high complexity and application-specific architectural modifications, 3) a significant increase in computational complexity for training and inference, 4) failure in situations where motion estimation is difficult, such as image regions with occlusion or lack of texture. Since these limitations make the approaches above unsuitable for in-situ learning, we aim to achieve robustness using an external temporal filter that uses the embeddings produced by the ConvNet.

III. PROPOSED METHOD

Our work builds on the findings from [5] for learning to recognize objects on-the-fly. However, to make the scalable real-time recognition system that we are after, we propose the following contributions. First, we replace the common and costly k-nearest neighbors search for inference used by [3], [5] for the normalized nearest centroids algorithm in the features space. Secondly, we propose an additional distance constraint that enforces discriminability between embeddings and their instance centroids, while maximizing the separation against the closest centroid from another object. Third, to
exploit the temporal consistency in the video frames, we propose a temporal filter that takes into account predictions from previous frames as well as the distance between them. We now explain each of these contributions:

A. Learning Discriminative Features on-the-fly

Since we are performing the nearest centroids search during deployment, we first propose a regularization term for minimizing the distance between a given embedding $x_a$, its corresponding instance centroid $\mu_a$ and maximizing its separation to the the closest centroid $\mu_c$ from another object, as shown in the following equation:

$$ L_{Triplet-Centroids} = [d(x_a, \mu_a) - d(x_a, \mu_c)]_+ $$

(1)

This formulation is fundamentally different from the Stability Training [7] and Single-Frame Regularization [13] approaches, where artificial perturbations are obtained by injecting Gaussian noise $X'$, as well as applying affine transformations $T$ into a given image $X$, and then minimizing the euclidean distance between the embeddings. It is also different and more efficient than directly minimizing the loss with respect to their centroids as in Deep Vector Quatization [6], and shown in section IV.

The loss function that we propose to learn discriminative features is the following:

$$ L_{CTN} = L_{Classification} + \alpha \cdot L_{Triplet-Centroids} + \beta \cdot L_{Similarity} $$

(2)

$$ L_{Classification} = L_{Softmax} = - \sum_i y_i \log(q(x_i)) $$

(3)

$$ L_{Triplet-Centroids} = [d(x_a, \mu_a) - d(x_a, \mu_c)]_+ $$

(4)

$$ L_{Similarity} = L_{Triplet} = [d(x_a, x_p) - d(x_a, x_n)]_+ $$

(5)

The model performs image classification utilizing the features $x$ from the dense layer in red in Fig. 2(b). The first regularizer utilizes the features $x_a$ from the anchor images and the distance between its class centroid $\mu_a$, and the closest centroid $\mu_c$ from the other objects; this regularizer enforces discriminability between features and centroids, useful when the features are not distributed as spherical Gaussian due to the limited training data conditions. The second regularizer consists of the Triplet Loss, which utilizes features $x_a, x_p, x_n$ from the denser layer in blue, and enforces discriminability between features from the same and different objects.

Each regularizer is multiplied by a hyper parameter, $\alpha$ and $\beta$ respectively in [2] which controls the contribution of each regularizer and their values are found empirically, as discussed in the ablation studies.

To train our approach efficiently, we compute all the centroids sparsely every number of epochs $n$, as oppose to every mini-batch iteration. Furthermore, we utilize only the closest centroid per example to compare, as oppose to compare each embedding against all other centroids (Fig. 2).

B. Accelerating Inference Time

During deployment, we use the embeddings produced by the dense layer in blue from Fig. 2(b) We first compute the centroids $\hat{\mu}$ by utilizing the labeled examples $(\hat{x}_1, y_1), (\hat{x}_2, y_2), ... (\hat{x}_n, y_n)$. We then sum all the embeddings belonging to the same object $l \in Y$, and then normalize the resulting vector as indicated in the equation:

$$ \hat{\mu}_l = \frac{1}{\|\sum_i (\hat{x}_i)\|} \sum_i (\hat{x}_i) $$

(6)

To estimate the identity $\hat{y}$ from a given embedding $\hat{x}$, of a query image, we selected the identity of the closest centroid $\hat{\mu}$ found:

$$ \hat{y} = \arg\min_{l \in Y} \|\hat{\mu}_l - \hat{x}\| $$

(7)

C. Temporally-Consistent Embeddings

As studied in [7], unstable learners can classify neighboring video-frames inconsistently due to visual perturba-
tions such as noise. In this work, we aim to empirically demonstrate that in these unstable classifiers, there is a direct relation between inconsistent predictions and dissimilar embeddings. Furthermore, by enforcing the similarity between neighboring video-frame embeddings, it is possible to make ConvNets more robust against such nuisances.

To obtain temporally-consistent embeddings, we then propose a temporal filter that performs a weighted average between the current and previous embeddings. The temporal filter is applied if the Euclidean distance $d$ between the current and previous video-frame embeddings is lower than a threshold $\delta$. A big Euclidean distance $d$ suggests a different object on the scene, and therefore, we give priority to the current prediction. We describe the temporal filter in the following equation:

$$\hat{y}_t = \begin{cases} 
\arg\min_{l \in Y} \|\hat{x}_t - \hat{x}_l\|, & \text{if } d \geq \delta \\
(1 - \gamma) \cdot \arg\min_{l \in Y} \|\hat{x}_t - \hat{x}_l\| + \gamma \cdot \hat{y}_{t-1}, & \text{o/w}
\end{cases}
$$

(8)

The hyper parameter $\gamma$ is the weighting factor between the current and previous frame prediction, the values $\gamma$ and $\delta$ are to be found empirically, and $d$ is the Euclidean distance between the current and previous video-frame embeddings.

IV. EXPERIMENTAL RESULTS

A. Datasets

We selected four datasets that depict objects from an egocentric view, as would be seen from a robot’s perspective for most mobile robots and manipulators. Apart from the iCub dataset, where an actual robot took images, we selected datasets that present recognition scenarios that emulate the in-situ learning scenario that we are after.

To test the models capability to learn additional objects on-the-fly, we utilize the methodology proposed by [3], which aims to measure the discriminability of the features generated from seen and unseen objects. To do so, each dataset is divided into novel and known sets, where two-thirds of the objects composed the known set which is used for training the model and the remaining third correspond to the novel set which is used for recognizing new instances without fine-tuning the model, and performing the nearest centroids search in the embeddings space. Each dataset is described as follows:

CORe50 [15]: Originally proposed for continuous learning, this dataset shows 50 objects across eleven environments and allows us to test the generalization capabilities not only to unseen objects’ poses but also to new environments. We utilize the standard testing set composed by scene 3, 7 and 11, and only scene 1 is used for training the model.

ToyBox [16]: It is composed of 360 toys manipulated by a demonstrator. Toybox allowed us to evaluate how well the model scales, by learning 120 novel objects on-the-fly. We utilize the hodgepodge videos for training and the translations and rotations across x,y,z-axis for testing. The testing set also depicts new conditions such as changes in scale, partial views, and occlusions.

**iCub transformations** [17]: This dataset contains 200 household objects shown by a demonstrator to an iCub Robot. We arbitrarily selected the mixed manipulations set, taken with the left camera for training the model and mixed manipulations set but the following day for testing. The testing set depicts additional backgrounds, viewpoints, and scales.

**In-situ household**: Since none of the state-of-the-art datasets allow evaluating the more realistic situation where the training images are collected within the place where such items are commonly used, such as TV remotes in a living room, we propose a new dataset to assist the benchmarking of in-situ learning approaches by depicting each instance in its commonplace. Our dataset consists of 20 objects with deferring training and testing conditions, without hand presence but depicting a variety of viewpoints, scales, clutter, occlusions, and illumination conditions. We plan to add more places and objects in the future.

B. Baselines

To evaluate the effectiveness of our method, we selected state-of-the-art approaches for learning discriminative features, as it is relevant when learning objects on-the-fly, as well as approaches that aim for robust and temporally consistent predictions without the use of recurrent connections. All the backbone ConvNet in the following models consist of ResNet-50, with an additional Dense Layer with a dimension of 1024 elements which is used as an embedding. We initialized all the models with weights learned from ImageNet, and use stochastic gradient descent as the optimizer with a learning rate of $l_r = 1 \times 10^{-3}$ and momentum $\mu = 0.9$. Each model was trained three times, and we chose the checkpoint with the best performance for the combined case of recognizing known and novel objects. We selected the following hyper parameters for each baseline, each of them are explained in their corresponding citation:

**Deep Vector Quantization** [6]: As a first experimental baseline we utilize a loss function that minimizes the Euclidean distance between features and its class manifold centroid, given by $\mathcal{L} = \|x - \mu\|_2$.

**Stability Training** [7]: We utilize $\alpha = 0.01$, and for generating random noise, we use a standard deviation of $\sigma = 0.04$.

**Invariance Regularization** [13]: We utilize $\alpha = 0.95$ and the affine transformations described in [13]. We use the same affine transformations as data augmentation for the other models, in order to make all approaches comparable.

**CenterLoss** [9]: We selected the hyper parameters $\lambda = 0.1$ and $\alpha = 0.005$ in the loss function [9] and utilized the PyTorch implementation from [18].

**S-Triplet** [5]: We selected the hyper parameters $\lambda = 0.0001$ in the loss function [9] and utilized the PyTorch implementation from [19].

Angular-margin approaches: CosFace [11], SphereFace [10] and ArcFace [12] posit as the state-of-the-art angular-margin approaches for learning discriminative features. We
utilize the generalized loss function and implementation presented in ArcFace [12], where the parameters $m_1$, $m_2$, and $m_3$ represent the constraint proposed in CosFace, SphereFace and ArcFace, respectively. As hyper parameters, we selected $m_1 = 0.55$, $m_2 = 0.5$, $m_3 = 4$, and $s = 30$. As in their original implementation, we utilize cosine distance to measure the similarity between embeddings.

**Softmax Loss**: We utilize the most commonly used loss function for classification, consisting of a combination of Cross-Entropy loss with a softmax operation.

**C. Implementation Details**

We initialize the backbone ConvNet from a pre-trained model with Imagenet. Thus, our model works with RGB images with size $224 \times 224$. For training our model, we use mini-batches of 64 images, we use stochastic gradient descent with a learning rate of $l_r = 1 \times 10^{-3}$, momentum $\mu = 0.9$ and weight decay regularization of $\omega d = 1 \times 10^{-4}$. Code with training scripts, links to download our dataset, and details on the instances chosen as known and novel can be found [here](#).

**D. Recognizing Known and Novel Objects on-the-fly**

To get an understanding of the usefulness of our proposed model, we show in Tables I-III the accuracy and standard error for three different situations: (a) Only using known objects, (b) only using novel objects and (c), a general case where there is no assumption about the object to test, and embeddings of known and novel objects are used for estimating the identity of a query object.

Overall, for the case of known objects, our model achieved an average increase in accuracy of 3.01% compared against the closest baseline, the S-Triplet model. For the case of novel objects, it achieved an average increase of 3.3%, and an average increase of 7.07% for the general case.

The temporal filter that uses the embeddings distances was consistently useful for increasing the accuracy for all the approaches, as we show in Table IV with the corresponding $\gamma$ and $\delta$ values used. For our model, there was an average increase in accuracy of 8.33% for the general case of known and novel objects combined. With the model still making mistakes when there are ambiguous viewpoints from the very beginning of a testing sequence, especially confusing novel objects with known ones.

**E. Scalability and Real-Time Inference**

To test scalability, we considered the general case of recognizing known and novel objects (Table III). We compare the accuracy, the wall-clock time taken for evaluating all testing images and the storage required for saving all training embeddings and their instance centroids.

We noticed a consistent decrease of around 2% in accuracy in each dataset of our normalized-nearest centroids algorithm against $k$-nearest neighbors ($k=5$). However, there was a considerable reduction in storage required and inference.
time. For all the datasets, storing only the centroids represented less than 1% of storage required compared to storing all training embeddings as required in k-nearest neighbors. Related to the inference time, the nearest centers algorithm performed two orders of magnitude faster than k-nearest neighbors (with k = 5), making it an overall more suitable approach for a scalable and real-time object recognition system.

F. Hyper parameters Searching

In order to find a suitable value for the controlling α and β hyper parameters in 2, we performed a grid search, ranging values from $1 \times 10^{-1}$ to $1 \times 10^{-5}$ by decreasing an order of magnitude each step for both hyper parameters. We started by finding the best value for α by setting β = 0. With the best overall value of $α = 1 \times 10^{-1}$, we found $β = 1 \times 10^{-2}$ to be the best overall value. In general, α and β with values greater than $1 \times 10^{-1}$ causes a degradation in accuracy. Therefore, we recommend choosing $α = 1 \times 10^{-1}$ and $β = 1 \times 10^{-2}$ as starting points.

Similarly, for the temporal filter we explore the values of α from 0.50 to 0.99 in increments of 0.01 and keeping δ = 3. Once we found the best value, which is around 0.95, we then explore values of delta from 1 to 6 in increments of 0.5. We found values around 4.5 to be the most suitable in our datasets.

G. Ablation Studies

As ablation studies, we first explore similarity metrics for comparing embeddings from neighboring video-frames. We considered Manhattan distance, cosine similarity, and Euclidean distance. To evaluate the usefulness of the additional Triplet-Centroids loss, we compare our proposed model against the S-Triplet, Softmax Loss, and a Supervised Centroids loss. We found that using Euclidean distance resulted in the highest overall accuracy, as shown in Fig. 3(a). We also explored different sizes for the embeddings, as we show in Fig. 3(b). Bigger sizes achieved an overall higher accuracy, with an approximate increase of 2% by doubling the embedding dimension. Noticeably, for the household dataset, there was a decrease in accuracy when using a dimension of 1024, indicating that the regularization offered by our loss function and weight decay are not enough to compensate the lack of training data and the model starts overfitting.

H. Discussion

As shown in Table III using the proposed Centroids Triplet Loss led to an average increase of 7.07% in accuracy, compared to the S-Triplet approach. This increase in accuracy suggests that, with the in-situ datasets used, the features produced by the S-Triplet did not distribute as a Gaussian hyper-sphere around its centroid, meaning that there were features closer to other objects’ centroid than its corresponding one, this was more notorious in the proposed Household and iCub dataset. In this regard, there is still more to be known about the properties of the manifolds generated in the in-situ learning scenario and we leave that as a future research direction. Replacing the k-nearest neighbors search by the nearest centroids algorithm resulted in a highly beneficial approach for the onboard learning capabilities that we are after, since using this algorithm allowed a faster inference time by up-to two orders of magnitude, and a storage space of only 1.8 KB of memory per object, using an embedding size of 1024 elements with numpy 1.18.1 and Python 3.7.1.

With respect to the proposed weighted average used as a temporal filter, while adding the embeddings separation helped to quickly react when new objects appear on the scene and gaining a further average increase of 7% in accuracy, there are cases where using a static threshold δ was not sufficient.

In conclusion, the combination of a classification, image similarity, and centroids-discriminability losses allowed our Centroids Triplet Network to learn generalizable features for in-situ object recognition. The use of the nearest centroids algorithm contributed to maintaining scalability and real-time performance when learning additional objects. Finally, the temporal filter was a helpful strategy for increasing the network robustness for known and novel objects.

ACKNOWLEDGMENTS

We would like to thank the visual egocentric research activity partially funded by UK EPSRC EP/N013964/1. We also want to thank Google for the GCP credits.
REFERENCES


