View metadata, citation and similar papers at core.ac.uk

brought to you by CORE

provided by Repositorio Institucional Universidad de Málaga

A weakly-supervised approach for discovering common objects in airport video surveillance footage*

F.M. Castro[1][0000−0002−7340−4976], Rubén Delgado-Escaño[1][0000−0002−2365−6593], N. Guil[1][0000−0003−3431−6516], and M.J. Marín-Jiménez[2][0000−0001−9294−6714]

[1] University of Málaga, Málaga, Spain
{fcastro, rubende, nguil}@uma.es
[2] University of Córdoba, Córdoba, Spain
mjmarin@uco.es

Abstract. Object detection in video is a relevant task in computer vision. Standard and current detectors are typically trained in a strongly supervised way, what requires a huge amount of labelled data. In contrast, in this paper we focus on object discovery in video sequences by using sets of unlabelled data. Thus, we present an approach based on the use of two region proposal algorithms (a pretrained Region Proposal Network and an Optical Flow Proposal) to produce regions of interest that will be grouped using a clustering algorithm. Therefore, our system does not require the collaboration of a human except for assigning human understandable labels to the discovered clusters. We evaluate our approach in a set of videos recorded at *apron area*, where the aeroplanes park to load passengers and luggage. Our experimental results suggest that the use of an unsupervised approach is valid for automatic object discovery in video sequences, obtaining a CorLoc of 86.8 and a mAP of 0.374 compared to a CorLoc of 70.4 and mAP of 0.683 achieved by a supervised Faster R-CNN trained and tested on the same dataset.

Keywords: Object discovery · Weakly-supervised learning · Region proposal · Deep Neural Networks.

1 Introduction

The goal of *object detection* is to define the spatial extent and the kind of objects present in an image or video sequence. The object detection problem has been studied for a long time from multiple points of view [1, 7, 20, 23]. Traditional approaches are based on manually designed descriptors that are computed, and then classified, along the image in a sliding window setup [7, 23]. With the advent of deep learning techniques, the features are self-learnt by the model, what greatly boosted the performance of the proposed approaches [1, 20].

* Supported by project TIC-1692 (Junta de Andalucía) and TIN2016-80920R (Spanish Ministry of Science and Tech.). We gratefully acknowledge the support of NVIDIA Corporation with the donation of the Titan X Pascal GPU used for this research.

However, all those approaches required to be trained in a strongly supervised way, that is, using manually labelled data for training. This requirement, coupled with the large amount of data needed for training a deep learning approach, makes costly the use of deep learning models with new classes that are not present in public datasets. Therefore, in order to add new classes to a dataset, it is necessary to label thousands of images in order to perform a good training process. Thus, there is a bottleneck that hampers the scaling of the detection models to larger number of classes: the lack of annotated images, as the annotation process is tedious and time-consuming.

The ideal solution to this problem would be to to train the models using the huge amount of unlabelled data available in many online media sharing pages, such as Flickr. However, only few works apply fully unsupervised solutions, and most of them have serious limitations, like having only one object in the scene [3,12] or being focused on a specific kind of object (*e.g.* humans) [29]. Moreover, the unsupervised solutions produce lower results than using a supervised one. Borrowing ideas from supervised and unsupervised learning, we found the weakly-supervised learning that uses unlabelled data combined with either some labelled samples or coarse grained information about some samples. The key idea is to combine the huge amount of unlabelled information available with some labelled data in order to facilitate the training process. In this category of learning approaches we find works that use image-level labels (*i.e.* the label of the visually dominant object) to learn to localise the object [8,25], or, on the other hand, they use some labelled samples like in [4,31].

In contrast to all those previous works, we propose a weakly-supervised approach for object discovery in videos. In Fig. 1 we show a sketch of our pipeline. Firstly, we automatically find regions of candidate objects in a sequence of frames by using the Region Proposal Network (RPN) of a Convolutional Neural Network (CNN) previously trained for object detection. In addition, we compute the optical flow maps between consecutive frames to discover areas with moving objects. The obtained regions are described by a feature vector obtained from a pretrained CNN. After that, the descriptors are grouped using a clustering algorithm in order to find similar objects. Finally, to assign the labels to the detected objects, a human may optionally revise some samples from each cluster. This labelling process will be performed only when new clusters are obtained, what means that there are new classes present in the video. Thus, instead of labelling hundreds of thousands images, it is only necessary to label a set of clusters.

Therefore, the main contributions of this work are: *(i)* a novel pipeline for weakly-supervised object discovery in videos; *(ii)* a combination of two complementary region proposals specially designed for video sequences; *(iii)* a fast weakly-supervised labelling process based on a clustering process; and, *(iv)* a thorough experimental study to validate the proposed framework.

In our experiments, we use videos obtained from a RGB camera that is continuously recording the *apron area* (area where the aeroplanes park to load passengers and luggage) of the Gdansk Airport, although it can be used in other scenarios with static cameras, like bus or train stations, ports, etc. According to

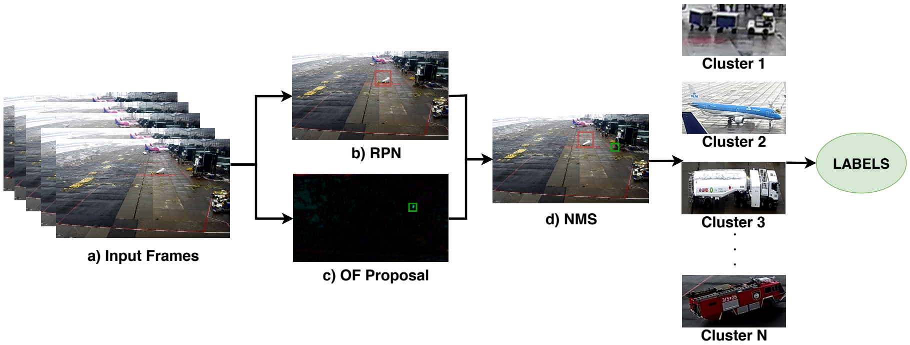

Fig. 1. Pipeline for weakly-supervised object discovery. a) The input is a sequence of RGB video frames. b) The RPN process a set of detections for each input frame. c) The OFRP obtains another set of detections. d) Non-maximum suppression is applied to combine overlapped detections. A final clustering step is performed to obtain a label per detection.

the results, our region proposal approach is more robust than a fully-supervised approach (fine-tuned model for this specific domain), specially with small regions or classes with changes in the shape such as persons. However, during the automatic labelling of discovered regions, our approach obtains worse results than the fully-supervised approach, mainly for classes with few samples.

The rest of the paper is organised as follows. We start by reviewing related work in Sec. 2. Then, Sec. 3 explains the proposed approach. Sec. 4 contains the experiments and results. Finally, we present the conclusions in Sec. 5.

2 Related Work

Weakly-supervised approaches. Weakly-supervised learning has gained importance in the last years [5, 17, 26–28] due to the huge amount of publicly available unlabelled data. By contrast, the amount of labelled data is very reduced, what penalises supervised approaches. Focusing on the problems of object detection/localisation, many researches use the key idea of having an image level label that only contains the object that appears in the image, without bounding-boxes (like in an image classification problem). In [8], the authors propose a multiple-instance learning approach that iteratively trains a detector and infers the bounding-boxes. Kantorov *et al.* [10] use two context aware models to improve the localisation taking into account the surrounding context region of the bounding-box. In [13], Li *et al.* propose the use of an object detector trained only with positive samples of a class to produce a set of heat maps that are refined and segmented to localise the trained class. Wang *et al.* [25] propose a different approach using a spatio-temporal minimisation process for video object discovery and segmentation across videos with irrelevant frames. In contrast to those previous approaches, some authors propose the use of some labelled samples together with a big amount of unlabelled samples. An example of this kind of work is the proposed in [4] where the authors use conditional random fields initialised from generic knowledge and, iteratively, they adapt to new classes.

Shi *et al.* [21] propose a different approach using a Bayesian joint topic modelling that uses a single generative model for all objects improving learning and localisation. A more complex approach is presented in [31] where the authors introduce a deep model that uses a joint learning to localise and segment objects. A completely different approach is presented in [30] where the weak information is provided as a sentence that describes the image.

Unsupervised approaches. Unlike weakly-supervised learning, unsupervised approaches do not use any kind of labelled data. Due to the difficulty of this type of approach, there are very few works that applies unsupervised learning to the object detection/localisation task. In [3,12], the authors propose an approach for dominant object discovery and tracking in videos using region proposals and a matching scheme in order to produce spatio-temporal tubes of detections in videos. A more recent point of view of this work is presented in [24] where the authors reformulate the approach as an optimization process improving previous results by a wide margin. Koh *et al.* [11] propose an approach for primary object discovery in videos exploiting the recurrence of a primary object in a video using a modelling scheme from motion and colour proposals. A specific approach for pedestrian detection in proposed in [29] where an iterative process of object discovery, object enforcement and label propagation is performed for training a progressive latent model for pedestrian detection. Ošep *et al.* [16] propose an automatic approach for object discovery in stereo videos using a generic tracker to find the objects, and a clustering process to group similar tracks. Note that, although this approach is somehow similar to ours, they use stereo cameras together with a tracker, while we use a single camera as input. Moreover, we rely on the optical flow maps to find out moving objects, while they use a pretrained object tracker that could tend to detect only objects that were seen during training. A similar approach is presented in [18] where they also use stereo cameras and depth information to perform object discovering and a clustering process to group similar detections.

In this work, we explore the weakly-supervised object discovery problem in videos obtained from common RGB cameras. Our approach is composed of two main parts: a region proposal step to produce interest areas and a clustering step to group similar areas. Finally, the obtained set of clusters can be manually labelled in order to use human understandable labels. Note that our approach can be applied in a fully unsupervised way using as labels the cluster indices.

3 Proposed Approach

In this section we describe our proposed framework to address the problem of weakly-supervised object discovery in videos using CNNs. The pipeline proposed is represented in Fig. 1. Using a sequence of consecutive frames as input, the following steps are performed: *(i)* region proposal using the RPN branch of a Faster R-CNN [20]; *(ii)* region proposal based on the optical flow maps obtained from consecutive frames; *(iii)* non-maximum suppression to combine and remove overlapped regions from both proposals; *(iv)* clustering process to group similar regions; and, *(v)* manual labelling of the obtained clusters.

3.1 Region Proposal

We describe here the two region proposal strategies used in our approach. In particular, we use the RPN branch of a pretrained Faster R-CNN [20] and the optical flow maps obtained from a pretrained Spatial Pyramid Network (SpyNet) [19]. Our intuition is that the RPN will be able to detect big objects present in the foreground, which are the most common kind of detections used for training this type of networks. Similarly, the region proposal based on the optical flow maps will be able to find out subtle and small objects.

Region Proposal Network. In our approach, we use the Region Proposal Network (RPN) of a pretrained Faster R-CNN [20] model. A RPN uses an image as input to produce a set of object proposals. To generate the regions, the input sample is fed into a set of convolutional layers in order to produce a feature map. After that, a small network is slided over that feature map with a window size of $n \times n$. This small network is composed of two sibling fully-connected layers, a box regression layer (*reg*) to obtain the bounding box coordinates and a box-classification layer (*cls*) to obtain the class of the detected object. Note that the sliding window is implemented in a direct way by using a $n \times n$ convolutional layer followed by two sibling 1×1 convolutional layers for *reg* and *cls*. Note that in this step the pretrained model applies a suppression of detections whose score is lower than a threshold T_S. The regions kept after the filtering process are fed into the next step of our pipeline.

Optical Flow Proposal. The optical flow proposal (OFRP) is based on the optical flow maps obtained from a pretrained Spatial Pyramid Network (SpyNet) [19] model. Two consecutive frames are fed into the network to produce an optical flow map F_t. To remove noise from the optical flow map produced by changes in the illumination or the conditions of the scenario, all positions whose optical flow components (x and y) are smaller than a threshold T_F are set to 0. After that, we binarize the optical flow map and find the contours to obtain the regions of the objects present in the map. In order to do this, we use the well known algorithm proposed by Suzuki *et al.* [22]. To avoid intermittent detections, we track each region in F_t to the next optical flow map F_{t+1} using the mean optical flow components of that region. If the region is missed in F_{t+1}, we remove the original detection from F_t. Finally, in order to prevent insignificant regions, we remove those proposed regions whose area is smaller than a threshold T_A. The regions kept after the filtering process are fed into the next step of our pipeline.

3.2 Non-Maxima Suppression

Since there are two region proposal algorithms running at the same time, when a big object is moving in consecutive frames, it is probable that the RPN proposal and the OFRP produce detections of the same object. Therefore, it is necessary to combine both detections into a single one to avoid overlapping regions. To combine both detections, we compute the Intersection over Union (IoU) metric between both detections. If the IoU is bigger than a threshold T_I and the aspect ratio of both regions is similar (*i.e.* the ratio of the biggest one between the smallest one must be bigger than a threshold T_{AR}), we keep the region proposed by the RPN algorithm, as it is more accurate than the optical flow one. Moreover, we apply non-maxima suppression to each individual proposal algorithm to remove overlapped regions whose IoU is greater than the same threshold T_I. After this step, we have the final set of regions used by the clustering algorithm.

3.3 Clustering

The objective of this step is grouping similar regions into clusters. By this way, instead of learning a classifier that assigns a label according to the features of a region, we just find the closest cluster to a region. In order to do this, we first have to describe the detected regions to fed that information into the clustering algorithm. To describe the regions we employ a pretrained ResNet-50 [9] model as feature extractor, where descriptors are given by the activations of the average pooling before the classification layer. Once the features of each region have been extracted, we apply a L2-normalisation to the features and we reduce their dimensionality to 128 with the UMAP algorithm [15]. This method is specially useful as it is able to reduce the dimensionality keeping the global structure of the data but preserving local neighbours relations. Finally we use them as input to the clustering algorithm. For this purpose, we use the HDBSCAN clustering algorithm [2] which is able to deal with different cluster shapes and densities with a good performance. These capabilities are really important in our problem since there could exist objects with many different number of samples or even objects with many different shapes, and the algorithm should deal with those situations.

3.4 Labelling Process

The last step of our pipeline is completely optional, since it is only necessary in order to assign a human understandable label to each cluster obtained in the previous step. To do this, we show the set of N samples with the highest scores, obtained by HDBSCAN, for each cluster to a human who establishes the labels of the clusters. Note that a higher score indicates a better membership of a sample to a given cluster. In order to assign a label to a cluster, at least, half of the showed cluster samples must belong to one of the considered classes. By this way, the labelling is robust against outliers. Note that this is the only step where the intervention of a human is necessary. Once clusters are labelled, that information can be used in order to remove false positives detections if they are grouped into different clusters. Thus, if a cluster only contains false positives, we can ignore that cluster to produce better detections.

4 Experiments and Results

4.1 Dataset

In our experiments, we are going to use a video dataset obtained from a RGB camera that is continuously recording the *apron area* (area where the aeroplanes park to load passengers and luggage) of the Gdansk Airport. Those cameras are publicly available online [3]. The dataset consists of 96 video-clips of one minute length recorded by a FullHD camera which provides a video stream with

[3] Live cameras: http://www.airport.gdansk.pl/airport/kamery-internetowe

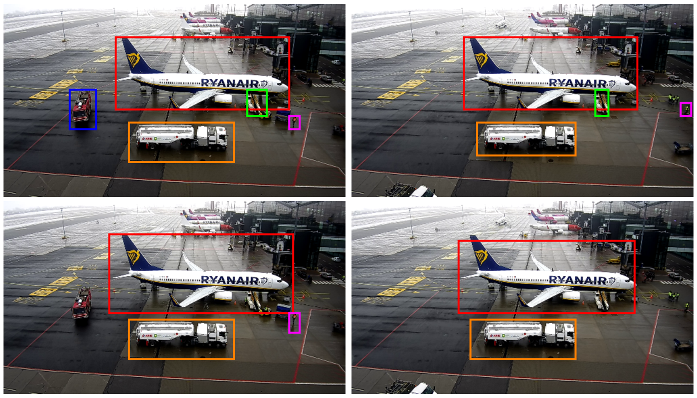

Fig. 2. Dataset. Different frames obtained from a RGB camera recording the *apron area* of the Gdansk Airport. Top row shows the ground-truth labels obtained manually. Bottom row shows the output of our weakly-supervised approach.

a resolution of 1920 × 1080 pixels and a frame rate of approximately 15 fps. Approximately, 60% of the videos are recorded during the morning and the other 40% are recorded during the afternoon/evening in order to deal with different illumination conditions. Some examples can be seen in Fig. 2. Note that in our experiments we are going to focus on the closest apron area since the other areas are excessively far to identify the appearing objects. Thus, in our experiments we are going to consider the following categories of objects: *car ('car'), firetruck ('ft'), fuel-truck ('fuel'), luggagetrain-manual ('lgm'), luggagetrain ('lg'), mobile-belt ('mb'), person ('pe'), plane ('pl'), pushback-truck ('pb'), stairs ('st')* and *van* ('van'). Note that the abbreviated name of each class used in the tables is included in parenthesis. For training the clustering algorithm and our dimensionality reduction, we use odd id video clips and, for testing, the even id video clips, *i.e.* half for training and half for testing.

Finally, in order to obtain test metrics and compare our approach with a fully-supervised method, we have manually labelled all videos. Then, the training labels are used for training a supervised approach and the test labels are used to compute the CorLoc and mAP metrics. Roughly, we have labelled a total of 32238 objects where the less frequent class is 'van' with 30 samples and the most common is 'mobile-belt' with 6082.

4.2 Implementation details

We ran our experiments on a computer with 32 cores at 2.3 GHz, 256 GB of RAM and a GPU NVidia Titan X Pascal running with Python 3.6 and Ubuntu 18.04. Faster R-CNN is implemented in TensorFlow 1.13 and we use the pretrained weights on the Open Images dataset provided in the TensorFlow detection model zoo[4]. SpyNet is implemented in PyTorch 1.0 and we use the pretrained weights available in the project repository [5]. Finally, the

[4] We use the model `faster_rcnn_inception_resnet_v2_atrous_oid_2018_01_28.tar.gz`

[5] SpyNet: https://github.com/sniklaus/pytorch-spynet

we use a ResNet-50 model implemented in TensorFlow 1.3 with the pretrained weights available in the samples repository of TensorFlow [6]. For the clustering process, we use the implementation of HDBSCAN available in pip repository. Regarding the parameters commented in Sec. 3, after a cross-validation process on a subset of the training data, we have selected the following values: $T_S = 0.3, T_{F_x} = 0.3, T_{F_y} = 0.003, T_A = 200, T_{AR} = 0.5, T_I = 0.75$.

4.3 Performance evaluation

We use two metrics to evaluate the performance of our approach. On the one hand, for region localisation we use the *Correct Localisation* metric (CorLoc), adopted as well in [12,24]. This metric is defined as the percentage of objects correctly localised according to the Pascal criterion: the IoU between the predicted region and the ground-truth region is bigger than 0.5 for the RPN and bigger than 0.3 for the OFRP. Note that we use a smaller threshold for the optical flow case because the bounding-boxes tend to include the shadows, *i.e.* making the bounding-boxes wider or higher. On the other hand, for the object classification task, we use the Mean Average Precision (mAP) [6], which is the mean of the average precision (AP) across all classes.

4.4 Experimental results

We first examine the impact of our two region proposal algorithms (*i.e.* RPN and OFRP) described in Sec. 3 according to their individual CorLoc metrics. Secondly, we evaluate the accuracy of the clustering algorithm compared with other traditional clustering algorithms. Finally, we compare the performance of our proposed approach with a pretrained CNN for object detection and the same CNN but fine-tuned for our dataset.

Region Proposal comparative. In this experiment, we evaluate the performance of each region proposal algorithm (*i.e.* RPN and OFRP), in terms of the CorLoc metric, by comparing the proposals produced by each approach with the annotated ground-truth. Moreover, we compare the performance of each algorithm depending on the category of object to be detected.

Tab. 1 summarizes the CorLoc results (higher is better) for our two proposal algorithms (‘RPN’ and ‘OFRP’), together with the combination of both (‘RPN+OFRP’) and our final approach (‘Ours’) considering the clustering labels to filter detections (more details in Sec. 3.4). Moreover, we also include the results of a fine-tuned Faster R-CNN with manually labelled training data (‘Supervised-Faster’) as explained in Sec. 4.1. Note that each row contains results using a different IoU threshold during the computation of the metric. In the first row, we use the standard value of 0.5. However, as explained in the previous section, the OFRP requires a smaller threshold. Thus, in the second row we use two different thresholds, one for the ‘RPN’ (0.5) and a second one for the optical flow (0.3). According to the results, it is clear that the ‘OFRP’ produces more and more accurate regions than the ‘RPN’, mainly due to the huge scale differences between objects, as we pointed out in Sec. 3. If we focus

[6] ResNet-50: https://github.com/tensorflow/models/tree/master/official/resnet

	RPN	OFRP	RPN+OFRP	Ours	Supervised-Faster
CorLoc (0.5)	12.3	24.8	37.1	**86.8**	70.4
CorLoc (RPN@0.5, OFRP@0.3)	12.3	33.5	45.8	**96.2**	-

Table 1. CorLoc results for our two region proposal algorithms. Each row shows the results for a different IoU threshold used to compute the metric. Each column represents a different approach. Best result is marked in bold. More details in the text.

	car	ft	fuel	lgm	lg	mb	pe	pl	pb	st	van
RPN	8.3	37.5	41.4	3.7	11.3	7.1	1.3	**95.2**	7.7	3	25
OFRP	**78.4**	**60.7**	**57.8**	**74.2**	**88.2**	**72.8**	**79.1**	4.8	**64.7**	**84.8**	**37.5**

Table 2. CorLoc results per class using only true positives during the computation. Each row represents a different proposal algorithm and each column represents a different class. Best results are marked in bold. More details in the text.

on the effect of the threshold, we can see that the performance increases clearly when the threshold is softer because it allows the metric to take into account bounding-boxes that contain objects and their shadows. Finally, if we apply the clustering step and the manual label of clusters, we are able to remove detections grouped into useless clusters (*i.e.* containing only false positives) improving the results as shown in column 'Ours'. Comparing our final results with the fully-supervised network, our approach is able to produce better proposals even using the more restrictive metric with an IoU of 0.5.

In order to clarify the contribution of each proposal algorithm, we measure the CorLoc metric over the true positive set of each class. By this way, we will see the detection capabilities per class of the two algorithms. Tab. 2 summarizes the results for this experiment. As we can see, the 'OFRP' produces better regions for most of the classes and only for the 'plane' class obtains worse results. This is because the planes appear in a static situation in most of the frames, thus, there is no optical flow in that situation. Focusing on the the 'RPN' results, we can see that it only obtains better results for bigger object classes (*i.e.* planes, cars, vans, and different types of tracks), what validates our intuition and makes necessary the use of the 'OFRP' for small objects (*e.g.* persons) since the RPN has never seen objects with such an small area.

Class prediction comparative. In this experiment we focus on the class prediction part of the detection problem. Therefore, we try to predict the class appearing in the bounding boxes obtained from the previous step. Firstly, we compare two clustering algorithms: k-Means [14] and HDBSCAN (*i.e.* the one selected). Moreover, we compare two algorithms for dimensionality reduction: UMAP (see Sec. 3) and PCA. To measure the performance of the different approaches, we compare the AP per class and the mAP using the labels obtained after the clustering process of the detected objects. Tab. 3 summarises the results for this experiment. Each row represents a different algorithm, where 'kNC' means k-Means with NC clusters and 'HSC' means HDBSCAN with SC samples per cluster. Moreover, each row includes the dimensionality reduction algorithm used (*i.e.* PCA or UMAP). On the other hand, each column represents a different class. 'mAP' column represents the mean AP for all classes and 'mAPv' is the mean of the common classes to all rows (*i.e.* 'ft', 'fuel', 'mb' and 'pl'). Note that '-' means that there is no cluster that predicts that specific class, so there

Algorithm	car	ft*	fuel*	lgm	lg	mb*	pe	pl*	pb	st	van	mAP	mAPv
k50+PCA	-	0.718	0.532	-	-	0.249	0.515	0.303	-	-	-	0.210	0.450
H30+PCA	-	0.714	0.471	0.159	-	0.341	0.367	0.364	-	-	-	0.219	0.472
H30+UMAP	-	0.718	0.746	0.264	-	0.487	0.451	0.419	0.001	-	-	0.280	0.592
H50+PCA	-	0.383	0.439	-	-	0.347	0.436	0.404	-	-	-	0.182	0.393
H50+UMAP	-	0.730	0.970	-	-	0.513	0.542	0.454	-	-	-	0.291	0.666
Baseline	-	0.561	0.640	-	-	0.102	-	0.831	-	-	-	0.194	0.533
Supervised-Faster	0.899	0.782	0.980	0.873	0.970	0.969	0.105	0.969	0.949	0.04	0	**0.685**	**0.925**

Table 3. AP results per class. Each row represents a different algorithm and each column represents a different class. 'mAP' column represents the mean AP for all classes and 'mAPv' is the mean of valid AP values. Only classes marked with '*' are considered for mAPv metric. Best results are marked in bold. More details in the text.

are no valid AP results and we ignore them during the computation of 'mAPv'. Focusing on the k-Means results, we can see that the performance is worse than using HDBSCAN with a cluster size of 30 samples with PCA. Therefore for the next experiments we focus on HDBSCAN. Comparing PCA with UMAP, we can see that in all cases, UMAP clearly boosts the results demonstrating their better dimensionality reduction performance. Then, if we compare the different cluster size values for HDBSCAN, it is clear that bigger clusters benefit the performance of our approach, specially if we only take into account the valid classes. We want to clarify that non-valid classes are due to the low number of samples (*i.e.* around one hundred compared to thousands of images for the other classes) available during training for those classes. Thus, the clustering algorithms tend to assign them to bigger clusters. For example, 'car' and 'push-back' objects only appear in the scene during the arrival and departure of the planes, respectively. Therefore, the number of samples is very limited compared with other classes that appear more frequently in the scene. Finally, the last two rows summarise the results for the baseline using a pretrained Faster R-CNN and a fine-tuned model using our trained data, respectively. Comparing both results with our best approach (H50+UMAP), we can see that our approach improves the results obtained by the baseline. However, the fine-tuned model obtains the best results on average, as it has been trained in a supervised way. Comparing the class-specific results with our best approach, we can see that the model also suffers with classes that have few samples (*i.e.* stairs or van). Moreover, focusing on *person* class – which appears in the videos with low contrast, very noisy frames, many shape changes and small bounding-boxes – our approach overcomes the results obtained by the fine-tuned model. Mainly, because our OF-based region proposal is more robust in those situations, as the fine-tuned model is not able to obtain a good representation for *persons.*

5 Conclusions

We have presented a weakly-supervised approach for automatic object discovery in videos. Our method consists of two main components: a region proposal, which produces the bounding-boxes, and a clustering algorithm, which groups similar detections to assign them an unsupervised label. We have tested our approach on video sequences of the *apron area* of an airport showing that it is able to detect and classify automatically objects appearing on those videos. Moreover, the collaboration of the human is only necessary in order to assign human understandable labels. Therefore, our approach is able to run automatically without the collaboration of the human.

Regarding the region proposal algorithms (RPN or OFRP), we have demonstrated that the combination of a pretrained RPN together with an pretrained OFRP, is able to improve the results obtained by a fine-tuned model for the specific problem. Moreover, our approach is especially robust dealing with small regions and classes with changes in the shape (*e.g.* persons).

Regarding the clustering algorithm, our results show that HDBSCAN combined with UMAP improves traditional approaches such as k-Means and PCA. In this case, the fully-supervised approach (fine-tuned Faster R-CNN) obtains the best results, but our weakly-supervised approach is able to obtain better results than the evaluated baseline. As future work, we plan to use the detections produced by our approach to retrain iteratively the CNN models in order to obtain better results in each iterative step with a minimum labelling process. By this way, the gap between our weakly-supervised approach and the fully-supervised network should decrease in each iteration.

References

1. Cai, Z., Vasconcelos, N.: Cascade R-CNN: Delving into high quality object detection. In: CVPR. pp. 6154–6162 (2018)
2. Campello, R.J., Moulavi, D., Zimek, A., Sander, J.: Hierarchical density estimates for data clustering, visualization, and outlier detection. TKDD **10**(1), 5 (2015)
3. Cho, M., Kwak, S., Schmid, C., Ponce, J.: Unsupervised object discovery and localization in the wild: Part-based matching with bottom-up region proposals. In: CVPR. pp. 1201–1210 (2015)
4. Deselaers, T., Alexe, B., Ferrari, V.: Localizing objects while learning their appearance. In: ECCV. pp. 452–466 (2010)
5. Durand, T., Mordan, T., Thome, N., Cord, M.: Wildcat: Weakly supervised learning of deep convnets for image classification, pointwise localization and segmentation. In: CVPR. pp. 642–651 (2017)
6. Everingham, M., Van Gool, L., Williams, C.K.I., Winn, J., Zisserman, A.: The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results. http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html
7. Felzenszwalb, P.F., Girshick, R.B., McAllester, D., Ramanan, D.: Object detection with discriminatively trained part-based models. IEEE PAMI **32**(9), 1627–1645 (2010)
8. Gokberk Cinbis, R., Verbeek, J., Schmid, C.: Multi-fold mil training for weakly supervised object localization. In: CVPR. pp. 2409–2416 (2014)
9. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR. pp. 770–778 (2016)
10. Kantorov, V., Oquab, M., Cho, M., Laptev, I.: Contextlocnet: Context-aware deep network models for weakly supervised localization. In: ECCV. pp. 350–365 (2016)
11. Koh, Y.J., Kim, C.S.: Unsupervised primary object discovery in videos based on evolutionary primary object modeling with reliable object proposals. IEEE Transactions on Image Processing **26**(11), 5203–5216 (2017)
12. Kwak, S., Cho, M., Laptev, I., Ponce, J., Schmid, C.: Unsupervised object discovery and tracking in video collections. In: ICCV. pp. 3173–3181 (2015)

13. Li, Y., Liu, L., Shen, C., van den Hengel, A.: Image co-localization by mimicking a good detectors confidence score distribution. In: ECCV. pp. 19–34 (2016)
14. Lloyd, S.: Least squares quantization in pcm. IEEE transactions on information theory **28**(2), 129–137 (1982)
15. McInnes, L., Healy, J., Melville, J.: Umap: Uniform manifold approximation and projection for dimension reduction. arXiv preprint arXiv:1802.03426 (2018)
16. Ošep, A., Voigtlaender, P., Luiten, J., Breuers, S., Leibe, B.: Large-scale object discovery and detector adaptation from unlabeled video. arXiv preprint arXiv:1712.08832 (2017)
17. Peyre, J., Sivic, J., Laptev, I., Schmid, C.: Weakly-supervised learning of visual relations. In: ICCV. pp. 5179–5188 (2017)
18. Pot, E., Toshev, A., Kosecka, J.: Self-supervisory signals for object discovery and detection. arXiv preprint arXiv:1806.03370 (2018)
19. Ranjan, A., Black, M.J.: Optical flow estimation using a spatial pyramid network. In: CVPR. pp. 4161–4170 (2017)
20. Ren, S., He, K., Girshick, R., Sun, J.: Faster R-CNN: Towards real-time object detection with region proposal networks. In: NIPS. pp. 91–99 (2015)
21. Shi, Z., Hospedales, T.M., Xiang, T.: Bayesian joint topic modelling for weakly supervised object localisation. In: ICCV. pp. 2984–2991 (2013)
22. Suzuki, S., et al.: Topological structural analysis of digitized binary images by border following. Computer Vision, Graphics and Image Processing **30**(1), 32–46 (1985)
23. Viola, P., Jones, M., et al.: Rapid object detection using a boosted cascade of simple features. CVPR **1**, 511–518 (2001)
24. Vo, H.V., Bach, F., Cho, M., Han, K., LeCun, Y., Perez, P., Ponce, J.: Unsupervised image matching and object discovery as optimization. arXiv preprint arXiv:1904.03148 (2019)
25. Wang, L., Hua, G., Sukthankar, R., Xue, J., Niu, Z., Zheng, N.: Video object discovery and co-segmentation with extremely weak supervision. IEEE PAMI **39**(10), 2074–2088 (2017)
26. Wang, X., Peng, Y., Lu, L., Lu, Z., Bagheri, M., Summers, R.M.: Chestx-ray8: Hospital-scale chest x-ray database and benchmarks on weakly-supervised classification and localization of common thorax diseases. In: CVPR. pp. 2097–2106 (2017)
27. Wei, Y., Liang, X., Chen, Y., Shen, X., Cheng, M.M., Feng, J., Zhao, Y., Yan, S.: Stc: A simple to complex framework for weakly-supervised semantic segmentation. IEEE PAMI **39**(11), 2314–2320 (2017)
28. Xu, Y., Kong, Q., Wang, W., Plumbley, M.D.: Large-scale weakly supervised audio classification using gated convolutional neural network. In: Proc. ICASSP. pp. 121–125 (2018)
29. Ye, Q., Zhang, T., Ke, W., Qiu, Q., Chen, J., Sapiro, G., Zhang, B.: Self-learning scene-specific pedestrian detectors using a progressive latent model. In: CVPR. pp. 509–518 (2017)
30. Yu, H., Siskind, J.M.: Sentence directed video object codiscovery. IJCV pp. 312–334 (2017)
31. Zhang, D., Han, J., Yang, L., Xu, D.: Spftn: A joint learning framework for localizing and segmenting objects in weakly labeled videos. IEEE PAMI (2018)