Frame Augmentation for Imbalanced Object Detection Datasets

Nada Elasal
David M. Swart
Nicholas Miller

Miovision
Kitchener, ON
Canada

Abstract

A major challenge in most object detection datasets is class imbalance. It is especially apparent in uncurated datasets where frames originate from a real-world setup such as a set of cameras collecting data from fixed locations. In that case, the dataset class distribution mirrors the real-world distribution, causing a bias towards over-represented classes if used for model training. In this paper we propose a synthesis technique for balancing the dataset, which exploits having sets of frames from the same camera view. The result is synthesized frames containing only rare objects, while guaranteeing realistic object placement both in terms of scene context and perspective. We train a deep learning object detection model on the augmented dataset and compare its performance to a model trained on the original, imbalanced dataset. Results show that including the synthesized frames in the training results in a significant performance boost for the rare classes.

1 Introduction

Deep convolutional neural networks (CNN) have become the most successful approach for object detection in images [3, 4, 7, 8]. Object detectors locate and identify all instances of desired classes in an image. They typically output enclosing rectangles. Similarly, they are trained on a large set of images where all objects in each image have been manually enclosed in bounding rectangles with labels specifying class. Training the object detector involves minimizing a loss function which is designed to reward labelled boxes from the neural network that closely match the manual rectangles and penalize mislabeled and false rectangles and misses.

One common classification challenge is class imbalance in the training set. While this is often a true reflection of the class distribution in the real-world (e.g., many more cars in traffic scenes than bicycles), it can lead to a strong bias in detectors trained on those datasets. To minimize the overall training loss, models may favour performance on the common classes and ignore the rare ones. In many applications missing rare objects is not acceptable. Previous research [1] suggests that simply weighting the loss function to compensate for underrepresented classes is not the best approach. It is best if samples of rare and common objects are balanced in the training set.

We address the object class imbalance in a successor to the MIO-TCD road scene traffic dataset [5]. This object detection dataset consists of traffic images containing vehicles labelled {Articulated Truck, Bicycle, Bus, Car, Motorcycle, Pedestrian, Pickup Truck, Single Unit Truck, or Work Van}. Due to road usage distribution, Articulated Truck, Bicycle, Motorcycle, Bus and WorkVan are rare object classes while Cars and Pedestrians are common.

There are two key features of the new dataset that we will take advantage of. First, the images are extracted from various videos captured from fixed cameras at intersections. This means that for a given scene there is a set of frames distributed over a long period of time. Second, the objects are annotated with a tight bounding polygon masks rather than bounding rectangles. See the orange and blue masks shown in Fig 1.

Previous work shows that a successful technique in addressing the class imbalance problem for CNNs is oversampling the under-represented classes [1]. In the case of object detection training data, properly sampling rare objects requires them to appear somewhere in the training images embedded in their natural context and surrounding [2]. Oversampling requires plenty of rare objects to appear in many training images, comparable to the number of common objects.

At first it seems tempting to merely clone frames from this dataset which contain rare vehicles. There two are issues with this approach: frames with rare objects often contain many instances of a common object defeating the goal of class re-balance; furthermore, the rare objects fail to be shown in novel contexts. Our new approach addresses these issues by combining different annotated rare objects accumulated over a time window and background images (those deemed via annotation to contain no objects) to synthesize many new training frames. The new frames contain several instances of rare objects for training in new but realistic contexts. See Fig. 1.

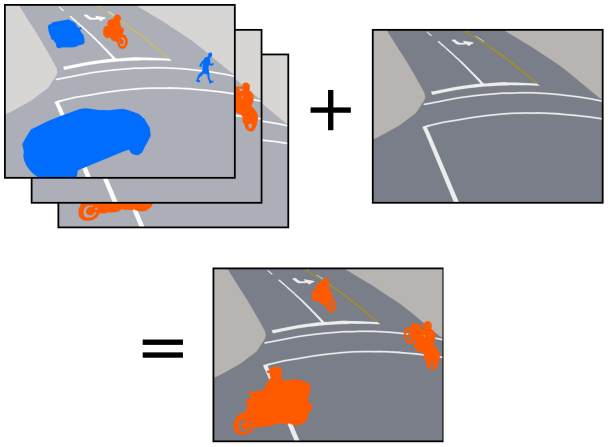

Fig. 1: Annotated masks as they appear in the dataset. Common object classes (Cars and Pedestrians) are shown in blue while relatively rare objects (Motorcycles) are orange. Rare objects are accumulated from video frames. They are combined with a background image from the same scene to generate new training images which oversample rare objects in new but realistic settings

Section 2 describes the detailed algorithm for frame synthesis and in section 3 we present the results and compare a model trained on the new dataset to a baseline model trained on the original dataset.

2 Algorithm

To boost the number of occurrences of rare classes in the dataset, we need frames that only contain those rare objects without additional common ones. Since such frames do not naturally occur in the dataset, we propose a method to synthesize them. Furthermore, we define an additional constraint on the scene placement of objects in the synthesized frames: objects should be placed in realistic locations in the image, respecting both context (e.g., vehicles should not be placed on sidewalks) and perspective (e.g., a location close to the camera should not contain an unrealistically small vehicle).

To create these frames, we segment the original dataset into sets of frames from different scenes $S = S_1, S_2, ..., S_M$, where S_i is a set of frames from the same camera view, separated by time. The main idea is to find the set of all annotations A of the rare objects in a given scene S_i, sample from those masks and paste them on an empty view of S_i at the same position as they originally appeared (Fig. 1). To find suitable backgrounds, we find the set of all frames in S_i with no annotations, B. To create one synthesized frame, one background is sampled from B and a set of k masks are sampled from A and pasted on the background in their original location. To ensure realistic topology, if a sampled mask overlaps with an existing pasted mask, the former is discarded. This process is repeated over all scenes in S until the class distribution of the combined dataset is roughly uniform. Note that it is not guaranteed that each scene will contain empty frames. Yet to create a reasonable set of synthesized frames it is enough if only some of the scenes in S satisfy this requirement. Fig. 2 shows example synthesized frames. Note that the placement of objects in the scene is realistic which preserves the real-world context for the neural network model to learn. It is worth mentioning that the randomization aspect due to the sampling of a set of masks and an empty frame introduces a favourable diversity in the set of synthesized frames.

Fig. 2: Example simulated frames used for oversampling. From left to right: Single Unit Trucks, Motorcycles, and Pickup Trucks

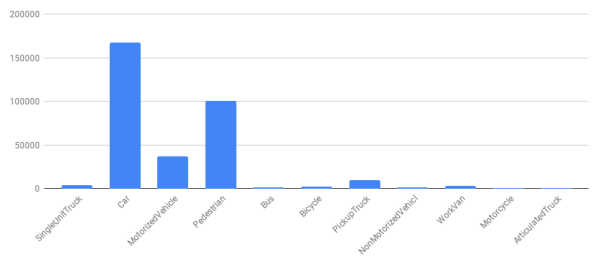

Fig. 3: Distribution of objects by class in the original training dataset

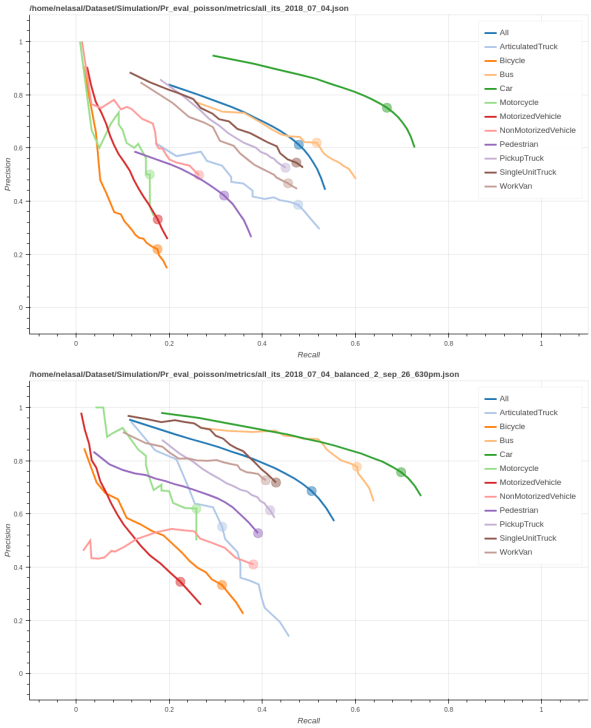

Fig. 4: Precision vs recall graph of the object detection results on D_{test}. Top is the baseline, bottom is after training with the balanced dataset. Points of optimal F1 are indicated with circles

3 Results

3.1 Dataset

Our dataset consists of traffic scenes with polygon annotations of $N = 11$ classes. To build the training dataset, we split the real frames into training and test sets, D_{train} and D_{test} respectively. The initial class distribution of D_{train} is shown in Fig 3. Next, we run the algorithm described in section 2 on D_{train} to create an additional set of simulated frames D_{sim} to balance out the class distribution. Finally, we combine D_{train} and D_{sim} to build the balanced training set for the object detection model, resulting in a total of of 54,067 frames. D_{test} (10,939 frames) is used as the model validation set. Note that validation is performed on real frames only to ensure that evaluation results mirror model performance in the real-world.

3.2 Model

We use the SSD object detection framework with MobileNet as a base network [3, 4]. The output of the model is a list of detection with each detection being a vector of 4 box coordinates $b = [b_{top}, b_{left}, b_{bottom}, b_{right}]$ and a list of class confidence values $c = [c1, c2, ..., cN]$. The network typically produces a large number of detections. As in the original paper a confidence threshold c_{th} is applied to filter out low confidence boxes, where a detection is discarded if $\max(c) < c_{th}$. Finally a non-maximum suppression step is applied to produce the final detections.

3.3 Evaluation Framework

Given a set of labelled predicted boxes and a set of labelled ground truth boxes we want to compute the detection accuracy of the model. We formulate the problem as a minimum weight matching problem in bipartite graphs, where the cost of matching a ground truth box to a prediction box is their intersection-over-union (IoU). This matrix is calculated for each class separately (i.e., only predictions with the same label as a given ground truth detection are candidates for matching). The optimal assignment is computed using the Hungarian algorithm. To ensure proximity of detection pairs, matches with an $IoU < 0.5$ are discarded. The set of remaining pairs represent true positives. Unmatched ground truth boxes are false negatives and unmatched predictions are false positives. Finally we compute precision and recall metrics. To plot a precision-recall (PR) curve, we vary the decision parameter c_{th} within the range $[0, 1]$ (Fig 4).

3.4 Comparison

Using the evaluation framework described in the previous section we compare results of a model trained on $D_{train} + D_{sim}$ to a baseline model only trained on real images D_{train}. Both are evaluated on the same validation set D_{test}. Fig 4 shows the PR curves by class for both models. Table 1 summarizes the F1-score by class at the optimal confidence threshold value c_{th} that maximizes the class F1-score. Results show that all classes except for one have significantly improved F1-scores with the model trained on both real and simulated images. Fig 5 shows the F1 improvement by class in percent, with top improvements at 65% (Bicycle) and 51% (Motorcycle) from the baseline. The clear improvement across classes suggests that the simulation technique has significantly contributed to boosting the accuracy of the object detection model on under represented classes.

4 Future Work

There may be an opportunity to improve our results even further by exploring other ways to paste and blend vehicles onto a background. See a discussion about such blending techniques by Dwibedi et al. [2]. Our straightforward pasting method does not do any blending; it uses the fact that the source and destination images have the same background scene (separated only by a little time). Even so, we do see that our pasted vehicles have a sharp, high-

	Articulated Truck	Bicycle	Bus	Car	Motorcycle	Pedestrian	Pickup Truck	Single Unit Truck	WorkVan
Baseline (D_{train})	**0.4269**	0.195	0.564	0.707	0.241	0.362	0.485	0.506	0.461
Balanced ($D_{train} + D_{sim}$)	0.4	**0.323**	**0.679**	**0.726**	**0.365**	**0.449**	**0.496**	**0.537**	**0.522**

Table 1: Comparison of optimal F1 scores on D_{test} for model trained on D_{train} and the balanced set ($D_{train} + D_{sim}$). The best score for each class is highlighted in bold

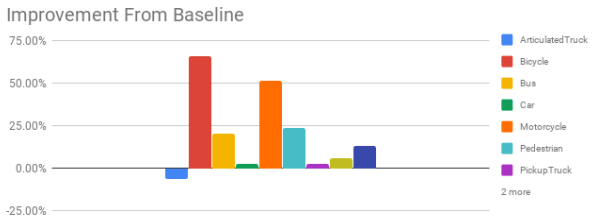

Fig. 5: Relative improvement in F1 score for model trained on the balanced dataset $D_{train} + D_{Sim}$

contrast edge which looks unnatural to our human eyes.

Early experiments to use a Poisson blending technique [6] have resulted in images that often look more natural. However the resulting detection results did not improve as much as expected. More work is needed to study why this was the case, as well as to explore other blending techniques.

References

[1] M. Buda, A. Maki, M. Mazurowski. A systematic study of the class imbalance problem in convolutional neural networks. *Neural Networks, vol. 106, pp. 249–259* (2018)

[2] D. Dwibedi, I. Misra, M. Hebert. Cut Paste and Learn: Surprisingly Easy Synthesis for Instance Detection. *International Conference on Computer Vision* (2017)

[3] A. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, H. Adam. MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. *arXiv, http://arxiv.org/abs/1704.04861* (2017)

[4] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C. Fu, A. Berg. SSD: Single Shot MultiBox Detector. *arXiv, http://arxiv.org/abs/1512.02325* (2015)

[5] Z. Luo, F.B.Charron, C.Lemaire, J.Konrad, S.Li, A.Mishra, A. Achkar, J. Eichel, P-M Jodoin. MIO-TCD: A new benchmark dataset for vehicle classification and localization. *IEEE Transactions on Image Processing* (2018)

[6] P. Pérez, M. Gangnet, A. Blake. Poisson Image Editing. *ACM Transactions on Graphics, vol. 22, pp. 313–318* (2003)

[7] S. Ren, K. He, R. Girshick, J. Sun. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2017)

[8] J. Redmon, A. Farhadi. YOLO9000: Better, Faster, Stronger. *IEEE Conference on Computer Vision and Pattern Recognition* (2017)