Learning To Filter Object Detections

Sergey Prokudin*[1], Daniel Kappler*[1], Sebastian Nowozin[2], and Peter Gehler[1]

[1] Max Planck Institute for Intelligent Systems, Tübingen, Germany,
first.lastname@tuebingen.mpg.de
[2] Microsoft Research, Cambridge, UK

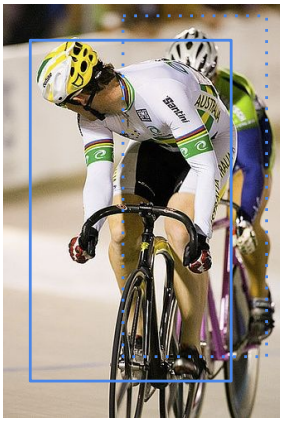
Abstract. Most object detection systems consist of three stages. First, a set of individual hypotheses for object locations is generated using a proposal generating algorithm. Second, a classifier scores every generated hypothesis independently to obtain a multi-class prediction. Finally, all scored hypotheses are filtered via a non-differentiable and decoupled non-maximum suppression (NMS) post-processing step. In this paper, we propose a filtering network (FNet), a method which replaces NMS with a differentiable neural network that allows joint reasoning and rescoring of the generated set of hypotheses per image. This formulation enables end-to-end training of the full object detection pipeline. First, we demonstrate that FNet, a feed-forward network architecture, is able to mimic NMS decisions, despite the sequential nature of NMS. We further analyze NMS failures and propose a loss formulation that is better aligned with the mean average precision (mAP) evaluation metric. We evaluate FNet on several standard detection datasets. Results surpass standard NMS on highly occluded settings of a synthetic overlapping MNIST dataset and show competitive behavior on PascalVOC2007 and KITTI detection benchmarks.

1 Introduction

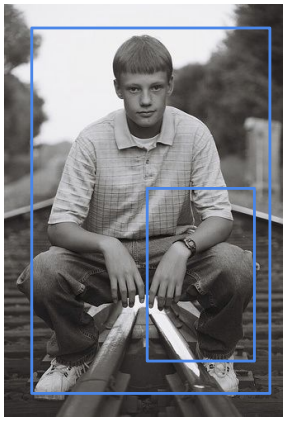
Object detection is a fundamental structured prediction problem in computer vision. This problem is regularly approached with three main processing steps. In the first *region proposal* step a set of object hypotheses is generated using a proposal algorithm. Second, a multi-class classifier scores each hypothesis independent of all other hypotheses. We further refer to this as *proposal classification* step. In a final *filtering step* the redundant hypotheses are suppressed via non-maximum suppression (NMS).

The final filtering step is typically crucial in order to achieve good performance, e.g. on PascalVOC this step doubles the performance. Nevertheless, today NMS is still the main building block of current detection algorithms and is used frequently in most modern detection algorithms [13, 14]. Greedy sequential NMS consists of the following heuristic steps: (i) sort proposals according to their classification scores, (ii) start from the highest scoring hypothesis remove

* both authors contributed equally to this work



(a) Highly occluded instances would be suppressed



(b) Parts of objects could be not suppressed

Fig. 1: Examples of NMS failures.

all hypotheses with an overlap of a predefined threshold, (iii) repeat step (ii) until all hypotheses have been selected or removed.

Speed, ease and effectiveness are strong positive points but NMS also has some drawbacks:

- **non-adaptive**: The NMS decision rule is hard-coded using the proposal classification scores, overlap ratio between hypotheses and a single predefined threshold. Therefore, it does not allow to "reason-away" sets of bounding boxes, a feature that would entail more complex and flexible features.
- **non-differentiable**: NMS is a greedy, sequential, heuristic procedure applied separately of bounding box scoring. It prevents the former classification step from being jointly trained for the final loss function.

Figure 1 shows two common NMS failure cases: (a) suppressing nearby detections of highly occluded objects and (b) not suppressing hypotheses representing only parts of an object, i.e. the knee.

Recently, much progress has been made in improving the individual classification results [6, 16] and fusing the proposal generation and classification steps [12–14, 16]. Yet, only a few approaches have been proposed in order to replace the final sequential NMS step, e.g., by [8]. However, the latter approach is not differentiable since it uses NMS features and thus hinder end-to-end training of the entire detection pipeline.

In this paper we aim to take some steps to turn the NMS process into a differentiable building block that can be used in conjunction with any multi-class classifier. There are some features of NMS that make this a challenging task and we take some careful steps to not loose the performance of NMS while proposing a replacement that can be used in a wider context. We propose to replace the sequential NMS step with an additional feed-forward neural network that can

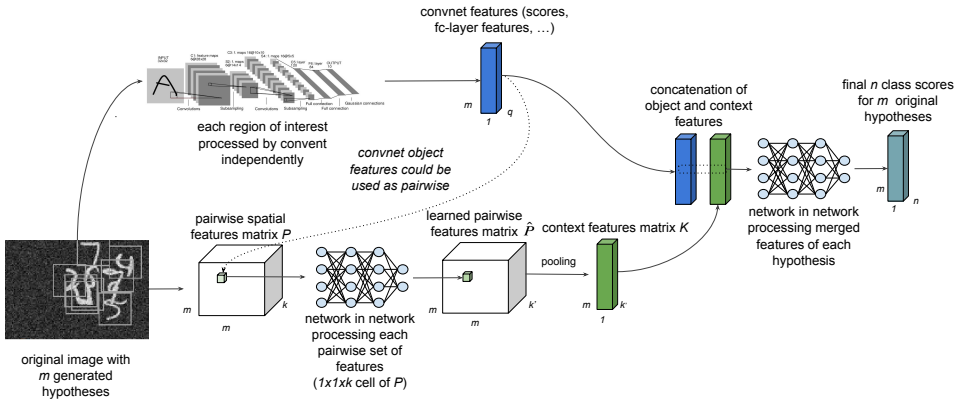


Fig. 2: FNet overview. Example on overlapping MNIST dataset, LeNet [10] is used for independent hypotheses classification.

be stacked on top of any existing classifier. For the remainder of this paper, we refer to this add-on as *filtering network*, or in short FNet. In contrast to the existing classifier, the proposed FNet processes the proposal hypotheses *jointly*, propagating errors to the individually processed hypotheses of the existing classifier.

Figure 2 depicts an overview of our proposed architecture, illustrated with LeNet [10] as the basic per hypothesis classifier. The main idea for the FNet structure is to use all information provided by all hypotheses in order to learn context features which allow to filter hypotheses based on global knowledge. The architecture is described in more detail in Section 4.

In order to verify ability of our approach to perform structured reasoning over a set of hypotheses, we replace NMS by learning an approximate NMS objective (Section 5.1). We demonstrate that FNet can reproduce NMS with high accuracy. Since FNet is composed of standard neural network components and has no sequential steps it achieves this performance while adding only a minor computational overhead to the detection pipeline. We further introduce a new loss function (net-loss), a sufficient objective to improve directly on the mean average precision (mAP) objective [5]. Results are reported on three datasets. Our experiments indicate that by leveraging features from the classifier networks, we are able to surpass NMS performance on a synthetic overlapping MNIST (oMNIST) dataset while being on par on KITTI and PascalVOC2007.

Thus, in summary the main contribution of this work is to replace NMS with a simple fully differentiable feed-forward network. FNet is is independent of the number of hypotheses, allows to make decisions over a set of hypotheses, and can be stacked on top of pre-trained object detection pipelines.

FNet is implemented in TensorFlow [1]. Just as traditional NMS post-processing, it can be easily combined with any existing object detection model.

2 Related work

There are two interconnected streams of research related to our approach in the literature: the first one aims to replace NMS with something more flexible, while the second one concentrates on building an end-to-end object detection pipeline.

Learning Non-Maximum Suppression: The work of [15] analyzes the drawbacks of sequential NMS, and proposes to use an affinity propagation algorithm to pass information between hypotheses. While sharing the common high-level idea of using information between pairs of hypotheses to perform better filtering, this method differs from ours in the model used for describing hypotheses interactions and the loss function being optimized. Another promising approach is shown in [8]. There, all hypotheses are mapped to a spatial grid based on their center locations. The extracted pre-trained classification scores and intersections between hypotheses are then used as inputs for a convolutional network operating on this grid structure. Another aprroaches based on Hough transform were proposed in [2], [9]. However, no end-to-end optimization was shown for these NMS replacements, leaving the approach detached from the underlying per hypotheses classifier networks.

End-to-end learning of object detection pipeline: Wan et al. [18] proposes a method to incorporate an object detector, deformable parts model and NMS in a fully differentiable pipeline. Nevertheless, the NMS step still remains a fixed transformation over a set of hypotheses, reformulated as a layer performing a particular operation. The same applies to Henderson et al. [7], who propose a way to propagate gradients directly for the mean average precision (mAP) loss. There NMS is treated similar to a max-pooling step, where only the hypotheses representing a local-maximum propagate gradients. The problems illustrated in Figure 1(a) therefore remain unchanged; hypotheses falling under the suppression condition will still be pruned out.

Finally, Stewart et al. [17] replace the NMS post-processing stage with LSTM cells in order to achieve better spatial reasoning for neighbouring hypotheses. However, this method requires the image to be divided into a regular grid of independent regions, e.g. 15×20, while predictions across regions are merged via a heuristic stitching step.

3 Problem Formulation

As mentioned before, the object detection pipeline is regularly a combination of three steps - searching for good hypotheses, generating independent predictions for each of them, and joint filtering of the final set. Our work focuses on the last step. Given a set of hypotheses for an image I, we assume the following information to be provided

$$H = \{\big[h_i, s(h_i), f(h_i)\big], i = 1, \dots, m\}, \tag{1}$$

where $h_i \in \mathbb{R}^4$ are hypothesis bounding box coordinates (regularly represented as (x, y)-coordinates of upper left corner and width and height of a box). Scores are denoted by $s(h_i) \in \mathbb{R}^n$ for all n classes of interest. This is the output for every hypothesis from the proposal classification step. $f(h_i) \in \mathbb{R}^q$ is a feature vector per hypothesis (for example, CNN features). Here and below square brackets stand for concatenation of the vectors. The total number of hypotheses per image m can vary between images.

During training, ground truth hypotheses are denoted by G:

$$G = \{\big[g_i, c_i\big], i = 1, \dots, d\}, \tag{2}$$

where $g_i \in \mathbb{R}^4$ - ground truth bounding box coordinates, $c_i \in \{1, \dots, n\}$ - class label for the ground truth, d is the total number of ground truth objects on image. The proposed filtering step re-scores all class scores for every hypotheses $h_i \in H$ while considering all other hypotheses in H:

$$H \rightarrow H' = \{\big[h_i, s'(h_i, H), f(h_i)\big], i = 1, \dots, m\}. \tag{3}$$

The classical NMS can be considered as copying the scores for unsuppressed hypotheses $s'(h_i) = s(h_i, H)$ and setting it to zero vector for suppressed ones $s'(h_j, H) = \hat{0}$. The new scores $s'(h_i, H)$ typically aim to minimize the mAP evaluation metric, described in [5] and discussed in Section 5.2.

4 Filtering network architecture

The focus of this work is on differentiable, thus, end-to-end learnable filtering of hypotheses for multi-class object detection. Our proposed *filtering network* FNet allows to optimize the underlying score generating network not only based on the scores but also its features in order to generate one matching hypothesis per ground truth bounding box labeling. The main idea of FNet is to utilize all the information (eq. 1) provided by the earlier steps of a pipeline by building a pairwise matrix P (eq. 5) in order to learn context features that will allow to filter hypotheses based on global knowledge. Thus, FNet is designed to directly solve the filtering step formalized in (eq. 3):

$$s'(h_i, H) = FNet(h_i, H), i = 1, \dots, m. \tag{4}$$

We start by building the pairwise matrix P

$$P_{i,j} = [f(h_i), f(h_j), h_i, h_j], \quad P \in \mathbb{R}^{m \times m \times k}, \tag{5}$$

consisting of two types of features, the feature vectors of the per-hypothesis network and the corresponding hypotheses locations. Based on P, we learn a new pairwise matrix $\hat{P}$

$$\hat{P}_{i,j} = \mathrm{NiN}^{\mathrm{pairwise}}(P_{i,j}), \quad \hat{P} \in \mathbb{R}^{m \times m \times k'}, \tag{6}$$

where NiN represents a network in network [11] which is convolved over the pairwise matrix P. The main idea behind the approach is that, given pairwise set of features $P_{i,j} \in \mathbb{R}^k$ (e.g. scores and features for both hypotheses under consideration plus ratio of overlap between corresponding bounding boxes), we learn a small network to abstract this data and produce a new feature vector $\hat{P}_{i,j} \in \mathbb{R}^{k'}$ that will represent learned relations between pair of hypotheses.

Since every image can have a different number of hypotheses m, we apply a reduction operator:

$$K = R(\hat{P}) : \mathbb{R}^{m \times m \times k'} \rightarrow \mathbb{R}^{m \times k'}, \tag{7}$$

which results in fixed sized context feature matrix K. Each i'th row of this matrix represents context features vectors for hypothesis i that we denote as $K_i = K(h_i)$. The original feature vector $f(h_i)$ is fused with its context feature vector $K(h_i)$ via another network in network, producing the final score $s'(h_i)$ for the given hypothesis

$$s'(h_i, H) = \text{NiN}^{\text{context}}\left(f(h_i), K(h_i)\right), \quad i = 1, \ldots, m. \tag{8}$$

Input features As discussed in the previous section, we generally consider two types of input features, network features $f(h_i)$ provided by earlier stages of the pipeline, and location coordinates h_i. Since there is usually problem specific knowledge present, in practice it is very helpful to add additional function of the input features

$$P_{i,j} = [f(h_i), f(h_j), f(h_i) - f(h_j), \text{sign}(f(h_i) - f(h_j)), \text{IoU}(h_i, h_j)], \tag{9}$$

where IoU stands for intersection over union between hypotheses areas, and sign returns an element-wise indication of the sign of a vector. Adding the difference between hypotheses and the sign provides a helpful signal to the network in order to decide whether or not there exists a better scored hypothesis. Using IoU as a feature provides further evidence of the relationship between two hypotheses besides their feature vectors and scores.

Reduction operator We select a combination of simple maximum and average pooling operations as the reduction operation of choice for all the considered experiments:

$$k_{it} = \Big[\max_j \hat{P}_{i,j,t}, \frac{1}{m}\sum_j \hat{P}_{i,j,t}\Big], t = 1, \ldots, k', K \in \mathbb{R}^{m \times k'} \tag{10}$$

5 Learning objectives

We explore two possible learning objectives to optimize FNet. In order to verify the learning capacity of the proposed FNet architecture, we introduce a loss

function that will force FNet to mimic decisions made by NMS. We hypothesize in Section 5.2 that a new loss function is required in order to improve over the previously presented NMS error cases. Henderson et al. [7] show that mAP is a complex structured loss over thousands of hypotheses, thus, in this paper we propose to substitute the mAP with a proxy loss function (net-loss) that can be evaluated on a per image basis.

5.1 Approximate Non-Maximum Suppression Objective

We can approximate the sequential NMS process with fixed IoU threshold a by decomposing decisions made for every hypothesis h_i with class score s_i into the following per-hypothesis labels:

$$l_z(h_i) = \begin{cases} 0, & \text{if } \exists h_j : \text{IoU}(h_i, h_j) > a \text{ and } s_z(h_j) > s_z(h_i); \\ 1, & \text{otherwise;} \end{cases} \tag{11}$$

where $z = 1, \ldots, n$, n is the number of classes and $s_z(h_i)$ is the score for class z and hypothesis i. Based on these labels, we can optimize a multi-class objective such as cross-entropy for the FNet score $s'(h_i, H)$ and the target labels of eq. 11. In that case filtering scores $s'(h_i, H)$ cannot be used directly to represent class probabilities, since the network learns to mimic suppression. It is not aware of cases when a hypothesis is not suppressed by NMS. Since the hypothesis itself might have a low independent score itself, we obtain the final score per hypothesis h_i by multiplying the original score $s(h_i)$ with our filtering value $s'(h_i, H)$. This approach results in decreased scores for those of the hypotheses that have high suppression probabilities:

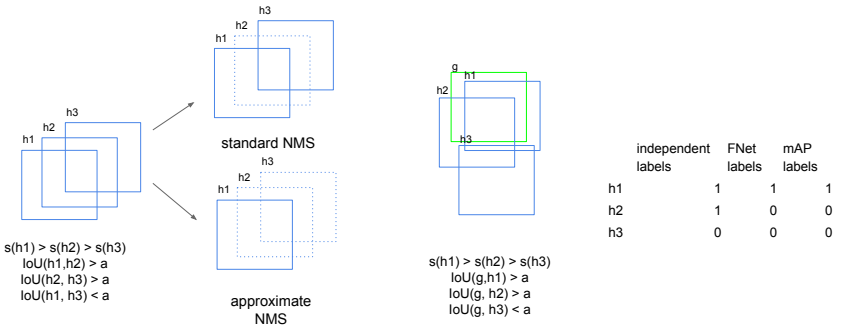
$$s''(h_i) = s(h_i) \cdot s'(h_i, H). \tag{12}$$

It is important to mention that there are cases when this non-sequential labeling will result in different selected set of hypotheses compared to the sequential NMS as shown in Figure 3(a). For illustrative purposes, we assume that we have an ordered set of hypotheses with class scores $s(h_1) > s(h_2) > s(h_3)$. According to our labeling procedure, the lowest scoring hypothesis h_3 will have a zero label because of the high overlap with h_2 and the higher score. Whereas NMS will process all hypotheses sequentially starting from h_1, selecting it, then suppressing the highly overlapping h_2, and finally selecting h_3 because of absence of h_2 in the remaining set.

In the majority of cases, however, the selected sets behave very similar. For example, for proposals produced by FasterRCNN [14] on the PascalVOC2007 test set, decisions made by normal NMS and our approximate version agree on 98.1% of all hypotheses under consideration.

5.2 Network Detection Objective

The previously introduced labeling and loss for learning NMS only functions as a testbed showing the capabilities of our FNet architecture, capable of explaining



(a) Difference in behaviour of standard and approximate NMS procedures.

(b) Comparison of different hypothesis labelings used to train the network.

Fig. 3: Overview of learning objectives.

away hypotheses in order to reproduce NMS results. Yet, this formulation will at best result in the approximate NMS solution in case of training convergence, thus, will not allow FNet to address the failure modes of sequential NMS. In the following we propose a new loss formulation which better approximates mAP. The mAP metric [5] has two significantly different properties compared to regularly used loss functions for end-to-end training. First, it penalizes the presence of multiple hypotheses corresponding to the same ground-truth region.Thus, a model intended to optimize for mAP should allow to explain away sets of hypotheses. Second, mAP is in fact structured over all hypotheses of all images in the test set, meaning that the change in the score obtained by hypothesis h_{ij} of the image I_i could result in different loss signal for hypothesis h_{pq} of image I_p. While there are works aiming to overcome this issue, e.g. [7], our work focuses on the hypotheses filtering. The ideas from [7] can be incorporated into our framework but is beyond the scope of this paper and therefore subject for future work.

Similar to the approximate NMS loss, we aim to generate per-hypothesis labels that will result in an improvement for mAP. In case of mAP, the aforementioned property of positive reinforcement of only the highest scoring hypothesis could be reformulated as the following per-hypothesis label:

$$l(h_i) = \begin{cases} 1, & \text{if } \exists g \in G : \text{IoU}(h_i, g) > a \text{ and } \nexists h_j, i \neq j : s_j > s_i, \text{IoU}(h_j, g) > a; \\ 0, & \text{otherwise.} \end{cases} \tag{13}$$

In other words, hypothesis h_i will get positive label if and only if there is some region g from the set of ground truth regions G that overlaps significantly with hypothesis under consideration, and hypothesis h_i has the maximum score of all hypotheses matching that ground truth region. The network is then trained to directly minimize cross-entropy between scores, output by FNet, and labels according to eq. 13. The difference between regular independent per-hypothesis

Table 1: Results on oMNIST single digit canvas.

digit	network	loss	lenet-feature	mAP
1	LeNet + NMS			0.87
1	+ FNet	nms	score [s]	0.93
1	+ FNet	net	score [s] + fc	**0.97**
3	LeNet + NMS			0.87
3	+ FNet	nms	score [s]	0.90
3	+ FNet	net	score [s] + fc	**0.96**
6	LeNet + NMS			0.87
6	+ FNet	nms	score [s]	0.92
6	+ FNet	net	score [s] + fc	**0.96**

Table 2: Results on oMNIST multi-class digit canvas.

network	loss	lenet-features	mAP
LeNet + NMS			**0.83**
+ FNet	nms	scores [s] + fc	0.81
+ FNet	net	scores [s] + fc	0.78

Table 3: Results on PascalVOC test

method	loss	mAP
FasterRCNN (no filtering)		0.270
+NMS		**0.680**
+FNet	net	0.675

labeling and our network labeling is shown in Figure 3(b). While both methods penalize ill-located hypothesis h_3, the labeling induced by our method also force redundant hypothesis h_2 to be filtered - exactly in the same way it would be treated by mAP evaluation.

6 Experiments

For all considered experiments simple 2-layer neural network with the 512 hidden units and ReLU activations was used to represent both pairwise and context network in network (NiN).

Overlapping MNIST We construct the dataset by randomly placing digits from the MNIST dataset on 128×96 black canvas with background Gaussian noise. In order to create a more realistic dataset we draw a number of digits per canvas uniformly from [0, 24]. For each digit, we draw a location uniformly at random from all valid coordinates of the canvas. An example of a generated image is shown in Figure 2.

We start with the setting when only one class of digits is placed on a canvas, resulting in images with highly overlapping instances of a single class. We experiment with three different digits being placed on a canvas ("1","3","6"). In all of the cases FNet shows a substantial performance gain over baseline NMS approach (Table 1). This suggests that the methods trained on real-world datasets with similar properties, i.e. a large number of overlapping instances of the same class (such as Caltech Pedestrian [4]), could potentially benefit from combining them with our architecture. The results for a multi-digit canvas are shown in Table 2. FNet still achieves comparable performance when trained for approximate NMS objective, while optimization for network loss gives notably worse performance. The issue, though, still could be addressed by proper hyperparameter tuning.

KITTI Similar to the oMNIST experiment we use the scores and features from the last fully-connected layer of a pre-trained network as our per-hypothesis

Table 4: Results on the KITTI benchmark validation set

		Car			Pedestrian		
method	loss	Easy	Mod	Hard	Easy	Mod	Hard
MS-CNN (no filtering)		0.722	0.669	0.540	0.540	0.494	0.442
+ NMS		**0.922**	**0.917**	0.813	**0.896**	**0.867**	0.744
+ FNet	nms	**0.921**	**0.916**	0.813	**0.896**	**0.866**	0.741
+ FNet	net	0.913	0.910	**0.865**	0.890	0.839	**0.746**

features vector $f(h_i)$, in this case MS-CNN [3]. The results for the per class trained FNet on the classes 'Car' and 'Pedestrian' are shown in the Table 4. We omit the results for the class "Cyclist" since we were unable to reproduce the baseline network behaviour.

The FNet results in Table 4 indicate that our proposed approach is indeed expressive enough to be on par with the sequential NMS. Interestingly, using the net-loss discussed in Section 5.2 results in slightly worse performance on the 'Easy' and 'Moderate' data examples, though improvements can be observed in the harder cases of both classes. Notice, all results reported on MS-CNN have been trained as a replacement of sequential NMS on top of MS-CNN and not end-to-end.

PascalVOC2007 For the PascalVOC2007 dataset we use the hypotheses and features from a pre-trained FasterRCNN [3] as our baseline method. The feature vector $f(h_i)$ is again constructed from the scores and the last fully-connected layer of the pre-trained network. We train FNet using the proposed network based labeling (net-loss) with a single multi-class objective. The results in Table 3 show a small performance drop compared to the sequential NMS filtering step. Similar to the KITTI results, no end-to-end training was performed to achieve these results.

7 Conclusion

We have shown an architecture that allows to learn a filtering behaviour based on a potentially varying set of hypotheses per image, while being end-to-end differentiable. We have presented an approximate NMS labeling and shown in experiments on oMNIST and KITTI datasets that our FNet architecture can match the sequential NMS performance by fitting this proxy objective. Further, this network allows to directly optimize an objective that is better aligned with final evaluation metric. We have shown on the synthetic oMNIST example that in case of a large amount of highly overlapping objects of a same class a combination of a flexible filtering and proper loss can result in a notable performance gain.

8 Acknowledgments

The authors would like to thank the anonymous reviewers for their valuable comments and suggestions to improve the quality of the paper. This work was supported by Microsoft Research through its PhD Scholarship Programme.

References

1. Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G.S., Davis, A., Dean, J., Devin, M., et al.: Tensorflow: Large-scale machine learning on heterogeneous distributed systems. arXiv:1603.04467 (2016)
2. Barinova, O., Lempitsky, V., Kholi, P.: On detection of multiple object instances using hough transforms. IEEE Transactions on Pattern Analysis and Machine Intelligence 34(9), 1773–1784 (2012)
3. Cai, Z., Fan, Q., Feris, R.S., Vasconcelos, N.: A unified multi-scale deep convolutional neural network for fast object detection. In: European Conference on Computer Vision. pp. 354–370. Springer (2016)
4. Dollar, P., Wojek, C., Schiele, B., Perona, P.: Pedestrian detection: An evaluation of the state of the art. PAMI 34 (2012)
5. Everingham, M., Van Gool, L., Williams, C.K., Winn, J., Zisserman, A.: The pascal visual object classes (voc) challenge. International journal of computer vision 88(2), 303–338 (2010)
6. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 770–778 (2016)
7. Henderson, P., Ferrari, V.: End-to-end training of object class detectors for mean average precision. arXiv:1607.03476 (2016)
8. Hosang, J., Benenson, R., Schiele, B.: A convnet for non-maximum suppression. In: German Conference on Pattern Recognition. pp. 192–204. Springer (2016)
9. Kontschieder, P., Bulò, S.R., Donoser, M., Pelillo, M., Bischof, H.: Evolutionary hough games for coherent object detection. Computer Vision and Image Understanding 116(11), 1149–1158 (2012)
10. LeCun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. Proceedings of the IEEE 86(11), 2278–2324 (1998)
11. Lin, M., Chen, Q., Yan, S.: Network in network. arXiv preprint arXiv:1312.4400 (2013)
12. Redmon, J., Divvala, S., Girshick, R., Farhadi, A.: You only look once: Unified, real-time object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 779–788 (2016)
13. Redmon, J., Farhadi, A.: Yolo9000: Better, faster, stronger. arXiv:1612.08242 (2016)
14. Ren, S., He, K., Girshick, R., Sun, J.: Faster r-cnn: Towards real-time object detection with region proposal networks. In: Advances in neural information processing systems. pp. 91–99 (2015)
15. Rothe, R., Guillaumin, M., Van Gool, L.: Non-maximum suppression for object detection by passing messages between windows. In: Asian Conference on Computer Vision. pp. 290–306. Springer (2014)
16. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv:1409.1556 (2014)
17. Stewart, R., Andriluka, M., Ng, A.Y.: End-to-end people detection in crowded scenes. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 2325–2333 (2016)
18. Wan, L., Eigen, D., Fergus, R.: End-to-end integration of a convolution network, deformable parts model and non-maximum suppression. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 851–859 (2015)