# **IEEE Copyright Notice**

© 2018 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.

E. Razinkov, I. Saveleva and J. Matas, "ALFA: Agglomerative Late Fusion Algorithm for Object Detection," 2018 24th International Conference on Pattern Recognition (ICPR), Beijing, 2018, pp. 2594-2599.

doi: 10.1109/ICPR.2018.8545182

keywords: convolution; feedforward neural nets;image fusion; object detection; pattern clustering; ALFA; agglomerative **Fusion** algorithm; object detection; agglomerative clustering;object detector predictions; bounding combination;PASCAL box locations; weighted VOC 2007;PASCAL VOC 2012; dynamic belief fusion;single object hypothesis; bounding boxes R-CNN;baseline clustering;SSD;DeNet;Faster combination strategies; DBF; Detectors; Proposals; Object detection; Feature extraction; Convolutional codes; Prediction algorithms; Heuristic algorithms.

URL: http://ieeexplore.ieee.org/stamp/stamp.jsp?tp= &arnumber=8545182&isnumber=8545020

# ALFA: Agglomerative Late Fusion Algorithm for Object Detection

Evgenii Razinkov Institute of Computational Mathematics Institute of Computational Mathematics and Information Technologies Kazan Federal University, Russia Email: Evgenij.Razinkov@kpfu.ru

Iuliia Saveleva and Information Technologies Kazan Federal University, Russia Email: JuOSaveleva@stud.kpfu.ru

Jiři Matas Faculty of Electrical Engineering Czech Technical University in Prague Prague, Czech Republic Email: matas@cmp.felk.cvut.cz

Abstract—We propose ALFA - a novel late fusion algorithm for object detection. ALFA is based on agglomerative clustering of object detector predictions taking into consideration both the bounding box locations and the class scores. Each cluster represents a single object hypothesis whose location is a weighted combination of the clustered bounding boxes.

ALFA was evaluated using combinations of a pair (SSD and DeNet) and a triplet (SSD, DeNet and Faster R-CNN) of recent object detectors that are close to the state-of-the-art. ALFA achieves state of the art results on PASCAL VOC 2007 and PASCAL VOC 2012, outperforming the individual detectors as well as baseline combination strategies, achieving up to 32% lower error than the best individual detectors and up to 6% lower error than the reference fusion algorithm DBF - Dynamic Belief Fusion.

#### I. Introduction

Object detection is an important and challenging task in computer vision with a lot of applications. In recent years accuracy of object detectors significantly improved due to use of learning. R-CNN [6] was the first breakthrough associated with using deep convolutional neural networks for object detection. Fast R-CNN [5] and Faster R-CNN [14] develop the idea further improving both detection speed and accuracy. You Only Look Once [12], Single-Shot Detector [11] introduced more lightweight approach and end-to-end training achieving remarkable accuracy while operating in real-time. YOLOv2 [13] and DSSD [4] and most recent DeNet [16] object detector push these boundaries even further.

It is known that aggregating machine learning algorithms in an ensemble tend to improve performance [1]. Aggregating different object detectors usually called fusion was shown to improve accuracy as well. So called late fusion methods treat base object detectors as black boxes using only their predictions as input.

This paper studies the problem of late fusion on object detector outputs. We want to explore whether modern deep object detectors differ from each other enough so their fusion could show significant performance boost in comparison to individual detectors.

The problem of object detector fusion has been addressed in the literature. Detect2Rank [8] uses Learn2Rank algorithms to rank detections from different object detectors. This goal is achieved by representing each detection with a feature vector

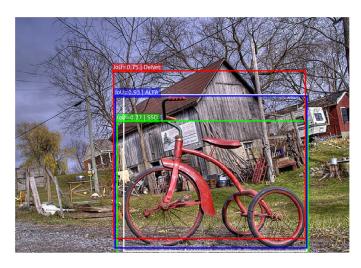


Fig. 1. Image from PASCAL VOC 2007 test set. Bounding boxes and IoU with ground truth: DeNet – red (IoU = 0.75); SSD – green (IoU = 0.77); ALFA – blue (IoU = 0.93). Ground truth bounding box is in white.

and learning a ranking system on a validation set. Handcrafted feature vector includes information about detector-detector context, object saliency and object-object relation information. Ranking is learned using L2 regularized support vector classifier, logistic regressor, support vector regressor and RankSVM. Non-maximum suppression is then used to remove multiple detections of the same object with lower scores.

Dynamic Belief Fusion is a late fusion algorithm that re-scores detections confidence using Dempster-Shafer theory [9]. DBF solves object detection as a binary classification problem - each bounding box either contains an object or not. Using precision-recall curves built for each base detector and for abstract hyperparameter-dependent "best possible detectors" all detections are re-scored and non-maximum suppression is applied. Authors of [9] claim that DBF outperforms all existing fusion methods including Detect2Rank. Unfortunately, the paper does not go into much detail regarding extending binary DBF framework to multiclass object detection scenario.

Faster R-CNN with ResNet-101 as a base network currently acheive state of the art results in object detection in terms of mAP while being quite slow. These object detectors were used as base detectors in an ensemble in [7]. Several Region Proposal Networks and Classification Networks were trained independently, at test time union set of region predictions from all RPNs is classified by an ensemble of classification networks. Non-maximum suppression is run afterwards. Using three models in an ensemble boosts performance from 34.9 mAP to 37.4 mAP on MS COCO object detection dataset [10].

None of these methods provides new bounding boxes as fusion outputs, bounding box for the object is always one of the bounding boxes predicted by base detectors. We believe that combining bounding boxes from all base detectors can lead to better object localization.

#### A. Contribution

Our contribution is as follows:

- We show that modern deep object detectors differ enough so their combination significantly outperforms individual detectors.
- We propose novel Agglomerative Late Fusion Algorithm (ALFA) for object detection that shows state of the art results on PASCAL VOC 2007 [2] and PASCAL VOC 2012 [3] object detection datasets reducing error by up to 32% in comparison with individual detectors and by up to 6% in comparison with the reference fusion methods.
- We clarify DBF extension to multiclass object detection scenario and provide experimental results for modern detectors fusion using DBF.
- We make source code for ALFA and our implementation of DBF publicly available making results of our research reproducible: http://github.com/IuliiaSaveleva/ALFA

## II. BASE OBJECT DETECTORS

Assume N base object detectors  $D_1, D_2, ..., D_N$  for K classes. After processing image I, detector i outputs  $m_i$  predictions of object presence. Each prediction consists of bounding box coordinates and a (K+1)-tuple of class scores:

$$D_i(I) = \{(r_1, c_1), ..., (r_{m_i}, c_{m_i})\}, \quad i = 1, ..., N,$$
 (1)

where  $r_j$  are the four coordinates of the axis-aligned bounding box and  $c_j$  are the class scores for the j-th detected object.

$$r = (x_{tl}, y_{tl}, x_{br}, y_{br}),$$

where  $(x_{tl},y_{tl})$  are the coordinates of the top-left corner of the bounding box and  $(x_{br},y_{br})$  are the coordinates of the bottom-right corner of the bounding box and

$$0 \le x_{tl} < x_{br} < I_{width}, \quad 0 \le y_{tl} < y_{br} < I_{height}.$$

Class scores  $c_i$  is a tuple

$$c_j = \left(c_j^{(0)}, c_j^{(1)}, ..., c_j^{(K)}\right),\,$$

where  $c_j^{(0)}$  is the "no object" probability and  $c_j^{(1)},c_j^{(2)},...,c_j^{(K)}$  are the probabilities for K classes,

$$\sum_{k=0}^{K} c^{(k)} = 1, \quad c^{(k)} \ge 0, \quad k = 0, ..., K.$$

The core problem for fusion of multiple detectors is the decision which detections are due to the same object.

#### III. THE PROPOSED METHOD

We formulate the problem of late fusion as agglomerative clustering and propose a parametrized similarity function that takes into account both spatial properties of the two predictions and their class scores. Parameters are learned on validation set. We define object proposal to be the set of predictions forming a cluster. A proposal therefore comprises one or more predictions.

Assuming that predictions with similar bounding boxes and class scores often correspond to the same object, only one bounding box and class score tuple is output per proposal. We explore several strategies for estimating the object proposal bounding box and classification. We employ non-maximum suppression with IoU threshold 0.5 to remove remaining multiple detections of the same object.

#### A. The Clustering Method

To define clustering procedure one needs to define similarity score function between two samples, similarity score functions between two clusters and stopping criteria.

1) Similarity Between Proposals: We assume that all predictions associated with the object proposal are due to the same object. Hence, each prediction bounding box and class scores should be similar to each other prediction bounding box and class scores. This intuition strongly suggests using complete-link clustering. Let  $C_i$  and  $C_j$  be two clusters and  $\sigma(p,\tilde{p})$  – similarity score function between predictions p and  $\tilde{p}$  that will be defined later. We define the following similarity score function for prediction clusters:

$$\sigma(C_i, C_j) = \min_{p \in C_i, \tilde{p} \in C_j} \sigma(p, \tilde{p}).$$
 (2)

2) Stopping Criteria: Clustering stops when

$$\max_{i,j} \sigma(C_i, C_j) < \tau, \tag{3}$$

where  $\tau$  is a hyperparameter. Therefore, every two predictions associated with the same object proposal have similarity score no less than  $\tau$ .

3) Similarity Between Predictions: We assume that detections with substantially different class scores often correspond to different objects on the image. Therefore, we want to incorporate class scores similarity into our similarity metric. We employ Bhattacharyya coefficient as a measure of similarity between class scores:

$$BC(\bar{c}_i, \bar{c}_j) = \sum_{k=1}^K \sqrt{\bar{c}_i^{(k)} \bar{c}_j^{(k)}},$$
 (4)

where  $\bar{c}$  is obtained from class score tuple c by omitting the zeroth "no object" component and renormalizing:

$$\bar{c}^{(k)} = \frac{c^{(k)}}{1 - c^{(0)}}, \quad k = 1, ...K.$$

Note that  $BC(\bar{c}_i,\bar{c}_j)\in [0,1]$  and equals 1 if and only if  $\bar{c}_i=\bar{c}_j.$ 

Two detections with similar shape, size and position on the image will often correspond to the same object. To decide whether two detections correspond to the same object we should account for similarity between their bounding boxes. We use intersection over union coefficient which is widely used as a measure of similarity between bounding boxes:

$$IoU(r_i, r_j) = \frac{r_i \cap r_j}{r_i \cup r_j}.$$
 (5)

We propose the following measure of similarity between object detectors predictions  $p_i = (r_i, c_i)$  and  $p_j = (r_j, c_j)$  that takes into account both class scores similarity and similarity of bounding boxes:

$$\sigma(p_i, p_j) = IoU(r_i, r_j)^{\gamma} \cdot BC(\bar{c}_i, \bar{c}_j)^{1-\gamma}, \tag{6}$$

where  $\gamma \in [0,1]$  is a hyperparameter. Note that  $\sigma(p_i,p_j) \in [0,1]$ ,  $\sigma(p_i,p_j)$  equals 0 if bounding boxes for predictions  $p_i$  and  $p_j$  do not intersect or their class scores do not overlap. With  $\gamma \in (0,1)$ ,  $\sigma(p_i,p_j)=1$  if and only if bounding boxes and class scores for  $p_i$  and  $p_j$  are equal correspondingly.

Since NMS is applied to every base detector predictions before the fusion procedure, we assume that multiple similar detections from the same base detector are often due to different objects. Therefore, detections from the same base detector should be assigned to different clusters. To achieve that, we set similarity scores between predictions from the same base detector to zero.

#### B. Decision on the class of the cluster

Assume predictions from detectors  $D_{i_1}, D_{i_2}, ..., D_{i_s}$  were assigned to object proposal  $\pi$ . Confidence of the prediction is a score value of the predicted class. Object detector fusion is not straightforward because of variable number of predictions corresponding to each object proposal.

To account for missing detections associated with an object proposal we assign an additional low-confidence class scores tuple to this object proposal for every detector that missed. Low-confidence class scores should not influence class prediction for the object proposal, therefore, last K components of a tuple should be equal:

$$c_{lc}^{(i)} = c_{lc}^{(j)}, \quad i, j = 1, ..., K.$$

Moreover, since recall of modern detectors is less than 1, missed detection does not always mean that there is no object. We use the following low-confidence class scores:

$$c_{lc} = \left(1 - \varepsilon, \frac{\varepsilon}{K}, \frac{\varepsilon}{K}, ..., \frac{\varepsilon}{K}\right), \tag{7}$$

where  $\varepsilon$  is a hyperparameter. Assigning low-confidence class scores to object proposal significantly lowers the confidence of the resulting predictions if  $\varepsilon$  is close to zero.

Possible strategies for dealing with class scores aggregation are inspired by classification ensembles and include:

 Choosing most confident prediction associated with the object proposal:

$$c_{\pi} = c_{i}, \tag{8}$$

where

$$j = \underset{i \in \pi}{\operatorname{argmax}} \max_{1 \le k \le K} c_i^{(k)}. \tag{9}$$

• Averaging fusion. Averaging class scores vectors of associated predictions:

$$c_{\pi}^{(k)} = \frac{1}{N} \left( \sum_{d=1}^{s} c_{i_d}^{(k)} + (N-s) \cdot c_{lc}^{(k)} \right), k = 0, ..., K.$$
(10)

• *Multiplication fusion*. Multiplying class scores corresponding to the same class and renormalizing:

$$c_{\pi}^{(k)} = \frac{\tilde{c}_{\pi}^{(k)}}{\sum_{i} \tilde{c}_{\pi}^{(i)}},\tag{11}$$

where

$$\tilde{c}_{\pi}^{(k)} = \left(c_{lc}^{(k)}\right)^{N-s} \prod_{d=1}^{s} c_{i_d}^{(k)}, \quad k = 0, ..., K.$$
 (12)

#### C. Object localization

We have explored several object localization strategies:

 Bounding box associated with the most confident prediction. Choosing bounding box corresponding to the most confident prediction associated with the object proposal:

$$r_{\pi} = r_{i},\tag{13}$$

where

$$j = \underset{i \in \pi}{\operatorname{argmax}} \max_{1 \le k \le K} c_i^{(k)}. \tag{14}$$

This principle is usually used in non-maximum suppression algorithms. NMS is often applied as a final step in fusion algorithms.

 Averaged bounding box. Averaging outputs of the weak learners is a common way to aggregate predictions to produce output of an ensemble:

$$r_{\pi} = \frac{1}{|\pi|} \sum_{i \in \pi} r_i. \tag{15}$$

• Bounding box obtained by weighted averaging. Correct object localization usually associated with more confident prediction. Following this intuition we assign weights to bounding boxes based on prediction confidence. Fusion outputs the following bounding box for object proposal  $\pi$ :

$$r_{\pi} = \sum_{i \in \pi} \frac{w_i}{\sum_{i \in \pi} w_i} \cdot r_i, \tag{16}$$

where

$$w_i = \max_{1 \le k \le K} (c_i^{(k)}). \tag{17}$$

 Average weighted by proposal class confidence This localization strategy is similar to the previous one except we use class scores of the label that is chosen for the object proposal as weights:

$$r_{\pi} = \frac{1}{\sum_{i \in \pi} c_i^{(l)}} \sum_{i \in \pi} c_i^{(l)} \cdot r_i, \tag{18}$$

where

$$l = \operatorname*{argmax}_{k>1} c_{\pi}^{(k)}. \tag{19}$$

This strategy relies on the assumption that localization of an object depends on the predicted class scores.

#### IV. EXPERIMENTS

We evaluated ALFA and baseline methods on PASCAL VOC 2007 and VOC 2012 object detection datasets using the following base object detectors:

- Single Shot Detector: SSD-300 with VGG-16 as a base network trained on PASCAL VOC 07+12 trainval dataset, i.e. on the union of the training and validation images of the PASCAL VOC 2007 and VOC 2012 datasets.
- DeNet with skip layers and ResNet-101 as a base network trained on PASCAL VOC 07+12 trainval dataset.
- Faster R-CNN with ResNet-101 as a base network trained on PASCAL VOC 07+12 trainval dataset. We do not use global context, multi-scale testing and bounding box refinement introduced in [7].

These object detection methods were selected for three reasons. First, none of them is slow and its combination that requires running all three is not impractical. Second, these are commonly used object detectors with performance no far from the state of the art. Third, for SSD and DeNet the source code and the weights for the object detector are available from the authors.

#### A. Base Detectors

Faster R-CNN [14] uses Region Proposal Network which takes last feature map as an input and outputs a number of bounding boxes that could potentially contain objects. The second component – Fast R-CNN – is used to determine whether patricular proposed bounding box contains an object of certain class or not. While in the original paper VGG [15] is used as base network, it was shown in [7] that using ResNet-101 as base network improves accuracy significantly. Faster R-CNN with ResNet-101 base network with multi-scale testing, inclusion of context and bounding box refinement, sometimes referred to as Faster R-CNN+++, shows state of the art performance in terms of accuracy but is slower than YOLO, SSD and DeNet.

Single Shot Detector [11] does not use RPN but instead relies on convolutional detectors to classify objects on different scales. SSD uses VGG convolutional network with additional convolutional layers for feature extraction. Convolutional detectors for different feature maps predict object classes and offsets with respect to corresponding default bounding boxes. Making use of feature maps at different depths allows SSD to detect objects with different sizes. Neurons from not-too-deep layers of CNN have smaller receptive fields thus are suitable for small objects detection. Neurons at deeper layers have larger receptive fields that allow detection of bigger objects. Convolutional detector consists of (K+1)+4 convolutions with  $3\times3\times d$  filter size predicting class scores

for K classes along with "no object" class and offsets of the bounding box with respect to associated with the detector default bounding box. Non-maximum suppression is applied to all the predictions from convolutional detectors at each scale.

DeNet [16] relies on region-of-interest estimation based on corner detection. DeNet predicts for each image pixel how likely it is a certain corner of an object bounding box. Predicted corners form regions-of-interest that are classified by neural network. Therefore, DeNet uses neither Region Proposal Network nor handcrafted default boxes for object localization. ResNet-101 [7] is used as a base network, deconvolutions are applied to the last layer of ResNet in order to increase localization accuracy. DeNet acheives remarkable performance in terms of speed and accuracy outperforming SSD.

#### B. Baseline methods

1) Dynamic Belief Fusion: We use Dynamic Belief Fusion as a baseline since it was shown to outperform other fusion methods [9]. We use our own implementation of DBF since source code for DBF is not available.

Dynamic Belief Fusion [9] relies on precision-recall curves, thus, binary decision problem is assumed. While [9] clearly states that DBF is applied to object detection on multiclass PASCAL VOC 2007 dataset, only binary procedure is described in the paper and it is not clear how to extend DBF to multiclass scenario. We tested three options how to handle multiclass case in DBF:

- Object detection is treated as a binary problem object is either present in a bounding box or not. In this case each detector has one class-agnostic precisionrecall curve. We do not aggregate bounding boxes with different labels so decision on a final label is trivial.
- 2) Each base detector is considered as K class-specific binary object detectors and every detection with class scores c is treated as K detections with confidence values  $c^{(1)},...,c^{(K)}$ . Each base detector has K class-specific precision-recall curves associated with it. We also assume K class-specific "best possible detectors".
- 3) Each base object detector is again treated as K class-specific binary object detectors with K class-specific precision-recall curves but this time there is only one class-agnostic "best possible detector".

We have implemented all three versions and measured their performance. We use the third variant as a baseline since it outperforms the first and the second option.

2) Non-Maximum Suppression: We use Greedy Non-Maximum Suppression as our second baseline since it was successfully used in [7] to aggregate object detectors outputs. GreedyNMS suppresses every prediction that overlaps with IoU > 0.5 with more confident prediction of the same class.

#### C. Note on confidence thresholds

It is possible to set confidence threshold of any base detector to exclude least confident detections. We can set threshold  $\theta$  for an object detector and consider only those predictions with class scores c for which the following holds:

 $c^{(l)} \ge \theta$ ,

where l is the predicted label.

Value of  $\theta$  for base detectors affects fusion performance. Bigger  $\theta$  results in decreased number of predictions speeding up fusion algorithm since amount of computation is proportional to the number of predictions.

We choose  $\theta$  for DBF and NMS to maximize fusion mAP. For ALFA we use two thresholds chosen to achieve two different goals: (i) maximize fusion mAP and (ii) maximize mAP while keeping fusion computation under 2-3 ms. We refer to this version as "Fast ALFA" in experiment results. Fusion performance is measured on Intel Core i5-6400.

For all tested fusion algorithms smaller values of  $\theta$  result in higher mAP and lower speed performance. We use  $\theta=0.015$  for NMS, DBF and ALFA. We use  $\theta=0.05$  for Fast ALFA.

### D. Note on evaluation procedure

Object detector outputs a class scores vector for every detection. There are two ways of handling obtained class scores vectors:

- Consider every detection with class scores vector c and bounding box r to be a single detection with label l, where  $l = \operatorname{argmax}_{1 \leq k \leq K} c^{(k)}$ , confidence  $c^{(l)}$  and bounding box r. We refer to mean average precision computed this way as mAP-s.
- Treat every detection with class scores vector c and bounding box r as multiple detections sharing the same bounding box one detection for each label l, where l=1,...,K, with confidence value  $c^{(l)}$  and bounding box r.

Since this approach is more common it is referred to as mAP in experiments results.

## E. Results on PASCAL VOC 2007

We employ 5-fold cross-validation on PASCAL VOC 2007 test to choose optimal lozalization and recognition strategies and to adjust hyperparameters  $\gamma$ ,  $\tau$  and  $\varepsilon$ . Best performing localization strategy is averaging weighted by proposal class confidence. Best hyperparameter values obtained through cross-validation are listed in Table I.

Precision-recall curves for DBF are also estimated using 5-fold cross-validation.

During 5-fold cross-validation mean average precision is computed as follows:

- We compute average precision  $AP_i^{(k)}$  on the i-th test part, i=1,...,5, for each class  $k,\ k=1,...,K$ .
- We compute weights for each class for the *i*-th test part:

$$\alpha_i^{(k)} = n_i^{(k)} / n^{(k)},$$

where  $n_i^{(k)}$  is the number of objects of class k in the i-th test part and  $n^{(k)}$  is the number of objects of class k in the whole test set.

• Average precision for each class is computed as follows:

$$AP^{(k)} = \sum_{i} \alpha_i^{(k)} AP_i^{(k)}.$$

• Mean average precision:

$$mAP = \frac{1}{K} \sum_{k=1}^{K} AP^{(k)}.$$

This procedure introduces small bias. To make the comparison fair we use the same mAP computation procedure for all individual detectors and fusion methods on PASCAL VOC 2007

Overall mAP and detection speed are provided in Table II.

# F. Results on PASCAL VOC 2012

We also evaluated all reviewed fusion methods performance on PASCAL VOC 2012 test set. Hyperparameters of ALFA and DBF were adjusted on PASCAL VOC 2007 test set. Results are presented in Table II. Surprisingly, DBF shows worse results than NMS in this scenario. We think that DBF does not generalize well across datasets with different difficulty since it relies on precision-recall curves.

TABLE I
BEST CROSS-VALIDATED HYPERPARAMETER VALUES

	Optimal v	values	Optimal values			
	(SSD + D	DeNet)	(FRCNN + SSD + DeNet)			
Evaluation	mAP-s	mAP	mAP-s	mAP		
Classification confidence	Multiplication	Averaging	Multiplication	Averaging		
τ	0.48	0.73	0.75	0.74		
$\gamma$	0.22	0.25	0.28	0.30		
$\varepsilon$	0.56	0.26	0.17	0.39		

TABLE II RESULTS ON PASCAL VOC 2007 AND VOC 2012

Detector	fps (Hz)	PASCAL	VOC 2007	PASCAL VOC 2012			
		mAP-s	mAP	mAP-s	mAP		
		(%)	(%)	(%)	(%)		
Faster R-CNN	7	77.95	78.83	72.72	73.59		
SSD300	59	79.26	80.37	72.89	74.17		
DeNet	33	78.09	79.26	70.73	72.10		
SSD + DeNet							
NMS	20.3	83.12	83.53	76.80	77.37		
DBF	16.9	83.29	83.88	75.74	76.38		
Fast ALFA	20.6	83.87	84.32	76.97	77.82		
ALFA	18.1	84.16	84.41	77.52	77.98		
SSD + DeNet + Faster R-CNN							
NMS	5.2	84.31	84.43	78.11	78.34		
DBF	4.7	84.97	85.24	75.71	75.69		
Fast ALFA	5.2	85.78	85.67	79.16	79.42		
ALFA	5.0	85.90	85.72	79.41	79.47		

#### G. Ablation study

The influence of the following design decisions is measured:

- Adding low-confidence class scores for every missed detection.
- Aggregating class scores instead of using the most confident prediction. Adding low-confidence detections do not affect performance when using class scores for most confident prediction.
- Taking class scores into account while generating object proposals.
- Using weighted average bounding box instead of using bounding box of the most confident prediction.

Results of the ablation study are summarized in Table III.

TABLE III
EFFECTS OF VARIOUS PARAMETERS ON FUSION PERFORMANCE

	PASCAL VOC 2007 test						
Adding low-confidence class scores				1	1	1	<b>&gt;</b>
Aggregating class scores		1		1	1	1	1
Incorporate class scores into distance metric		1	1			1	<b>\</b>
Aggregating bounding boxes		1	1		1		1
ALFA (FRCNN + SSD + DeNet)	84.25	82.38	84.57	84.79	85.00	85.12	85.72

#### V. DISCUSSION

ALFA shows higher mean average precision values for each detector combination on both PASCAL VOC 2007 and VOC 2012 when compared with base detectors and baseline fusion methods. Fast ALFA is slightly inferior to ALFA but still outperforming baseline fusion methods while being marginally faster

To achieve close-to-real-time performance fusion method needs not only very fast base object detectors but also computationally light aggregation procedure. The key to fast implementation of ALFA is to break all objects into groups so that each object in the group is similar to at least one another object in this group. This step can be done quite effectively. Agglomerative clustering is then applied to each group independently. Computation time for our fusion procedure is as low as 1.2 ms for two detectors and 1.6 ms for three detectors (for  $\theta=0.05$ ) while code is written in Python.

It is reasonable to assume, however, that ALFA performance will be slower for images crowded with objects from the same class with significantly overlapping bounding boxes.

### VI. CONCLUSION

We propose ALFA – a novel late fusion algorithm for object detection. ALFA shows state of the art results on PASCAL VOC 2007 and VOC 2012 datasets outperforming individual detectors and existing fusion frameworks regardless of evaluation procedure while being computationally light. The classification error expressed as 1-mAP, is reduced by up to

32% in comparison to the base detectors and by up to 6% when compared with the state of the art fusion method DBF, that, as experiments indicate, does not generalize well across different datasets.

# ACKNOWLEDGMENT

E. Razinkov was funded by the Russian Government support of the Program of Competitive Growth of Kazan Federal University among World's Leading Academic Centers and by Russian Foundation of Basic Research, project number 16-01-00109a. J. Matas was supported by Czech Science Foundation Project GACR P103/12/G084.

#### REFERENCES

- [1] Dietterich, Thomas G. "Ensemble methods in machine learning." *Multiple classifier systems* 1857 (2000): 1-15.
- [2] Everingham, Mark, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. "The pascal visual object classes (voc) challenge." *International journal of computer vision* 88, no. 2 (2010): 303-338.
- [3] Everingham, Mark, SM Ali Eslami, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. "The pascal visual object classes challenge: A retrospective." *International journal of computer* vision 111, no. 1 (2015): 98-136.
- [4] Fu, Cheng-Yang, Wei Liu, Ananth Ranga, Ambrish Tyagi, and Alexander C. Berg. "DSSD: Deconvolutional Single Shot Detector." arXiv preprint arXiv:1701.06659 (2017).
- [5] Girshick, Ross. "Fast r-cnn." In *Proceedings of the IEEE international conference on computer vision*, pp. 1440-1448. 2015.
- [6] Girshick, Ross, Jeff Donahue, Trevor Darrell, and Jitendra Malik. "Rich feature hierarchies for accurate object detection and semantic segmentation." In *Proceedings of the IEEE conference on computer vision and* pattern recognition, pp. 580-587. 2014.
- [7] He, Kaiming, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. "Deep residual learning for image recognition." In *Proceedings of the IEEE* conference on computer vision and pattern recognition, pp. 770-778. 2016.
- [8] Karaoglu, Sezer, Yang Liu, and Theo Gevers. "Detect2rank: Combining object detectors using learning to rank." *IEEE Transactions on Image Processing* 25, no. 1 (2016): 233-248.
- [9] Lee, Hyungtae, Heesung Kwon, Ryan M. Robinson, William D. Nothwang, and Amar M. Marathe. "Dynamic belief fusion for object detection." In *Applications of Computer Vision (WACV)*, 2016 IEEE Winter Conference on, pp. 1-9. IEEE, 2016.
- [10] Lin, Tsung-Yi, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollr, and C. Lawrence Zitnick. "Microsoft coco: Common objects in context." *In European conference on computer* vision, pp. 740-755. Springer, Cham, 2014.
- [11] Liu, Wei, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C. Berg. "Ssd: Single shot multibox detector." In *European conference on computer vision*, pp. 21-37. Springer, Cham, 2016.
- [12] Redmon, Joseph, Santosh Divvala, Ross Girshick, and Ali Farhadi. "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, Joseph, and Ali Farhadi. "YOLO9000: better, faster, stronger." arXiv preprint arXiv:1612.08242 (2016).
- [14] Ren, Shaoqing, Kaiming He, Ross Girshick, and Jian Sun. "Faster R-CNN: Towards real-time object detection with region proposal networks." In Advances in neural information processing systems, pp. 91-99. 2015.
- [15] Simonyan, Karen, and Andrew Zisserman. "Very deep convolutional networks for large-scale image recognition." arXiv preprint arXiv:1409.1556 (2014).
- [16] Tychsen-Smith, Lachlan, and Lars Petersson. "DeNet: Scalable Realtime Object Detection with Directed Sparse Sampling." arXiv preprint arXiv:1703.10295 (2017).