FISHER-SELECTIVE SEARCH FOR OBJECT DETECTION

Ilker Buzcu[*]

Department of Electrical Engineering
University of California, Los Angeles
Los Angeles, CA, USA

A. Aydin Alatan[†]

Department of Electrical and Electronics Engineering
Center for Image Analysis (OGAM)
Middle East Technical University
Ankara, Turkey

ABSTRACT

An enhancement to one of the existing visual object detection approaches is proposed for generating candidate windows that improves detection accuracy at no additional computational cost. Hypothesis windows for object detection are obtained based on Fisher Vector representations over initially obtained superpixels. In order to obtain new window hypotheses, hierarchical merging of superpixel regions are applied, depending upon improvements on some objectiveness measures with no additional cost due to additivity of Fisher Vectors. The proposed technique is further improved by concatenating these representations with that of deep networks. Based on the results of the simulations on typical data sets, it can be argued that the approach is quite promising for its use of handcrafted features left to dust due to the rise of deep learning.

Index Terms— Visual Object Recognition, Fisher Vectors, Selective Search

1. INTRODUCTION

A paradigm shift occurred in computer vision research in 2012, when AlexNet, a deep convolutional neural network (ConvNet) based entry swept the competition in the ImageNet Large Scale Visual Recognition Challenge (ILSVRC)[1]. Looking at the current state of top tier conferences, it becomes quite hard to argue that anything but results drive the direction of research in this field. The entries to last year's ILSVRC consisted entirely of a roster of deep networks, a trend that does not look to change soon.

Most of the time, a paradigm shift means that decades of research done within the old approaches becomes irrelevant. In this work, we try to salvage some ideas of the past, and combine them with the current state of the art in computer vision research in order to create an ensemble better than both the old and the new paradigms. Specifically, we apply this approach to the problem of object detection in images.

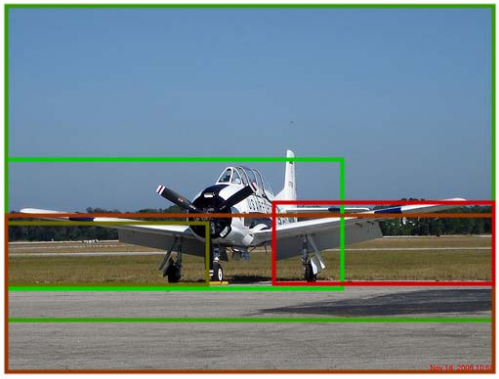

Fig. 1. Airplane detection results with the lite FSS region proposals. The proposals are color-coded in a green-red spectrum, with a higher classification score corresponding to a greener bounding box. Only the top 5 proposals are shown. (Best viewed in color)

Detection of an object, as defined in both PASCAL Visual Object Classes [2] challenges and the ILSVRC [1] competitions, is accomplished by correctly answering two questions:

- Where is the object?
- What is the class of the object?

It is therefore natural that the same techniques used in image classification are a vital part of object detection systems. Currently, many current object detection systems rely on classifiers applied to candidate windows within the image.

2. RELATED WORK

Combinations of deep learning with handcrafted feature extraction methods is an area of research that has been explored in limited ways, with mixed results. A recent approach [3] proposes a classification architecture that makes use of Fisher

[*]This author performed his work at Middle East Technical University.
[†]This work is funded by ASELSAN Research Center.

Vectors[4], where dimensionality reduced Fisher Vectors obtained from images are used to train a fully connected deep neural network. [5] introduces a layered variant of Fisher Vectors in the vein of deep neural networks. In both cases, the improvements obtained by introducing handcrafted elements are overshadowed by the improvements that occur in pure deep ConvNets in a yearly basis. Consequently, such attempts at combining the two paradigms have not found widespread use.

The main problem of relaying the decision making to a classifier is that the number of possible object localizations is exceptionally large for an exhaustive search to be applicable. Typically a few constraints are introduced to reduce this number [6], which results in Sliding Window-type methods, used in [7], [8] among many others.

Smarter approaches also exist to bring the number of candidate windows to a tractable level, without incurring too much computational overhead. Per [9], these can be divided into two groups: *window-scoring* and *grouping* based methods. Window-scoring based approaches reduce the number of proposals via some sort of *objectness* measurement - a measure of how much an image region resembles an object. *Objectness*[10], *BING*[11], *Rahtu*[12] are examples of this family of object proposal generators.

Grouping based methods try to construct object regions, instead of trying to eliminate a large amount of windows from a huge set of candidate regions. *CPMC*[13], *Multiscale Combinatorial Grouping*[14], as well as *Selective Search*[6] belong to this family of proposal generation methods.

2.1. Selective Search

Selective Search(SS) is a proposal generating algorithm that generates proposals from hierarchical, similarity based segmentations [6]. The method uses a hierarchical, bottom-up grouping process to generate regions of varying sizes, starting from an oversegmentation, or superpixels, as described in [15]. Diversification in proposals is achieved in three ways: one way is to run the algorithm in several, complementary color spaces; another is to vary the similarity measurement formula, and the final one is to vary the oversegmentation hyperparameter which changes the initial region map. The similarity measures are made up of a combination of color similarity, texture similarity, size, and the fill measure.

2.2. The Fisher Vector

The Fisher Vector (FV) representation is the state-of-the-art approach to handcrafted representations based on the pooling of local features. The main idea was first published in [4] in the form of the Fisher Kernel, a general way of deriving a discriminative kernel from a generative model of data.

If a Gaussian mixture model is said to describe the local features in all images, the FV of a single feature can be computed as the concatenation of the following closed-form expressions[16]:

$$\Phi_{\mu_{i,j}}(\mathbf{x}) = q_i(\mathbf{x})\frac{x_j - \mu_{i,j}}{\sqrt{w_i}\sigma_{i,j}}, \tag{1a}$$

$$\Phi_{{\sigma^2}_{i,j}}(\mathbf{x}) = \frac{q_i(\mathbf{x})}{\sqrt{2w_i}}[(\frac{x_j - \mu_{i,j}}{{\sigma^2}_{i,j}})^2 - 1]; \tag{1b}$$

where

$$q_i(\mathbf{x}) = \frac{w_i p_i(\mathbf{x}|\mathbf{\Theta_i})}{\sum_{i=1}^{N} w_i p_i(\mathbf{x}|\mathbf{\Theta_i})}, \tag{2}$$

which corresponds to a soft assignment of feature $\mathbf{x}$ to the i^{th} Gaussian with parameters $\mathbf{\Theta}_i = \{\mu_i, diag(\boldsymbol{\sigma^2}_i), w_i\}$.

The FV representation of an image is obtained by average pooling the patch features in the high dimensional FV feature space, with $|\mathbf{X}|$ referring to the number of local features in region $\mathbf{X}$:

$$\Phi(\mathbf{X}) = \frac{1}{|\mathbf{X}|}\sum_{\mathbf{x}\in\mathbf{X}}\Phi(\mathbf{x}). \tag{3}$$

One final note is regarding the *Improved Fisher Vector*(IFV) formulation proposed in [17]. Each element of the IFV, $\Phi^I(X)$ can be computed in terms of the corresponding element of the FV, $\Phi(X)$, as:

$$\Phi_k^I(\mathbf{X}) = sign(\Phi_k(\mathbf{X}))\sqrt{\frac{|\Phi_k(\mathbf{X})|}{\sum_k |\Phi_k(\mathbf{X})|}}. \tag{4}$$

In terms of classification accuracy, the IFV is shown to bring a large improvement over the vanilla FV[17].

3. FISHER-SELECTIVE SEARCH

The Fisher-Selective Search (FSS) algorithm is our proposed extension to the standard Selective Search algorithm. A drawback of many detection frameworks in the literature is that there is no interaction between the proposal generation stage and the classifier stage. If some part of the region classification process can be reused in the region proposal generation part of the framework, not doing so would be a waste of resources. Moreover, since the goals of both stages are not far apart from each other, we should be able to find some smart approach to make the algorithms more similar, so that we can compute the common part of both stages in one go.

An argument against such an approach is the generalization angle: if we take a proposal generating algorithm and infuse it with parts of our specific classification algorithm, it does not suddenly become unusable with any other classification strategy, but in that case it certainly loses its computational advantages. Even so, the fusion algorithm might still be preferred over the original one if it provides enough additional accuracy to justify the trade-off in computation time. Such an approach is similar to using ensembles of classifiers in the sense that it combines more than one classification method to

improve overall accuracy. Ensemble methods are quite prevalent [18] [19]; while they improve accuracy, the improvement is not linear with the number of classifiers to be combined.

In the proposed Fisher-Selective Search, we take advantage of the fact that Fisher encoding is additive in the new feature space. The implication is that in a merging strategy, the Fisher Vector representation of the merger region is the weighted average of FV's of the initial regions, as in Eqn. 5:

$$\hat{\mathbf{\Phi}}_t = \frac{c_i \hat{\mathbf{\Phi}}_i + c_j \hat{\mathbf{\Phi}}_j}{c_i + c_j}, \tag{5}$$

where c_i, c_j are the number of local features contained in regions r^i, r^j. This relation follows easily from the definition of the region FV in Eqn. 3.

After computing the FV's of the initial oversegmentation regions, propagating the FV's throughout the merging process comes at virtually no cost. While the Improved Fisher Vector(IFV) formulation does not have this property, it can be derived from the original FV (with Eqn. 4) at will.

3.1. Fisher-Selective Metrics

The Fisher-Selective Search proposes two new decision metrics to be used as part of the merging strategy: one describing the similarity between two regions in terms of their FV representations and another that tries to construct high scoring regions.

The FV similarity metric has its basis in the fact that distance between two FV's is a good indicator of similarity between the regions they represent. Linear distances in the FV feature space is meaningful in the sense that they are equivalent to computing dissimilarity with the Fisher Kernel. Thus, we propose the FV similarity metric as the inverse of Euclidean distance between two FV's:

$$s_{FVsim}(r^i, r^j) = \frac{1}{1 + \sqrt{\sum_k (\Phi_k(\mathbf{X}_i) - \Phi_k(\mathbf{X}_j))^2}}. \tag{6}$$

Here, $\mathbf{X}_i$ corresponds to the local features contained in r^i, and $\mathbf{\Phi}(\mathbf{X}_i)$ is the *Improved* FV representation of region r^i. This operation maps the L2 norm of the distance vector between two FV's to a value between 1 and 0, which is important since all other Selective Search metrics have the same property which allows us to combine them into more robust similarity metrics appropriately. The function is monotonically decreasing: similarity always decreases as distance increases.

The second metric is a problem-specific one that prioritizes merging of high-scoring regions with some classifier. Since many object recognition problems are multi-class, we propose a strategy of defining the score of a potential merge as the maximum of its scores on all tasks:

$$s_{FVobj}(r^i, r^j) = \frac{1}{1 + \exp[-\max(\mathbf{y}(\mathbf{\Phi}_t))]}, r^t = r^i \cup r^j, \tag{7}$$

where $\mathbf{y}(\mathbf{\Phi}_t)$ is the multiclass classification score vector for the IFV of the merged region. Again, the scores are mapped to between 0 and 1, this time with the monotonically increasing sigmoid function. It should be noted that this metric counts as an objectness metric for a given problem. In a way, the inclusion of this metric bridges the gap between the window scoring-based and grouping-based proposal generation methods.

3.2. Detection With Fisher-Selective Search

The inclusion of FSS-specific metrics defined earlier allows us to skip having a separate classification stage; the outputted bounding boxes come with already computed classification scores. This results in a very compact detection framework. The process as a whole is described in Algorithm 1.

Algorithm 1: The Fisher-Selective Search object detection algorithm.

Input: Image, Local features with locations, Generative model parameters, Classifiers
Output: Set of bounding box-corresponding classification score pairs (B, Y)
Obtain initial regions ([15]) $R = r^1, r^2, ..., r^n$;
for $i \leftarrow 2$ **to** l **do**
  Count and store the number of features in r^i as $C = c^1, c^2, ..., c^n$;
  Compute the FV $\hat{\mathbf{\Phi}}_i$ of r^i;
  Compute the corresponding IFV $\mathbf{\Phi}_i$;
Initialize similarity set $S \leftarrow \emptyset$;
for *each neighboring pair* (r^i, r^j) **do**
  Calculate similarity $s(r^i, r^j)$;
  $S \leftarrow S \cup s(r^i, r^j)$;
while $S \neq \emptyset$ **do**
  Get $s(r^i, r^j) = \max S$;
  Merge $r^t \leftarrow r^i \cup r^j$;
  Compute the new FV $\hat{\mathbf{\Phi}}_t$ using merged region FV's and feature counts;
  Compute the corresponding IFV $\mathbf{\Phi}_i$;
  Remove old similarities:
  $S \leftarrow S - (s(r^i, r^*) \cup s(r^j, r^*))$;
  Calculate new similarities $S_t \leftarrow s(r^t, r^k)$ for each neighbor r_k of r_t;
  $S \leftarrow S \cup S_t, R \leftarrow R \cup r^t$;
Extract bounding boxes B from each region R;
Compute corresponding classification scores Y from FV's.

4. EXPERIMENT RESULTS

In this section, we test the performance of the Fisher-Selective Search algorithm on the VOC2012 detection task, comparing it to the original SS algorithm. We construct two different versions of both methods: one "full" and one "lite" version for each. The full versions try out more merging strategies and work in more color spaces than their lite counterparts; therefore, the full versions are richer in the number of proposals. The differences between each version are given in Table 1, along with the results in terms of average precision(AP), as explained in [2].

All of the classifiers used in these algorithms were trained using SS proposals. Specifically, we use the ground truth bounding boxes as the positive examples, and produce negative examples by running the SS algorithm on the training data and selecting the proposals that have a small overlap with one or more of the ground truth bounding boxes. Improvements can be made to this strategy. For instance, [20] proposes running the first iteration of the classifier through the training images a second time, and adding misclassified regions to the training examples for the final classifier. We stick to a simpler approach, as we care about the relative performance of the methods with the common classifier. We also point out that learning with SS bounding boxes produces a slight bias in favor of the SS methods.

Table 1. Differences between each proposal generation method, and their average precision(AP) in VOC 2012 data. k=desired number of superpixels; H=the resulting number of hierarchical groupings. SS metrics are: C=Color, T=Texture, S=Size, F=Fill; e.g. CTSF is a combination of all 4.

Acronym	Metrics	Color Types	k	H	AP
SS_full	CTSF, TSF, Size, Fill	HSV, Lab, RGI, H, Intensity	100, 200	40	0.137
FSS_full	FVobj, FVsim, CTSF, TSF	HSV, Lab, RGI, H	100, 200	32	0.154
SS_lite	CTSF, TSF	HSV, Lab	50, 150	8	0.092
FSS_lite	FVobj, FVsim	HSV, Lab	50, 150	8	0.144

Comparing the full versions, we see a good amount of improvement even with a slightly reduced number of merging strategies, i.e. hierarchies. Where the FSS algorithm really shines is the comparison of the lite versions: discarding all of the merging strategies used in the original SS algorithm still leaves us with two strong merging strategies, namely the Fisher similarity score and the Fisher objectness metric. The reduction in average precision is very low, and the lite version still beats the full SS algorithm by a healthy margin. By contrast, the lite version of the original SS algorithm falls off quite a bit compared to the full version.

The results show that incorporating a discriminative representation into the merging strategy removes the need for much variation in proposal generation. Instead, we can stick with few strong strategies to construct the proposals and achieve better recall in a reduced number of proposals (Figure 1). The magic of Fisher-Selective Search is in its additive formulation and applicability to arbitrarily shaped regions. These properties allow us to compute the FV for the whole hierarchy in linear time. The FSS computation takes on average only 1.2 seconds more than the 10.2 seconds per image required for SS. We can recoup this additional cost by using the FV scores directly, or we can trade off speed for accuracy and combine the FV scores with state-of-the-art classifiers. We have explored this possibility by training another classifier from the second-to-last layer of the pre-trained deep convolutional neural network CaffeNet [21], as well as an ensemble classifier of the hybrid feature vector, made up of the concatenation of FV and CaffeNet features.

We compare 3 different scenarios: "FSS_full", the leading algorithm of Table 1; "FSS_ConvNet", FSS proposals classified by their CaffeNet representations; and "SS_FVConvNet", SS proposals classified by their hybrid representations. The results are found in Table 2. We see that the introduction of deep convolutional representations to our framework has an immensely positive effect on performance: AP is doubled in both ensemble strategies. Implicitly formulating the information contained in Fisher Vectors in the form of better region proposals seems to be equivalent to an explicit hybrid representation in terms of AP.

Table 2. Comparison of ensembles with VOC2012 data.

Average Precision		
FSS_full	**FSS_ConvNet**	**SS_FVConvNet**
0.154	0.299	0.296

5. CONCLUSION

In this work, we proposed an extension to the widely used Selective Search algorithm, called the Fisher-Selective Search, that showed great promise in experimental results, beating the vanilla Selective Search algorithm by a significant margin. We argued that the extension adds no additional cost to the standard *proposal generation followed by classification* framework, as long as a Fisher Vector classifier is used. We achieved this by leveraging the additive property of the Fisher Vector representation, which is a property not shared by the state of the art deep ConvNet representations. Furthermore, we showed that by using a separate classifier we can improve the accuracy by a large amount at the cost of a small additional computational overhead.

6. REFERENCES

[1] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei, "ImageNet Large Scale Visual Recognition Challenge," p. 43, sep 2014.

[2] Mark Everingham, S. M. Ali Eslami, Luc Van Gool, Christopher K. I. Williams, John Winn, and Andrew Zisserman, "The Pascal Visual Object Classes Challenge: A Retrospective," *International Journal of Computer Vision*, vol. 111, no. 1, pp. 98–136, jun 2014.

[3] Florent Perronnin and Diane Larlus, "Fisher Vectors Meet Neural Networks: A Hybrid Classification Architecture," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 3743–3752.

[4] T.S. Jaakkola and D. Haussler, "Exploiting generative models in discriminative classifiers," *Advances in neural information processing systems*, pp. 487–493, 1999.

[5] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman, "Deep Fisher Networks for Large-Scale Image Classification," in *Advances in Neural Information Processing Systems*, 2013, pp. 163–171.

[6] J. R. R. Uijlings, K. E. A. van de Sande, T. Gevers, and A. W. M. Smeulders, "Selective Search for Object Recognition," *International Journal of Computer Vision*, vol. 104, no. 2, pp. 154–171, 2013.

[7] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*. 2001, vol. 1, pp. I–511–I–518, IEEE Comput. Soc.

[8] Sachin Sudhakar Farfade, Mohammad Saberian, and Li-Jia Li, "Multi-view Face Detection Using Deep Convolutional Neural Networks," *CoRR*, vol. abs/1502.0, 2015.

[9] Jan Hendrik Hosang, Rodrigo Benenson, Piotr Dollár, and Bernt Schiele, "What makes for effective detection proposals?," *CoRR*, vol. abs/1502.0, 2015.

[10] Bogdan Alexe, Thomas Deselaers, and Vittorio Ferrari, "What is an object?," *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 73–80, 2010.

[11] Ming-Ming Cheng, Ziming Zhang, Wen-Yan Lin, and Philip H S Torr, "{BING}: Binarized Normed Gradients for Objectness Estimation at 300fps," in *IEEE CVPR*, 2014.

[12] Esa Rahtu, Juho Kannala, and Matthew Blaschko, "Learning a category independent object detection cascade," in *Proceedings of the IEEE International Conference on Computer Vision*, 2011, pp. 1052–1059.

[13] Joao Carreira and Cristian Sminchisescu, "CPMC: Automatic object segmentation using constrained parametric min-cuts," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 7, pp. 1312–1328, 2012.

[14] Pablo Arbelaez, Jordi Pont-Tuset, Jon Barron, Ferran Marques, and Jitendra Malik, "Multiscale combinatorial grouping," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2014, pp. 328–335.

[15] Pedro F. Felzenszwalb and Daniel P. Huttenlocher, "Efficient graph-based image segmentation," *International Journal of Computer Vision*, vol. 59, no. 2, pp. 167–181, 2004.

[16] Florent Perronnin and Christopher Dance, "Fisher kernels on visual vocabularies for image categorization," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2007.

[17] Florent Perronnin, Jorge Sánchez, and Thomas Mensink, "Improving the Fisher Kernel for Large-scale Image Classification," in *Proceedings of the 11th European Conference on Computer Vision: Part IV*, Berlin, Heidelberg, 2010, ECCV'10, pp. 143–156, Springer-Verlag.

[18] Y Freund and RE Schapire, "A desicion-theoretic generalization of on-line learning and an application to boosting," *Computational learning theory*, vol. 55, no. 1, pp. 119–139, 1995.

[19] Jerome Friedman, Trevor Hastie, and Robert Tibshirani, "Additive logistic regression: A statistical view of boosting," 2000.

[20] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik, "Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation," in *2014 IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 580–587.

[21] Yangqing Jia, "Caffe: An open source convolutional architecture for fast feature embedding," 2013.