# Supplementary Material: HCP: A Flexible CNN Framework for Multi-label Image Classification

Yunchao Wei, Wei Xia, Min Lin, Junshi Huang, Bingbing Ni, Jian Dong, Yao Zhao, *Senior Member, IEEE* Shuicheng Yan, *Senior Member, IEEE*

This document provides detailed justifications of model components for the results used in the paper. Basically, the experimental results are based on VOC 2007.

## 1 Justifications of Image-fine-tuning and Hypotheses-fine-tuning

This justification is based on 10 selected hypotheses generated by BING [2] for each training image.

First, to demonstrate the necessity of image-fine-tuning, we compare with the method which does not utilize image-fine-tuning to initialize the last fully-connected layer. As shown in Table 1, without image-fine-tuning, the performance will drop by 4%. The reason may be explained as follows. With the max pooling operation, our method requires a reasonable prediction for each hypothesis at the beginning so that an advisable hypothesis can be selected to optimize the parameters during the back propagation process. However, without image-fine-tuning, random initialized parameters for the last layer may generate an irrational predicted result for each hypothesis at the beginning. Then, with the cross-patch max pooling, incorrect hypotheses may be selected to optimize the shared CNN, leading to a local optimum for the optimization.

TABLE 1
Comparison of HCP performance in terms of different I-FT settings on VOC 2007 with Alex Net [5] as the shared CNN (mAP in %).

| | Without I-FT | With I-FT |
|---|---|---|
| mAP | 77.5 | 81.5 |

Second, to demonstrate the necessity of hypotheses-fine-tuning, we compare the cross-patch max-pooling result based on the image-fine-tuning model with that based on the hypotheses-fine-tuning model. From Table 2, we can see that the improvement from single image to 500 hypotheses based I-FT is limited, while the classification performance can be significantly boosted with HCP. Therefore, hypotheses-fine-tuning can further improve the classification accuracy.

TABLE 2
Comparison in terms of 500 hypotheses for I-FT model and HCP model on VOC 2007 with Alex Net as the shared CNN (mAP in %).

| | I-FT | HCP |
|---|---|---|
| Single Image | 74.4 | - |
| 500 hypotheses | 75.5 | 81.5 |

## 2 Justification of Loss Function

To validate the effectiveness of the squared loss function utilized in the HCP framework, we compare it with other multi-label loss functions. Four loss functions, i.e., cross-entropy loss [3], hinge loss [1], label-wise cross-entropy loss and pairwise ranking loss [3], are mainly employed to make the comparison. Details of loss functions are shown in the following.

## 2.1 Details of Loss Functions

Denote $X = \{\boldsymbol{x}_i | i = 1, \dots, n\}$ as the image training set. Similar to the notations defined in [3], we denote the deep CNN by $f(\cdot)$ where all the layers filter the given image $\boldsymbol{x}_i$. The $f(\cdot)$ produces a $c$-dimensional output of activations, where $c$ is the number of the predefined labels.

### 2.1.1 Cross-entropy Loss

The softmax function is used to compute the posterior probability of a given image $\boldsymbol{x}_i$ belonging to the class $j$, i.e.

$$p_{ij} = P(j|x_i) = \frac{\exp(f_j(\boldsymbol{x}_i))}{\sum_{k=1}^{c} \exp(f_k(\boldsymbol{x}_i))} \tag{1}$$

where $f_j(x_i)$ is the $j^{th}$ activation value for the image $\boldsymbol{x}_i$. Let $\boldsymbol{y}_i = [y_{i1}, y_{i2}, \cdots, y_{ic}]$ be the label vector of the $i^{th}$ image, where $y_{ij} = 1$ $(j = 1, \cdots, c)$ if the image is annotated by the class $j$, and $y_{ij} = 0$ otherwise. The ground-truth probability vector of the $i^{th}$ image is defined as $\hat{\boldsymbol{p}}_i = \boldsymbol{y}_i / ||\boldsymbol{y}_i||_1$. Given the network prediction in Eqn. ( 1), the loss function is then defined as

$$\begin{aligned} J &= -\frac{1}{n} \sum_{i=1}^{n} \sum_{j=1}^{c} \hat{p}_{ij} \log(p_{ij}) \\ &= -\frac{1}{n} \sum_{i=1}^{n} \sum_{j \in \mathcal{C}_+} \frac{1}{c_{i,+}} \log(p_{ij}) \end{aligned} \tag{2}$$

where $c_{i,+}$ is the number of the ground-truth labels for the $i^{th}$ image, and $\mathcal{C}_+$ denotes the index set of the labeled classes.

### 2.1.2 Pairwise Ranking Loss

This loss function is usually utilized for the annotation problem. Specifically, the target of this loss function is to rank the positive labels to have higher scores than negative labels. The loss function is then defined as

$$J = \frac{1}{n} \sum_{i=1}^{n} \sum_{j \in C_+} \sum_{k \in C_-} \max(0, 1 - f_j(\boldsymbol{x}_i) + f_k(\boldsymbol{x}_i)) \tag{3}$$

where $C_+$ and $C_-$ denote the index sets of labeled and non-labeled classes, respectively.

### 2.1.3 Hinge Loss

The hinge loss is used for maximum-margin classification, which is most notably for support vector machines (SVM). Indicated by [1], the loss function is defined as

$$J = \frac{1}{n} \sum_{i=1}^{n} \sum_{j=1}^{c} \max\left(0, 1 - y_{ij} f_j(\boldsymbol{x}_i)\right)^2 \tag{4}$$

where

$$y_{ij} = \begin{cases} +1 & if\ \boldsymbol{x}_i\ is\ annotated\ with\ the\ j^{th}\ label \\ -1 & otherwise. \end{cases}$$

### 2.1.4 Label-wise Cross-entropy Loss

We present a new loss function for multi-label classification, which is called label-wise cross-entropy (LWCE) loss. The LWCE loss splits the traditional cross-entropy loss into multiple *one vs. all* cross-entropy losses for each class. Specifically, we use the deep CNN as the prediction mapping and change the output dimension of the last fully-connected layer of the network from $c$ to $2c$. The two succeeding dimensions are then fed to the softmax function to compute the cross-entropy loss for each class. Denote $\hat{f}_j(\cdot) \in \mathbb{R}^2$ as the $2-$dimensional output of activations for the $j^{th}$ $(j = 1 \cdots c)$ label. We formulate the posterior probability of the given image $\boldsymbol{x}_i$ belonging to the $j^{th}$ label as

$$p_{ij} = P(j|x_i) = \frac{\exp(\hat{f}_{j1}(\boldsymbol{x}_i))}{\sum_{k=1}^{2} \exp(\hat{f}_{jk}(\boldsymbol{x}_i))} \tag{5}$$

where $\hat{f}_{jk}$ is the $j^{th}$ dimension of $\hat{f}_j$ for $k \in \{1, 2\}$.

Suppose $\boldsymbol{y}_{ij} = [y_{ij1}, y_{ij2}]$ is the 2-dimensional label vector of the $j^{th}$ class, where $y_{ij1} = 1, y_{ij2} = 0$ if the image is annotated with class $j$, and otherwise $y_{ij1} = 0, y_{ij2} = 1$. Then, the loss function to be minimized can be written as

$$\begin{aligned} J &= -\frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{c}\sum_{k=1}^{2} y_{ijk}\log(p_{ijk}) \\ &= -\frac{1}{n}\sum_{i=1}^{n}\left(\sum_{j\in\mathcal{C}_+}\log(p_{ij1}) + \sum_{j\in\mathcal{C}_-}\log(p_{ij2})\right) \end{aligned} \tag{6}$$

where $C_+$ and $C_-$ denote the index sets of labeled and non-labeled classes, respectively.

## 2.2 Results

First, from the experimental results as shown in Table 3, it can be observed that the squared loss and the cross-entropy loss reach better results than other loss functions. The underlying reason may be explained as follows. With the softmax operation across all labels, the context relationship among labels has been implicitly utilized (concurrent or exclusive) in squared loss and cross-entropy loss, which is valuable for multi-label image classification as for VOC datasets. However, for both hinge loss and label-wise cross-entropy loss, the optimization for each class is independent, where the context information of other labels in not taken into consideration.

Second, experimental result based on pairwise ranking loss shows that it is less effective than other loss functions. The target of this loss function is to rank the positive labels with higher predicted scores than negative labels within a particular sample. It however cannot guarantee that positive samples are ranked higher than negative samples for a specific class. Therefore, its performance may be worse.

Third, the experiments show that squared loss is slightly better than cross-entropy loss, and the underlying reason may be as follows. As can be seen from the mathematical definition in Eqn. (2), the cross-entropy loss only aims to promote the predicted score(s) on ground-truth label(s) and does not care about the predicted scores on the negative labels, and thus it is possible that a particular negative label may also have relatively a large value, which may degrade the multi-label classification performance (confidence value is already not too big when the number of positive labels is larger than one). However, squared loss attempts to suppress the predicted scores of all the negative labels to zero. Also minimizing the squared loss mathematically shall encourage the squared losses for all individual negative labels of an image to be similar to each other, namely balanced, which may benefit the classification accuracy since there will be no big confidences for all negative labels.

All these experiments are based on 10 selected hypotheses (generated by BING) for training and 500 hypotheses for testing.

TABLE 3
Comparison of HCP performance in terms of different loss functions on VOC 2007 with Alex Net as the shared CNN (mAP in %).

| Loss Function | mAP |
|---|---|
| Squared Loss | 81.5 |
| Cross-entropy Loss [3] | 80.2 |
| Hinge Loss [1] | 79.9 |
| Label-wise Cross-entropy Loss | 78.9 |
| Pairwise Ranking Loss [3] | 78.4 |

# 3 Justification of Hypothesis Generators

To validate the effectiveness of using object proposal method, two other schemes, i.e., random crops and uniform crops, are introduced for comparison. For the random crops, we estimate a multivariate Gaussian distribution for the bounding box center position, square root area, and log aspect ratio. After calculating mean and covariance on the training set, we sample proposals by following this distribution. For the uniform corps, we uniformly sample the bounding box center position, square root area, and log aspect ratio. These two cropped methods are consistent with those proposed in [4]. We utilize the codes released by [4] to generate the corresponding bounding boxes for each image. Since no objectness confidence score is assigned to each bounding box, we randomly select one hypothesis from each cluster center for training. In addition, to evaluate how other better yet more complex proposal generation methods affect the performance compared with BING, we add new experiments based on another proposal generation method, EdgeBoxes [7], due to its acceptable speed and high quality of proposals. All these experiments are based on 10 selected hypotheses for the training of HCP. Top-500 hypotheses generated by the corresponding generators (i.e., Uniform, Random, BING and Edgeboxes) are used for testing.

Table 4 shows the comparison results on the VOC 2007 *test* set. It can be observed that our HCP framework still works well based on uniform/random crops. However, the results of these two schemes are worse than those proposal-generation schemes, i.e., BING and EdgeBoxes. The underlying reason is that proposal generator can cover or recall hypotheses with more complete and diverse objects compared with random/uniform crops.

TABLE 4
Comparison of HCP performance in terms of different hypotheses selection strategies on VOC 2007 with Alex Net as the shared CNN (mAP in %).

| Hypotheses Selection | mAP |
|---|---|
| Uniform | 80.6 |
| Random | 80.9 |
| BING | 81.5 |
| Edgeboxes | 82.3 |

TABLE 5
Comparison of HCP performance in terms of different number of clusters and proposal methods on VOC 2007 (mAP in %).

| | Alex Net | | VGG Net | |
|---|---|---|---|---|
| | BING | EdgeBoxes | BING | EdgeBoxes |
| 5-clusters | 81.4 | 81.7 | 88.5 | 89.4 |
| 10-clusters | 81.5 | 82.3 | 90.1 | 90.3 |
| 15-clusters | 82.2 | 82.7 | 90.6 | 90.9 |

## 4 Justification of the Number of Clusters

We vary the number of clusters during the HCP training stage to observe its influence based on two proposal methods, i.e., BING [2] and EdgeBoxes [7]. The experiments are addressed based on $n$-clusters ($n$=5,10,15), where $n$ is the number of centers during the clustering operation. The comparison results are shown in Table 5 and all the testing results are based on top-500 hypotheses for each image.

From Table 5, it can be observed that the performance of HCP can be boosted with the increase of the cluster number. This is because more clusters can bring higher coverage rate for objects of training images. In addition, training as well as testing with hypotheses produced by Edgeboxes performs better than that of BING. The main reason is the quality of the hypotheses. As indicated in [4], hypotheses generated by Edgeboxes are better than those produced by BING.

It takes about 1 second to cluster the top-500 bounding boxes generated by BING or EdgeBoxes for training. Based on the top-500 hypotheses, the testing time of Alex Net and VGG Net [6] is about 3s/image and 10s/image on a GPU including the time of proposal generation (BING: 0.2s/image, EdgeBoxes: 0.25s/image).

Details of all experimental justifications on VOC 2007 are show in Table 6.

## References

[1] K. Chatfield, K. Simonyan, A. Vedaldi, and A. Zisserman. Return of the devil in the details: Delving deep into convolutional nets. In *British Machine Vision Conference*, 2014.
[2] M.-M. Cheng, Z. Zhang, W.-Y. Lin, and P. H. S. Torr. BING: Binarized normed gradients for objectness estimation at 300fps. In *Computer Vision and Pattern Recognition*, 2014.
[3] Y. Gong, Y. Jia, T. K. leung, A. Toshev, and S. Ioffe. deep convolutional ranking for multi label image annotation. In *International Conference on Learning Representations*, 2014.
[4] J. Hosang, R. Benenson, P. Dollár, and B. Schiele. What makes for effective detection proposals? *arXiv preprint arXiv:1502.05082*, 2015.
[5] A. Krizhevsky, I. Sutskever, and G. Hinton. Imagenet classification with deep convolutional neural networks. In *Neural Information Processing Systems*, pages 1106–1114, 2012.
[6] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.
[7] C. L. Zitnick and P. Dollár. Edge boxes: Locating object proposals from edges. In *European Conference on Computer Vision*, pages 391–405. Springer, 2014.

TABLE 6
Details of experimental results on VOC 2007.

| | plane | bike | bird | boat | bottle | bus | car | cat | chair | cow | table | dog | horse | motor | person | plant | sheep | sofa | train | tv | mAP |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Justification of loss functions (Training: 10 clusters, Testing: Top-500 hypotheses, Proposal method: BING, Shared CNN: Alex Net) : | | | | | | | | | | | | | | | | | | | | | |
| Squared | 95.1 | 90.1 | 92.8 | 89.9 | 51.5 | 80.0 | 91.7 | 91.6 | 57.7 | 77.8 | 70.9 | 89.3 | 89.3 | 85.2 | 93.0 | 64.0 | 85.7 | 62.7 | 94.4 | 78.3 | 81.5 |
| Cross-entropy | 93.7 | 90.1 | 92.6 | 86.1 | 47.8 | 80.2 | 90.9 | 90.5 | 54.6 | 75.9 | 66.2 | 88.8 | 90.0 | 85.8 | 90.9 | 65.7 | 82.2 | 59.6 | 93.9 | 77.8 | 80.2 |
| Hinge | 91.0 | 88.6 | 88.0 | 85.5 | 53.3 | 79.6 | 92.4 | 87.3 | 55.9 | 69.8 | 71.3 | 85.5 | 90.1 | 86.6 | **97.3** | 63.1 | 78.7 | 61.9 | 91.3 | 79.7 | 79.9 |
| Label-wise | 89.7 | 86.9 | 86.3 | 83.9 | 49.7 | 78.4 | 91.9 | 87.3 | 55.4 | 72.8 | 72.8 | 85.2 | 88.0 | 84.6 | 97.1 | 59.2 | 80.3 | 60.1 | 91.0 | 77.2 | 78.9 |
| Pairwise Ranking | 95.0 | 83.8 | 92.7 | 87.5 | 38.8 | 77.1 | 90.5 | 90.2 | 55.4 | 74.0 | 65.4 | 88.1 | 80.9 | 83.6 | 93.1 | 61.6 | 81.2 | 58.6 | 93.6 | 75.9 | 78.4 |
| Uniform/Random cropping *vs.* proposal methods (Training: 10 clusters, Testing: Top-500 hypotheses, Shared CNN: Alex Net) : | | | | | | | | | | | | | | | | | | | | | |
| Uniform | 95.7 | 90.8 | 90.7 | 88.5 | 48.7 | 78.9 | 90.2 | 90.6 | 57.7 | 73.0 | 74.5 | 88.2 | 87.0 | 83.3 | 90.3 | 64.7 | 82.6 | 64.1 | 94.0 | 77.2 | 80.6 |
| Random | 95.8 | 89.8 | 90.6 | 88.9 | 49.1 | 79.4 | 90.4 | 90.7 | 60.7 | 75.0 | 70.2 | 88.7 | 88.8 | 81.6 | 90.5 | 64.8 | 83.8 | 66.5 | 94.2 | 78.5 | 80.9 |
| BING | 95.1 | 90.1 | 92.8 | 89.9 | 51.5 | 80.0 | 91.7 | 91.6 | 57.7 | 77.8 | 70.9 | 89.3 | 89.3 | 85.2 | 93.0 | 64.0 | 85.7 | 62.7 | 94.4 | 78.3 | 81.5 |
| EdgeBoxes | 95.2 | 91.0 | 93.3 | 88.9 | 55.1 | 80.7 | 91.5 | 91.8 | 60.0 | 77.2 | 72.0 | 89.9 | 90.7 | 86.6 | 92.4 | 66.5 | 84.7 | 65.1 | 94.2 | 79.9 | 82.3 |
| Justification of proposal methods based on 5 clusters (Testing: Top-500 hypotheses): | | | | | | | | | | | | | | | | | | | | | |
| Alex+BING | 94.4 | 88.9 | 92.7 | 88.9 | 53.3 | 81.5 | 91.8 | 91.1 | 59.4 | 76.0 | 71.0 | 88.8 | 89.1 | 84.8 | 92.8 | 64.8 | 84.7 | 62.8 | 93.7 | 76.5 | 81.4 |
| Alex+EdgeBoxes | 94.9 | 90.7 | 92.8 | 87.6 | 52.2 | 80.7 | 91.3 | 91.6 | 61.2 | 74.9 | 72.0 | 89.1 | 90.0 | 86.4 | 92.6 | 63.6 | 84.7 | 63.9 | 93.7 | 79.8 | 81.7 |
| VGG+BING | 98.0 | 95.1 | 96.7 | 94.7 | 68.3 | 93.0 | 94.4 | 97.1 | 69.8 | 87.4 | 77.3 | 95.2 | 94.0 | 91.9 | 96.2 | 77.2 | 93.7 | 67.9 | 97.5 | 85.0 | 88.5 |
| VGG+EdgeBoxes | **99.0** | 95.3 | 97.3 | 95.7 | 71.0 | 91.8 | 94.3 | 97.0 | 70.6 | 88.4 | 80.9 | 95.9 | 95.0 | 92.6 | 95.0 | 73.7 | 93.9 | 75.8 | **98.3** | 86.5 | 89.4 |
| Justification of proposal methods based on 10 clusters (Testing: Top-500 hypotheses): | | | | | | | | | | | | | | | | | | | | | |
| Alex+BING | 95.1 | 90.1 | 92.8 | 89.9 | 51.5 | 80.0 | 91.7 | 91.6 | 57.7 | 77.8 | 70.9 | 89.3 | 89.3 | 85.2 | 93.0 | 64.0 | 85.7 | 62.7 | 94.4 | 78.3 | 81.5 |
| Alex+EdgeBoxes | 95.2 | 91.0 | 93.3 | 88.9 | 55.1 | 80.7 | 91.5 | 91.8 | 60.0 | 77.2 | 72.0 | 89.9 | 90.7 | 86.6 | 92.4 | 66.5 | 84.7 | 65.1 | 94.2 | 79.9 | 82.3 |
| VGG+BING | 98.9 | 96.8 | 97.4 | 96.1 | 72.4 | 93.1 | 95.4 | 97.2 | 69.0 | 90.6 | 78.9 | 96.6 | 95.6 | 93.5 | 96.1 | **79.2** | 94.3 | 74.3 | 98.1 | 88.8 | 90.1 |
| VGG+EdgeBoxes | 98.7 | 96.2 | 97.7 | 95.5 | 72.2 | 93.0 | 95.3 | 97.2 | 70.7 | **91.1** | 81.9 | 96.3 | **96.2** | 93.2 | 95.7 | 79.0 | 94.2 | 75.0 | 97.9 | 89.0 | 90.3 |
| Justification of proposal methods based on 15 clusters (Testing: Top-500 hypotheses): | | | | | | | | | | | | | | | | | | | | | |
| Alex+BING | 95.1 | 89.4 | 93.0 | 89.6 | 52.5 | 82.1 | 91.3 | 91.8 | 60.4 | 77.6 | 73.2 | 90.1 | 88.8 | 85.8 | 92.4 | 67.4 | 85.9 | 64.3 | 94.5 | 78.3 | 82.2 |
| Alex+EdgeBoxes | 95.4 | 90.7 | 92.9 | 88.9 | 53.9 | 81.9 | 91.8 | 92.6 | 60.3 | 79.3 | 73.0 | 90.8 | 89.2 | 86.4 | 92.5 | 66.9 | 86.4 | 65.6 | 94.4 | 80.4 | 82.7 |
| VGG+BING | 98.6 | 95.6 | 97.2 | **96.3** | 74.9 | 94.2 | 95.5 | **97.6** | 72.5 | 89.9 | **82.6** | 96.1 | 95.6 | 93.1 | 96.7 | 78.4 | 93.9 | 74.6 | 97.6 | 90.2 | 90.6 |
| VGG+EdgeBoxes | 98.6 | **97.1** | **98.0** | 95.6 | **75.3** | **94.7** | **95.8** | 97.3 | **73.1** | 90.2 | 80.0 | **97.3** | 96.1 | **94.9** | 96.3 | 78.3 | **94.7** | **76.2** | 97.9 | **91.5** | **90.9** |