

# Generic Object Detection with Dense Neural Patterns and Regionlets

Will Y. Zou<sup>1</sup>

<http://ai.stanford.edu/~wzou>

Xiaoyu Wang<sup>2</sup>

<http://www.xiaoyumu.com>

Miao Sun<sup>3</sup>

<http://vision.ece.missouri.edu/~miao>

Yuanqing Lin<sup>2</sup>

<http://www.linyq.com>

<sup>1</sup> Stanford University

Stanford, CA, 94305

<sup>2</sup> NEC Laboratories America

Cupertino, CA, 95014

<sup>3</sup> University of Missouri

Columbia, MO, 65201

---

## Abstract

This paper addresses the challenge of establishing a bridge between deep convolutional neural networks and conventional object detection frameworks for accurate and efficient generic object detection. We introduce Dense Neural Patterns, short for DNPs, which are dense local features derived from discriminatively trained deep convolutional neural networks. DNPs can be easily plugged into conventional detection frameworks in the same way as other dense local features (like HOG or LBP). The effectiveness of the proposed approach is demonstrated with the Regionlets object detection framework. It achieved 46.1% mean average precision on the PASCAL VOC 2007 dataset, and 44.1% on the PASCAL VOC 2010 dataset, which dramatically improves the original Regionlets approach without DNPs. It is the first approach efficiently applying deep convolutional features for conventional object detection models.

## 1 Introduction

Detecting generic objects in high-resolution images is one of the most valuable pattern recognition tasks, useful for large-scale image labeling, scene understanding, action recognition, self-driving vehicles and robotics. At the same time, accurate detection is a highly challenging task due to cluttered backgrounds, occlusions, and perspective changes. Predominant approaches [5] use deformable template matching with hand-designed features. However, these methods are not flexible when dealing with variable aspect ratios. Wang *et al.* recently proposed a radically different approach, named *Regionlets*, for generic object detection [23]. It extends classic cascaded boosting classifiers [22] with a two-layer feature extraction hierarchy, and is dedicatedly designed for region based object detection. Despite the success of these sophisticated detection methods, the features employed in these frameworks are still traditional features based on low-level cues such as histogram of oriented gradients (HOG) [3], local binary patterns (LBP) [1] or covariance [19] built on image gradients.

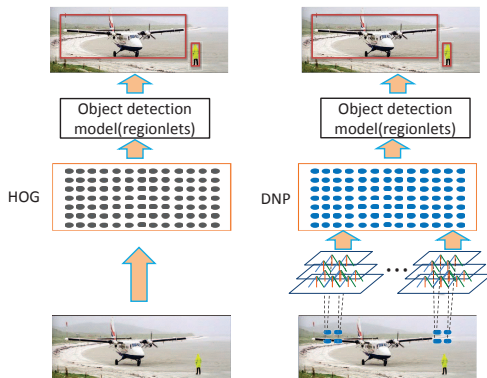


Figure 1: Deep Neural Patterns (DNP) for object detection

With the success in large scale image classification [11], object detection using a deep convolutional neural network also shows promising performance [7, 18]. The dramatic improvements from the application of deep neural networks are believed to be attributable to their capability to learn hierarchically more complex features from large data-sets. Despite their excellent performance, the application of deep CNNs has been centered around image classification, which is computationally expensive when transferred to perform object detection. For example, the approach in [7] requires around 2 minutes to evaluate one image. Furthermore, their formulation does not take advantage of venerable and successful object detection frameworks such as DPM or *Regionlets* which are powerful designs for modeling object deformation, sub-categories and multiple aspect ratios.

These observations motivate us to propose an approach to efficiently incorporate a deep neural network into conventional object detection frameworks. To that end, we introduce the *Dense Neural Pattern* (DNP), a local feature densely extracted from an image with an arbitrary resolution using a deep convolutional neural network trained with image classification datasets. The DNPs not only encode high-level features learned from a large image data-set, but are also local and flexible like other dense local features (like HOG or LBP). It is easy to integrate DNPs into the conventional detection frameworks. More specifically, the receptive field location of a neuron in a deep CNN can be back-tracked to exact coordinates in the image. This implies that spatial information of neural activations is preserved. Activations from the same receptive field but different feature maps can be concatenated to form a feature vector for that receptive field. These feature vectors can be extracted from any convolutional layers before the fully connected layers. Because spatial locations of receptive fields are mixed in fully connected layers, neuron activations from fully connected layers do not encode spatial information. The convolutional layers naturally produce multiple feature vectors that are evenly distributed in the evaluated image crop ( a  $224 \times 224$  crop for example). To obtain dense features for the whole image which may be significantly larger than the network input, we resort to “network-convolution” which shifts the crop location and forward-propagate the neural network until features at all desired locations in the image are extracted. As the result, for a typical PASCAL VOC image, we only need to run the neural network several times to produce DNPs for the whole image depending on the required feature stride, promising low computational cost for feature extraction. To adapt our features for the *Regionlets* framework, we build normalized histograms of DNPs inside each sub-region

of arbitrary resolution within the detection window and add these histograms to the feature pool for the boosting learning process. DNPs can also be easily combined with traditional features in the *Regionlets* framework as explained in Sec. 3.3.

## 2 Review of Related Work

Generic object detection has been improved over years, due to better deformation modeling, more effective multi-viewpoints handling and occlusion handling. Complete survey of the object detection literature is certainly beyond the scope of this paper. Representative works include but not limited to Histogram of Oriented Gradients [3], Deformable Part-based Model and its extensions [5], *Regionlets* [23], etc. This paper aims at incorporating discriminative power of a learned deep CNN into these successful object detection frameworks. The execution of the idea is based on *Regionlets* object detection framework which is currently the state-of-the-art detection approach without using a deep neural network. More details about *Regionlets* are introduced in Sec. 3.3.

Recently, deep learning with CNN has achieved appealing results on image classification [11]. This impressive result is built on prior work on feature learning [8, 14]. The availability of large datasets like ImageNet [4] and high computational power with GPUs has empowered CNNs to learn deep discriminative features. A parallel work of deep learning [12] without using convolution also produced very strong results on the ImageNet classification task. In our approach, we choose the deep CNN architecture due to its unique advantages related to an object detection task as discussed in Sec. 3.1. The most related work to ours is [7] which converts the problem of object detection into region-based image classification using a deep convolutional neural network. Our approach differs in two aspects: 1) We provide a framework to leverage both the discriminative power of a deep CNN and recently developed effective detection models. 2) Our method is 74x faster than [7]. There have been earlier work in applying deep learning to object detection [15]. Among these, most related to ours is the application of unsupervised multi-stage feature learning for object detection [17]. In contrast to their focus on unsupervised pre-training, our work takes advantage of a large-scale supervised image classification model to improve object detection frameworks. The deep CNN is trained using image labels on an image classification task.

## 3 Dense Neural Patterns for Object Detection

In this section, we first introduce the neural network used to extract dense neural patterns, Then we provide detailed description of our dense feature extraction approach. Finally, we illustrate the techniques to integrate DNP with the *Regionlets* object detection framework.

### 3.1 The Deep Convolutional Neural Network for Dense Neural Patterns

Deep neural networks offer a class of hierarchical models to learn features directly from image pixels. Among these models, deep convolutional neural networks (CNN) are constructed assuming locality of spatial dependencies and stationarity of statistics in natural images [11, 13, 16]. The architecture of CNNs gives rise to several unique properties desirable for object detection. Firstly, each neuron in a deep CNN corresponds to a receptive

field [9] whose projected location in the image can be uniquely identified. Thus, the deeper convolutional layers implicitly capture spatial information, which is essential for modeling object part configurations. Secondly, the feature extraction in a deep CNN is performed in a homogeneous way for receptive fields at different locations due to convolutional weighting. More specifically, different receptive fields with the same visual appearance produce the same activations. This is similar to a HOG feature extractor, which produces the same histograms for image patches with the same appearance. Other architectures such as local receptive field networks with untied weights (Le et al., 2012) or fully-connected networks<sup>1</sup> do not have these properties. Not only are these properties valid for a one-layer CNN, they are also valid for a deep CNN with many stacked layers and all dimensions of its feature maps<sup>2</sup>. By virtue of these desirable properties, we employ the deep CNN architecture. We build a CNN with five convolutional layers inter-weaved with max-pooling and contrast normalization layers as illustrated in Figure 2. In contrast with [11], we did not separate the network into two columns, and our network has a slightly larger number of parameters. The deep CNN is trained on large-scale image classification with data from ILSVRC 2010. To train the neural network, we adopt stochastic gradient descent with momentum [14] as the optimization technique, combined with early stopping [6]. To regularize the model, we found it useful to apply data augmentation and the dropout technique [8, 11]. Although the neural network we trained has fully connected layers, we extract DNPs only from convolutional layers since they preserve spatial information from the input image.

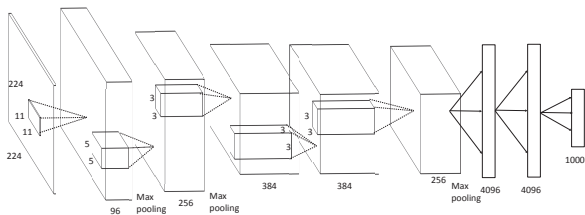


Figure 2: Architecture of the deep convolutional neural network for extracting dense neural patterns.

### 3.2 Dense Neural Patterns

After the deep CNN training on large-scale image classification, the recognition module is employed to produce dense feature maps on high-resolution detection images. We call the combination of this technique and the resulting feature set Dense Neural Patterns (DNPs).

The main idea for extracting dense neural pattern is illustrated in Figure 3 and Figure 4. In the following paragraphs, we first describe the methodologies to extract features using a deep CNN on a single image patch. Then, we describe the geometries involved in applying “*network-convolution*” to generate dense neural patterns for the entire high-resolution image.

Each sub-slice of a deep CNN for visual recognition is commonly composed of a convolutional weight layer, a possible pooling layer, and a possible contrast-normalization layer [10].

<sup>1</sup>Neural networks in which every neurons in the next layer are connected with every neuron on the previous layer

<sup>2</sup>To see this in an intuitive sense, one could apply a “*network-convolution*”, and abstract the stack of locally connected layers as one layer

All three layers could be implemented by convolutional operations. Therefore, seen from the perspective of preserving the spatial feature locations, the combination of these layers could be perceived as one convolutional layer with one abstracted kernel. The spatial location of the output can be traced back by the center point of the convolution kernel.

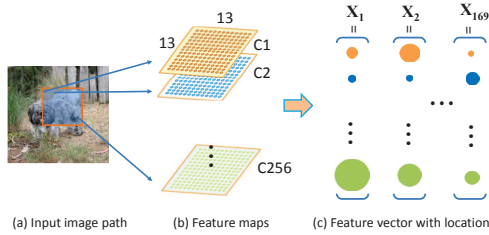


Figure 3: Neural patterns extraction with location association. (a) A square region ( $224 \times 224$ ) as the input for the deep neural network. (b) Feature maps generated by filters in the fifth convolution layer, spatially organized according to their inherited 2-D locations. Each map has  $13 \times 13$  neural patterns. (c) Feature vector generated for each feature point. A bigger circle indicates a larger neural activation.

As shown in Figure 3(b), each convolution kernel produces a sheet of neural patterns. To tailor dense neural patterns into a flexible feature set for object detectors, we compute the 2-D location of each neural pattern and map it back to coordinates on the original image. As an example, we show how to compute the location of the top-left neural pattern in Figure 3(b). The horizontal location  $x$  of this top-left neural pattern feature is computed with Equation 1:

$$x_i = x_{i-1} + \left( \frac{W_i - 1}{2} - P_i \right) S_{i-1} \quad (1)$$

where  $i > 1$ ,  $x_1 = \frac{W_1 - 1}{2}$ ,  $x_{i-1}$  is the top-left location of the previous layer,  $W_i$  is the window size of a convolutional or pooling layer,  $P_i$  is the padding of the current layer,  $S_{i-1}$  is the actual pixel stride of two adjacent neural patterns output by the previous layer which can be computed with Equation 2

$$S_i = S_{i-1} \times s_i. \quad (2)$$

Here  $s_i$  is the current stride using neural patterns output by previous layers as “pixels”. Given equation 1 and equation 2, the pixel locations of neural patterns in different layers can be computed recursively going up the hierarchy. Table 1 shows a range of geometric parameters, including original pixel  $x$  coordinates of the top-left neural pattern and the pixel stride at each layer. Since convolutions are homogeneous in  $x$  and  $y$  directions, the  $y$  coordinates can be computed in a similar manner. Coordinates of the remaining neural patterns can be easily computed by adding a multiple of the stride to the coordinates of the top-left feature point. To obtain a feature vector for a specific spatial location  $(x, y)$ , we concatenate neural patterns located at  $(x, y)$  from all maps(neurons) as illustrated in Figure 3(c).

Now that a feature vector can be computed and localized, dense neural patterns can be obtained by “*network-convolution*”. This process is shown in Figure 4. Producing dense neural patterns to a high-resolution image could be trivial by shifting the deep CNN model with  $224 \times 224$  input over the larger image. However, deeper convolutional networks are usually geometrically constrained. For instance, they require extra padding to ensure the

Table 1: Compute the actual location  $x_i$  of the top-left neural pattern and the actual pixel distance  $S_i$  between two adjacent neural patterns output by layer  $i$ , based on our deep CNN structure.

| $i$ | Layer | $W_i$ | $s_i$ | $P_i$ | $S_i$     | $x_i$ |
|-----|-------|-------|-------|-------|-----------|-------|
| 1   | conv1 | 11    | 4     | 1     | 4         | 6     |
| 2   | pool1 | 3     | 2     | 0     | 8         | 10    |
| 3   | conv2 | 5     | 1     | 2     | 8         | 10    |
| 4   | pool2 | 3     | 2     | 0     | 16        | 18    |
| 5   | conv3 | 3     | 1     | 1     | 16        | 18    |
| 6   | conv4 | 3     | 1     | 1     | 16 <td 18 |       |
| 7   | conv5 | 3     | 1     | 1     | 16        | 18    |
| 8   | pool3 | 3     | 2     | 0     | 32        | 34    |

map sizes and borders work with strides and pooling of the next layer. Therefore, the activation of a neuron on the fifth convolutional layer may have been calculated on zero padded values. This creates the inhomogeneous problem among neural patterns, implying that the same image patch may produce different activations. Although this might cause tolerable inaccuracies for image classification, the problem could be detrimental to object detectors, which is evaluated by localization accuracy. To rectify this concern, we only retain central  $5 \times 5$  feature points of the feature map square.

The DNP feature representation has some desirable characteristics which make it substantially different from and complementary to traditional features used in object detection.

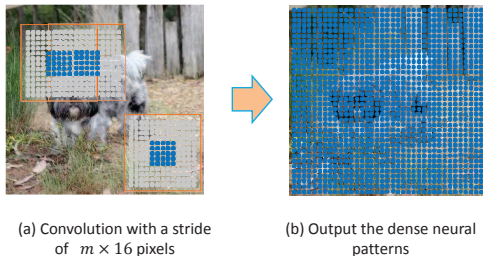


Figure 4: Dense feature maps obtained by shifting the classification window and extract neural patterns at center positions.

### 3.3 Regionlets with Local Histograms of Dense Neural Patterns

The *Regionlets* approach for object detection was recently proposed in [23]. Compared to classical detection methodologies, which apply a object classifier on dense sliding windows [3, 5], the approach employs candidate bounding boxes from Selective Search [20].

The *Regionlets* approach employs boosting classifier cascades as the window classifier. The input to each weak classifier is a one-dimensional feature from an arbitrary region  $R$ . The flexibility of this framework emerges from max-pooling features from several sub-regions inside the region  $R$ . These sub-regions are named *Regionlets*. In the learning process, the most discriminative features are selected by boosting from a large feature pool. It naturally learns deformation handling, one of the challenges in generic object detection. The *Regionlets* approach offers the powerful flexibility to handle different aspect ratios of objects.

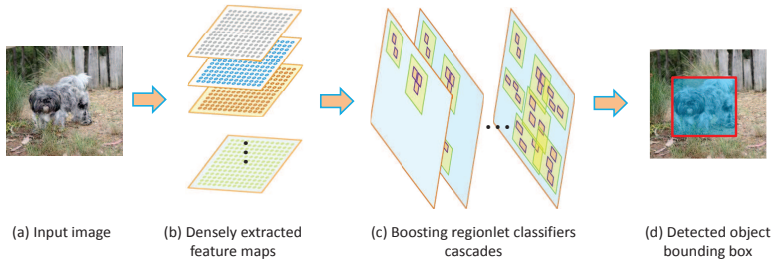


Figure 5: *Regionlets* object detection framework. It learns cascaded boosting classifiers to detect objects of interest. The object searching space is defined using segmentation cues.

The algorithm is able to evaluate any rectangular bounding box. This is because it removes constraints that come with fixed-grid-based feature extraction.

The dense neural patterns introduced in 3.2 encode high-level features from a deep CNN at specific coordinates on the detection image. This makes them a perfect set of features for the *Regionlets* framework. The basic feature construction unit in the *Regionlets* detection model, *i.e.* a regionlet, varies in scales and aspect ratios. At the same time, the deep neural patterns from an image are extracted using a fixed stride which leads to evenly distributed feature points in both horizontal and vertical directions. Thus a regionlet can cover multiple feature points or no feature point. To obtain a fixed length visual representation for a regionlet of arbitrary resolution, we build a local DNP histogram, or average pooling of DNPs, inside each regionlet. Denote DNPs in a regionlet  $r$  as  $\{\mathbf{x}_i | i \in (1, \dots, N_r)\}$ , where  $i$  indicates the index of the feature point,  $N_r$  is the total number of feature points in regionlet  $r$ . The final feature for  $r$  is computed as:

$$\mathbf{x} = \frac{1}{N_r} \sum_{i=1}^{N_r} \mathbf{x}_i. \quad (3)$$

Each dimension of the deep neural patterns corresponds to a histogram bin and their values from different spatial locations are accumulated inside a regionlet. The histograms are normalized using L-0 norm. While most histogram features define a fixed spatial resolution for feature extraction, our definition allows for a histogram over a region of arbitrary shape and size. Following [23], max-pooling is performed among regionlets to handle local deformations.

To incorporate DNP into the *Regionlets* detector learning framework, in which the weak learner is based on a 1-D feature, we uniformly sample the  $DNP \times Regionlets$  configuration space to construct the weak classifier pool. Each configuration specifies the spatial configuration of *Regionlets* as well as the feature dimension of *DNP*. Because the representation is 1-D, the generated feature pool can be easily augmented to the pool of other features such as HOG, LBP or Covariance.

Constructing DNP feature representations for other template-based detectors (similar as HOG template) is fairly simple. Naturally we just need to concatenate all DNPs in the detection window. The features can also be directly applied to the Deformable Part-based Model by replacing the HOG features with the 256 dimensional neural patterns.

Table 2: Detection results using traditional feature and Deep Neural Patterns on PASCAL VOC 2007. The combination of traditional features and DNP shows significant improvement.

| Features                                 | Mean AP     |
|--|-------------|
| DNP Layer 1                              | 24.9        |
| DNP Layer 2                              | 33.5        |
| LBP                                      | 33.5        |
| Covariance                               | 33.7        |
| DNP Layer 3                              | 34.5        |
| HOG                                      | 35.1        |
| DNP Layer 4                              | 38.9        |
| DNP Layer 5                              | 40.2        |
| HOG, LBP, Covariance                     | 41.7        |
| <b>HOG, LBP, Covariance, DNP Layer 5</b> | <b>46.1</b> |

Table 3: Performance comparison between two feature combination strategies: 1) Combination of neural patterns from the fifth layer and neural patterns from a shallow layer(second layer). 2) Combination of neural patterns from the fifth layer and hand-crafted low-level features.

| Features                           | Mean AP |
|------------------------------------|---------|
| DNP Layer 5                        | 40.2%   |
| DNP Layer 5 + Layer 2              | 40.4%   |
| DNP Layer 5 + HOG, LBP, Covariance | 46.1%   |

## 4 Experiments

To validate our method, we conduct experiments on the PASCAL VOC 2007 and VOC 2010 object detection benchmark datasets. PASCAL VOC datasets contain 20 categories of objects. The performance is measured by mean average precision (mAP) over all classes. In the following paragraphs, we describe the experimental set-up, results and analysis.

We train a deep neural network with five convolutional layers and three fully connected layers on 1.2 million images in ILSVRC 2010. All input images are center-cropped and resized to  $256 \times 256$  pixels. We augment the data with image distortions based on translations and PCA on color channels. The deep CNN reached 59% top 1 accuracy on the ILSVRC 2010 test set. While our aim is to demonstrate the effectiveness of DNPs in object detection, a deep CNN with better performance is likely to further improve the detection accuracy.

The original *Regionlets* [23] approach utilizes three different features, HOG, LBP and covariance. In our experiments, we add to the feature pool DNP features from different layers. During cascade training, 100 million candidate weak classifiers are generated from which we sample 20K weak classifiers. On each test image, we form proposed object hypothesis as [20] and pass them along the cascaded classifiers to obtain final detection result.

### 4.1 Detection Performance

We firstly evaluate how the deep neural patterns alone perform with the *Regionlets* framework, followed with evaluation of the combination of DNP and HOG, LBP, Covariance features. Finally, we compare our method with other state-of-the-art approaches.

Table 2 summarizes the performance(sorted in ascending order) of traditional features,



DNP and their combinations on PASCAL VOC 2007. It is interesting that DNPs from the second layer and third layer have comparable performance with the well engineered features such as HOG, LBP and Covariance features. DNPs from the fifth layer outperforms any single features, and are comparable to the combination of all the other three features. The most exciting fact is that DNPs and hand-designed features are highly complementary. Their combination boosts the mean average precision to 46.1%, outperforming the original Regionlets approach by 4.4%. Note that we did not apply any fine-tuning of the neural network on the PASCAL dataset.

The combination of DNPs and hand-crafted low-level features significantly improves the detection performance. To determine whether the same synergy can be obtained by combining low-level and high-level DNPs, we combine the DNPs from the fifth convolutional layer and the second convolutional layer. The performance is shown in Table 3. However, the combination only performs slightly better (0.2%) than using the fifth layer only. This may be because the fifth layer features are learned from the lower level which makes these two layer features less complementary.

Table 4: Detection results(mean AP%) on PASCAL VOC 2007 and VOC 2010 datasets.

|                              | VOC 2007    | VOC2010     |
|------------------------------|-------------|-------------|
| DPM [5]                      | 33.7        | 29.6        |
| SS_SPM [20]                  | 33.8        | 34.1        |
| Objectness [2]               | 27.4        | N/A         |
| BOW [21]                     | 32.1        | N/A         |
| Regionlets [23]              | 41.7        | 39.7        |
| R-CNN pool <sub>5</sub> [7]  | 40.1        | N/A         |
| R-CNN FT fc <sub>7</sub> [7] | <b>48.0</b> | 43.5        |
| DNP+Regionlets               | 46.1        | <b>44.1</b> |

Table 5: Speed comparison with directly extracting CNN features for object candidates [7].

|                                   | R-CNN pool <sub>5</sub> | Ours         |
|-----------------------------------|-------------------------|--------------|
| Resize object candidate regions   | Yes                     | No           |
| Number of model convolutions      | ~ 2213                  | ~ 30         |
| Feature extraction time per image | 121.49s                 | <b>1.64s</b> |

Table 4 shows detection performance comparison with other detection methods on PASCAL VOC 2007 and VOC 2010 datasets. We achieved 46.1% and 44.1% mean average precision on these two datasets which are comparable with or better than the current state of the art by [7]. Here we compare to results with two different settings in [7]: features from the fifth convolutional layer after pooling, features from the seventh fully connected layer with fine-tuning on the PASCAL datasets. The first setting is similar to us except that features are pooled. Our results are better(46.1% vs 40.1% on VOC 2007) than [7] on both datasets in this setting. The approach in [7] requires resizing a candidate region and apply the deep CNN thousands of times to extract features from all candidate regions in an image. The complexity of our method is independent of the number of candidate regions which makes it orders of magnitude faster. Table 5 shows the comparison with [7] in terms of speed using the first setting.<sup>3</sup> The experiment is performed by calculating the average time across processing all images in the PASCAL VOC 2007 dataset. DNPs extraction takes 1.64 seconds

<sup>3</sup>The time cost of the second setting in [7] is higher because of the computation in fully connected layer.

per image while [7] requires 2 minutes.

## 4.2 Visual Analysis

We devise a visualization techniques for the most important features used by the detector. The learning process for boosting selects discriminative weak classifiers. The importance of a feature dimension roughly corresponds to how frequently it is selected during training. We count the occurrence of each dimension of the DNPs in the final weak classifier set and determine the most frequent dimension. We retrieve image crops from the dataset which give the highest responses to the corresponding neurons in the deep CNN.

Figure 6 shows the visualization. The left column describes the object category we want to detect. Right columns show visual patches which give high responses to the most frequently selected neural pattern dimension for the category. They are obviously quite correlated. It indicates that the selected neural patterns encode part-level or object-level visual features highly correlated with the object category.

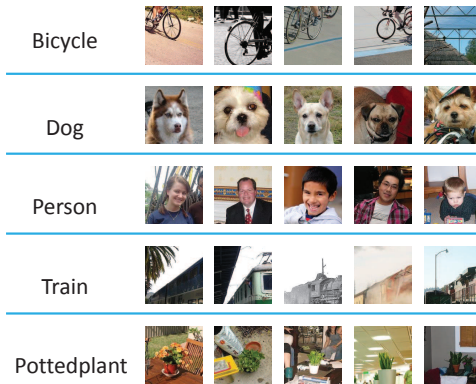


Figure 6: Visualization of the high-level information encoded by neural patterns from the fifth convolutional layer.

## 5 Conclusion

In this paper, we present a novel framework to incorporate a discriminatively trained deep convolutional neural network into generic object detection. It is a fast effective way to enhance existing conventional detection approaches with the power of a deep CNN. Instantiated with *Regionlets* detection framework, we demonstrated the effectiveness of the proposed approach on public benchmarks. We achieved comparable performance to state-of-the-art with 74 times faster speed on PASCAL VOC datasets. We also show that the DNPs are complementary to traditional features used in object detection. Their combination significantly boosts the performance of each individual feature.

## Acknowledgments

This work was done during the internship of the first author at NEC Laboratories America in Cupertino, CA.

## References

- [1] Timo Ahonen, Abdenour Hadid, and Matti Pietikäinen. Face description with local binary patterns: Application to face recognition. *T-PAMI*, 2006.
- [2] Bogdan Alexe, Thomas Deselaers, and Vittorio Ferrari. Measuring the objectness of image windows. *T-PAMI*, 2012.
- [3] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *CVPR*, 2005.
- [4] Jia Deng, Alexander C. Berg, Kai. Li, and Li Fei-Fei. What does classifying more than 10,000 image categories tell us? In *ECCV*, 2010.
- [5] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part based models. *T-PAMI*, 2010.
- [6] Federico Girosi, Michael Jones, and Tomaso Poggio. Regularization theory and neural networks architectures. *Neural Computation*, 1995.
- [7] Ross B. Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. *CVPR*, 2014.
- [8] Geoffrey E Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, and Ruslan R Salakhutdinov. Improving neural networks by preventing co-adaptation of feature detectors. *arXiv:1207.0580*, 2012.
- [9] David H Hubel and Torsten N Wiesel. Receptive fields and functional architecture of monkey striate cortex. *The Journal of Physiology*, 1968.
- [10] Kevin Jarrett, Koray Kavukcuoglu, Marc Aurelio Ranzato, and Yann LeCun. What is the best multi-stage architecture for object recognition? In *ICCV*, 2009.
- [11] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, 2012.
- [12] Q. V. Le, M. A. Ranzato, R Monga, M Devin, K Chen, G. D. Corrado, J Dean, and A. Y. Ng. Building high-level features using large scale unsupervised learning. In *ICML*, 2012.
- [13] Yann LeCun and Yoshua Bengio. Convolutional networks for images, speech, and time series. *The handbook of brain theory and neural networks*, 3361, 1995.
- [14] Yann LeCun, Léon Bottou, Genevieve B Orr, and Klaus-Robert Müller. Efficient backprop. In *Neural networks: Tricks of the trade*, pages 9–50. Springer, 1998.
- [15] Yann LeCun, Fu Jie Huang, and Leon Bottou. Learning methods for generic object recognition with invariance to pose and lighting. In *CVPR*, 2004.
- [16] M Ranzato and Yann LeCun. A sparse and locally shift invariant feature extractor applied to document images. In *ICDAR*, 2007.
- [17] Pierre Sermanet, Koray Kavukcuoglu, and Soumith Chintala. Pedestrian detection with unsupervised multi-stage feature learning. In *CVPR*, 2012.
- [18] Pierre Sermanet, David Eigen, Xiang Zhang, Michaël Mathieu, Rob Fergus, and Yann LeCun. Overfeat: Integrated recognition, localization and detection using convolutional networks. *CoRR*, abs/1312.6229, 2013.
- [19] Oncel Tuzel, Fatih Porikli, and Peter Meer. Pedestrian detection via classification on riemannian manifolds. *T-PAMI*, 2008.
- [20] Koen EA Van de Sande, Jasper RR Uijlings, Theo Gevers, and Arnold WM Smeulders. Segmentation as selective search for object recognition. In *ICCV*, 2011.
- [21] Andrea Vedaldi, Varun Gulshan, Manik Varma, and Andrew Zisserman. Multiple kernels for object detection. In *ICCV*, 2009.
- [22] P. Viola and M. J. Jones. Robust real-time object detection. *IJCV*, 2001.
- [23] Xiaoyu Wang, Ming Yang, Shenghuo Zhu, and Yuanqing Lin. Regionlets for generic object detection. In *ICCV*, 2013.