Semi-supervised Learning of Feature Hierarchies for Object Detection in a Video

Yang Yang Guang Shu Mubarak Shah
Center for Research in Computer Vision, University of Central Florida
yyang@cs.ucf.edu, gshu@eecs.ucf.edu, shah@crcv.ucf.edu

Abstract

We propose a novel approach to boost the performance of generic object detectors on videos by learning video-specific features using a deep neural network. The insight behind our proposed approach is that an object appearing in different frames of a video clip should share similar features, which can be learned to build better detectors. Unlike many supervised detector adaptation or detection-by-tracking methods, our method does not require any extra annotations or utilize temporal correspondence. We start with the high-confidence detections from a generic detector, then iteratively learn new video-specific features and refine the detection scores. In order to learn discriminative and compact features, we propose a new feature learning method using a deep neural network based on auto encoders. It differs from the existing unsupervised feature learning methods in two ways: first it optimizes both discriminative and generative properties of the features simultaneously, which gives our features better discriminative ability; second, our learned features are more compact, while the unsupervised feature learning methods usually learn a redundant set of over-complete features. Extensive experimental results on person and horse detection show that significant performance improvement can be achieved with our proposed method.

1. Introduction

Object detection has been explored extensively and has achieved significant success in the past decade [5, 6, 8, 27, 29, 30]. Most of the state-of-the-art detectors are designed for a single static image and are trained from a large set of labeled examples. The performance of a detector will be inevitably degraded when it is applied to frames in a video taken under conditions which are very different from those of the training examples. Because of the large variation across different environments, a generic classifier trained on extensive datasets may perform sub-optimally in a particular test environment. In general, the construction of appearance-based object detector is time-consuming and difficult because a large number of training examples must be collected and manually labeled in order to capture different variations in object appearance. Therefore, how to adapt a learned generic detector to the images in a specific video taken under different visual conditions becomes a very important problem to be explored.

A large amount of work [2, 34, 9, 10, 14, 20, 24, 23, 18, 24] has been reported on improving object detection in video frames. Several authors [2, 9, 10] propose to improve detection and tracking simultaneously through detection by tracking and vice versa. The detection results serve as a cue to build the tracking results, and the detection component maybe further improved by the result of trackers through online learning. But the improvement will be heavily downgraded if we directly use the noisy detections as initialization of the trackers. Besides detection-by-tracking, researchers have also devoted their efforts on developing online learning/adaptation algorithms for detectors [14, 34, 20, 24, 22, 18, 23]. However, online retraining of the detector is usually hard due to the less training samples and expensive due to the model complexity.

In this paper, we propose to improve the detection results of a generic detector on a video by refining the detection scores in an offline fashion, without requiring any trajectory information nor annotation from the video. To achieve this, the original detector with a low detection threshold setting is first applied to the frames in the target video. All detected visual examples are collected to form the candidate detection pools using both positive and negative examples pertaining to the target video.

Since selections of the right features plays an important role in object detection, we argue that, the classical hand-crafted features, such as HOG, SIFT, may not be universally suitable and discriminative enough to every type of video. In a particular video, the way objects appear would share some similar properties which could be leveraged to distinguish them from the non-objects. Hence, unlike other proposed methods, which are built on using hand-designed features, we learn the good features directly from the raw pixels of the video itself.

In order to learn discriminative and compact features, we propose a new feature learning method using a deep neural network based on auto encoders. It differs from the existing unsupervised feature learning methods in two ways: first it optimizes both discriminative and generative properties of the features simultaneously, which gives our features better discriminative ability; second, our learned features are more compact, while the unsupervised feature learning methods usually learn a redundant set of over-complete features. Moreover, we learn a discriminative feature hierarchy from local patches to global images. Extensive experiments with qualitative and quantitative results demonstrate the efficacy of our approach.

The rest of the paper is organized as follows. We first review the related literature in section 2. The proposed method is presented in section 3 in this order: Preliminary, generative feature learning, discriminative feature learning and learning higher levels. Extensive experiment results, comparisons and analysis are reported in section 4. Finally, we conclude in section 5 with a brief discussion on future work.

2. Related Work

Several works have been proposed to detect objects in videos. They can be divided into two categories. One is detection by tracking [2, 9, 10], which use the trajectory information to help improving detection results and the improved detection can be used backward to improve tracking. Another is detection by detection [14, 34, 20, 24, 22, 18, 23, 31] and most methods in this category treat this as a semi-supervised problem and try to propagate the label to new examples correctly. Authors in [31] used HOG feature with tree coding in a non-parametric detector adaptation method. Javed et.al [14] proposed a co-training based approach using color and edges as the feature representation. Authors in [18] trained two disparate classifiers simultaneously by carefully choosing independent or complementary hand designed features.

Most of the successful methods above heavily depend on choosing the correct low-level features, such as SIFT, HOG and color histogram. Notice that the object appearance in video frames should share some regularities among each other, which could be used for discriminative classification. Our work intend to learn the good features directly from the raw pixels of a video.

Feature learning, which finds concise, slightly higher-level representations of inputs, has has been successfully applied to object recognition and scene recognition. Most of the methods [4, 17, 16, 12, 15, 33, 26, 32, 11] are unsupervised learning algorithms. The goal is to use unlabeled data to help in a supervised learning task, even if the unlabeled data cannot be associated with the labels of the supervised task. However, in our case since we have the label information(the confidence scores from the detector), and since examples from video frames are highly correlated , we would like to use the label information to directly learn a more discriminative feature set for better classification. Therefore, we need to learn the features generatively and discriminatively. Several methods [35, 21, 19] have shown significant improved results with discriminative features. The authors in [19] used sparse coding to learn multiple dictionaries for each category. [21] proposed to learn a semi-supervised method on top of bag-of-words representation for document recognition. The authors in [35] proposed a single level hybrid learning method for incremental feature learning. In this paper, we propose a new feature learning method using a deep neural network based on auto encoders with invariance design. We learn three levels of discriminative features from local to global by optimizing both discriminative and generative properties of the features simultaneously.

3. The Model

We formulate our problem as a semi-supervised classification problem. First we apply an original detector on a video to get a substantial amount of candidate detections for rescoring. Those detections are initially labeled as confident-positive, confident-negative or hard examples by their confidences. Later we use the confident-positive and confident-negative examples to learn video-specific features. Then, we re-score the hard examples by training a classifier using the learned features. After the rescoring, a small number of hard examples with high confidence are moved into the confident-positive or confident negative sets for next iteration of feature learning. We repeat the above steps until no hard samples become confident ones. In our experiments, it usually converges in 4-7 iterations. The flow chart of the framework is illustrated in figure 1.

To learn a set of representative features, we propose to use both a supervised and an unsupervised objective based on auto-encoders[28]. We require the representation to be generative, which can produce good reconstructions of the input images, at the same time, to be discriminative, which can give good predictions of the image class labels.

Further, we learn the feature hierarchies from local to global by increasing the receptive field size (the 2D patch size). Our aim is to capture the local features such as edges with different orientations or color, as well as the global characteristic like the structure and shape. To do so, we stack the the auto-encoders to form a deep network.

In the following part of this section, we will start with introducing the preliminaries about auto-encoders, then move to unsupervised generative feature learning using auto-encoders and discriminative feature learning. Finally, we will describe how to learn higher level features.

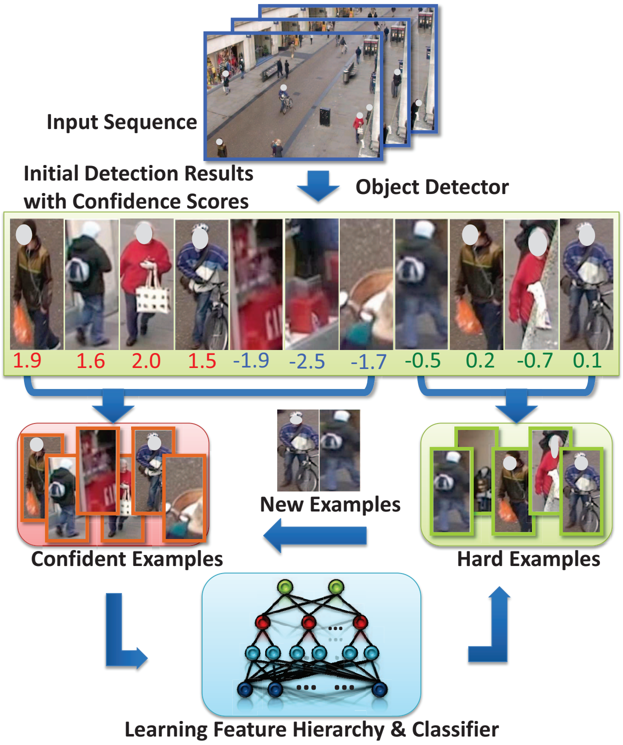

Figure 1: The flow chart of our proposed method for video object detection. The confident (positive and negative) and hard examples are first collected based on the confidence scores given by the object detector. Then the feature hierarchies and classifiers are learned from the confident examples and used for re-scoring the hard samples. The hard samples with high confidence scores are included into the confident examples iteratively until no hard examples become confident ones.[Best viewed in color]

3.1. Preliminaries: Auto-encoders

We start by describing the algorithm for our basic learning module, based on the auto-encoders[28], an unsupervised learning architecture used to pre-train deep networks. Suppose we have N randomly sampled local patches $x^{(i)} \in \mathrm{R}^D$ from the training set (the dark blue nodes in figure 2), to learn features from them, the conventional auto-encoders attempts to reconstruct the data by minimizing the following loss function:

$$E_{AE} = \sum_{i=1}^{N} \|x^{(i)} - W_2 s(W_1 x^{(i)} + b_1) + b_2\|^2 \quad (1)$$
$$+ Z(W_1 x^{(i)} + b_1),$$

where $W_1 \in \mathrm{R}^{N_1 \times D}$ is a weight matrix which maps the visible nodes to hidden nodes, $b_1 \in \mathrm{R}^{N_1}$ is a hidden bias vector, and $s(x) = \frac{1}{1+exp(-x)}$ is a non-linear sigmoid function. $W_2 \in \mathrm{R}^{D \times N_1}$ is a weight matrix which reconstructs the visible node from the hidden node, $b_2 \in \mathrm{R}^D$ is an input bias vector. Z is a regularization function. To simplify the formulation, we use linear activation($s(x) = x$), no biases and tied weights ($W = W_1 = W_2^T$). Hence, the cost function of auto-encoders can be simplified as:

$$E_{AE} = \sum_{i=1}^{N} \|x^{(i)} - W^T h(i)\|^2 + Z(h^{(i)}). \quad (2)$$

where, we let $h^{(i)} = W x^{(i)}$ as the light blue nodes in the neural network shown in figure 2.

3.2. Generative Feature Learning

A generative objective function measures an average reconstruction error between the input x and the reconstruction $x' = W^T W x$. The reason is that if the model achieves a good reconstruction from the code, then we can be sure that the representation has preserved most of the information from x. To make the learned features invariant to local transformation, we further impose a second layer on the top of the auto-encoders by hard coded weights V which pools from several adjacent neurons h, as shown in figure 2 the red nodes. The regularization function Z is set to enforce the activation of the second layer to be sparse. Hence, the loss function with the second layer pooling unit for unsupervised generative feature learning is as below:

$$E_{gen} = \sum_{i=1}^{N} \|x^{(i)} - W^T h^{(i)}\|_2^2 + \lambda \sum_{i=1}^{N} \|\sqrt{V(h^{(i)})^2}\|_1. \quad (3)$$

If we let the second layers activation $p^{(i)} = \sqrt{V(h^{(i)})^2}$, then equation 3 can be written as:

$$E_{gen} = \sum_{i=1}^{N} \|x^{(i)} - W^T h^{(i)}\|_2^2 + \lambda \sum_{i=1}^{N} \|p^{(i)}\|_1. \quad (4)$$

In this equation, the index i denotes data samples. Square and square-root operations are element-wise here. V is a subspace-pooling matrix with groups of size of two as illustrated in the figure 2(we use four in the experiments). More specifically, each row of V picks and sums two neighboring feature dimensions in a non-overlapping fashion. The last term regularizes for sparsity in the pooling units. This design of pooling units is very similar with Independent Subspace Analysis(ISA) [13] and has the advantage of being able to learn overcomplete hidden representations. As V is hardcoded, we can efficiently optimize the loss function respect to the filter W via stochastic gradient descent.

3.3. Discriminative Feature Learning

The generative feature learning methods intend to learn the features or filters W by minimizing the reconstruction

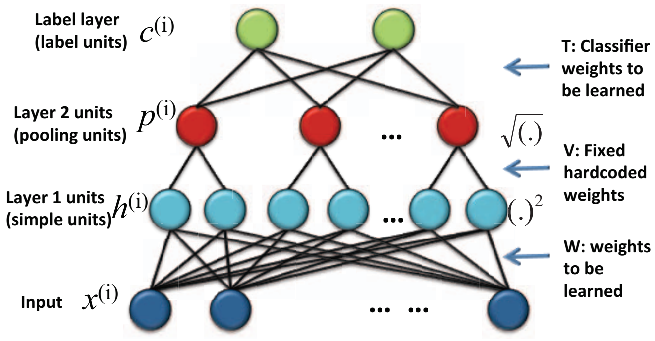
Label layer (label units) $c^{(i)}$
Layer 2 units (pooling units) $p^{(i)}$
Layer 1 units (simple units) $h^{(i)}$
Input $x^{(i)}$
T: Classifier weights to be learned
$\sqrt{(.)}$
V: Fixed hardcoded weights
$(.)^2$
W: weights to be learned

Figure 2: The neural network architecture for learning features at one level. Each dark blue node is an input pixel. Each light blue node is one feature response of the corresponding feature(filter) w. The red nodes are the pooling units pooling a non-overlap pair of feature responses(subspace is 2). The green node is the classification label which are used for discriminative feature learning.[Best viewed in color]

error and learn a set of redundant overcomplete features. However, a good generative property does not necessarily implies a good discriminative ability. In the experiments, we found out that by randomly picking out some filters learned in a generative way, the classification performance does not drop. It means that not all the features are useful in terms of classification. Moreover, a redundant set of features will increase the computational complexity and we want to avoid it. Notice that we have the collected confident set with labeled images. In order to incorporate this information, we add another objective to learn the features. The filters W are now not only learned from reconstructing the input x, but also a classifier predicting the label c from the representation p. A discriminative objective function computes an average classification loss between the actual label $c \in [0,1]^K$ and the predicted label $c' \in [0,1]^K$. More precisely, the loss function is used as a performance measure and we pose an optimization problem as follows:

$$E_{dis} = \sum_{i=1}^{N} \|\text{softmax}(Tp^{(i)}) - c^{(i)}\|_1, \qquad (5)$$

where softmax$(a)_k = \frac{exp(a_k)}{\square_{k'} exp(a_{k'})}, k = 1, ..., K$ for $a \in \mathsf{R}^K$. $c'^{(i)} = \text{softmax}(Tp^{(i)})$. The label c is a binary vector with a softmax unit that allows one element to be 1 out of K dimensions for K-way classification problem. T is the to be learned classifier weights as shown in figure 2.

When the input $x^{(i)}$ is local patches, the label c of $x^{(i)}$ is very hard to obtain since the object and non-object can possibly share the same local patches. To maintain the discriminative property, we can enforce the loss function at the image level instead of each local patch. We perform average pooling on the image from the feature maps of the local patches and the loss function can be modified as:

$$E'_{dis} = \sum_{j=1}^{N_I} \|\text{softmax}(\frac{1}{N_p} T \sum_{t=1}^{N_p} p^{(tj)}) - c^{(j)}\|_1. \qquad (6)$$

where we sum over N_I labeled images and the representation of image j is calculated by averaging the activation p of all the patches from image j. We can efficiently learn the features W using stochastic gradient descent. The byproduct of this algorithm is the weights T which is leaned jointly with W and we can utilize it in later classification process.

In our semi-supervised classification problem, the labeled data at an earlier stage does not represent the distribution of the whole data. To avoid from overfitting by the discriminative loss function, we further combine the discriminative and generative loss function to learn the discriminative features as follows:

$$E = E_{gen} + \beta E'_{dis}, \qquad (7)$$

where β is a coefficient balancing E_{gen} and E'_{dis}. The first term is common to many unsupervised learning algorithm and makes the system model the structure and the dependencies among the input components of x. The second term represents the supervised goal ensuring that codes are also going to be good for discriminating between class. In the rest of the paper, for simplicity, we will call the features learned by equation 7 as discriminative(hybrid) ones and equation 4 as generative ones.

3.4. Learning Higher Levels

We learn the features W from small image patches (small receptive field size) sampled from the confident labeled images at the beginning as the first level. Each feature in the first level is capturing local edges or color information. However, we expect to learn a more complex set of features, which can capture the conjunction of edges or even a global structure of the object, within a larger receptive field. To learn the higher-level features, we adopt a convolutional neural network architecture [12, 15] that progressively makes use of auto-encoders as sub-units as shown in Figure 3. The key ideas are as follow. We learn the first level filters by minimizing equation 7 on small input patches. Then we use the learned m filters to convolve with a larger region of the input image to obtain m feature maps. The max pooling operation is then performed over a certain neighborhoods. We can therefore extract local patches from these locally-invariant multidimensional feature maps and feed them to another level which is also implemented by auto-encoders. In our experiments, the stacked model is

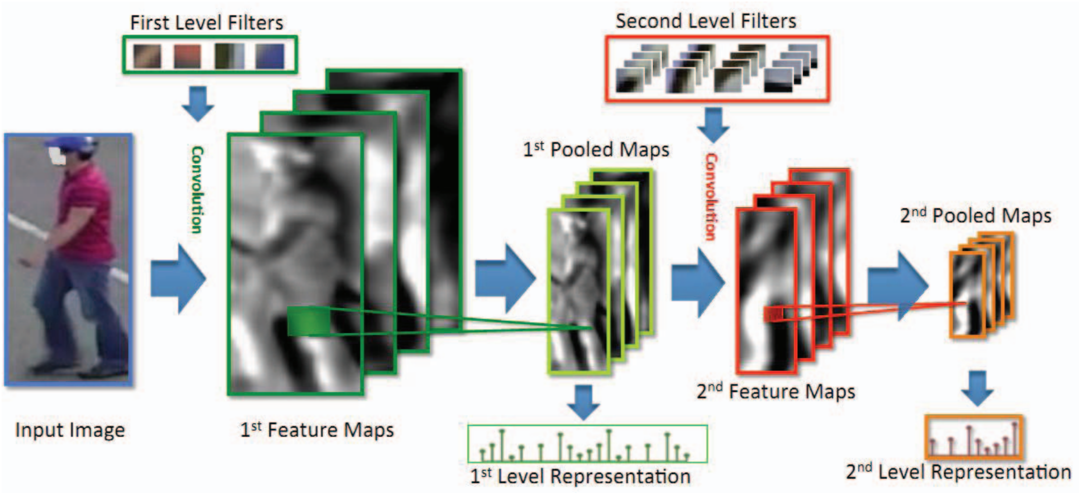

Figure 3: Object representation using a 2-level model. The first level learns features from color local patches using proposed algorithm. The feature maps are obtained by convolving each feature with the input image. Each feature map is then pooled from a 4×4 pixel non-overlapping grid to generate the pooled map. The concatenation of the pooled map serves as the representation of that level. The difference of the second level with the first level is that: the features in the second level are learned from the pooled local feature maps, instead of the larger local patches from the input images. [Best viewed in color]

trained greedily layerwise in the same manner as other algorithms proposed in the deep learning literature[11]. More specifically, we train the first level features until convergence before training the second level.

4. Experimental Results

4.1. Experimental Setup

We extensively experimented on the proposed method using four benchmark dataset: PETS2009 Dataset, Oxford Town Center dataset [3], PNNL Parking Lot datasets [25] and CAVIAR cols1 dataset [1] for human detection. Besides, we collected three videos from YouTube for horse detection. The frame resolution of the three videos is 450×360 and each video length is around 5 to 10 minutes. The number of frames containing horses is around 3000. We manually annotated the dataset. The challenge is the clutter background, occlusion and various poses of the horse. In all the sequences, we only use the detector scores and do not use any tracking results nor any annotation from the video.

In our experiments, we use the pre-trained pedestrian and horse model from [7]. We set a high recall and low precision point for this method in order to obtain almost all true detections and many false alarms. According to the detector confidence, we divide all detections into two groups: the ones with confident above a threshold are the positive examples; and the rest are hard examples and will be classified later. All the examples are resized to 128×64.

We learn three levels of features to represent the images. The first two levels are learned from 8×8 pixel wise patches from the input image and the first level pooled feature map respectively. The third level is learned directly from the second level pooled feature maps. The number of filters at each level is set as $m = 400$, the subspace size is 4. The number of feature maps at each level is therefore 100. The connected pooled value from each grid of each feature map serves as the final representation of that level. The final image representation is the combination of the three levels as illustrated in Figure 3. A linear SVM classifier is trained from the confident examples on the learned image representation and used for re-scoring the hard examples. We adopt the evaluation criterion of PASCAL VOC challenge. A detection is treated as a true positive if it has more than 0.5 overlap with the ground truth. We report the detection average precision (AP) to compare the performances.

In the following subsections, we first report our human detection performance, then compare the generative and discriminative features quantitatively and qualitatively. The features at each level are then analyzed based on the detection performance and we report our horse detection results in the end.

4.2. Human Detection Performance

We first evaluate our proposed method in terms of the human detection performance. The precision recall curve are shown in figure 4, where the red curves show the standard detector results and the blue curves show our results. Note that although the original detection results already had a sharp drop in precision near the maximum recall, our algorithm is still able to push the curve up. The AP is reported in the first and third row of table 1. Overall, our pro-

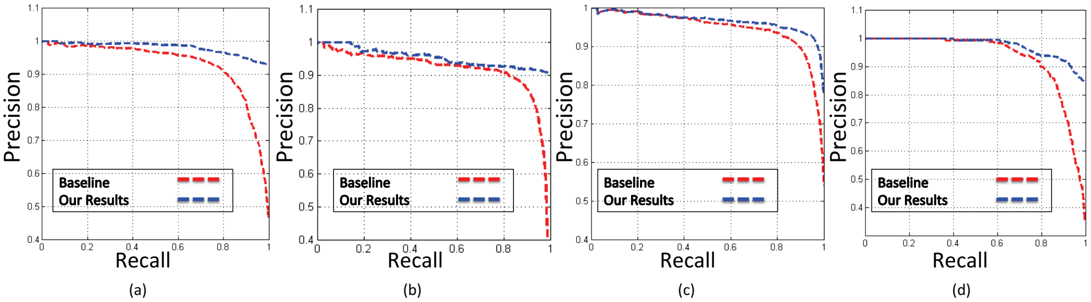

Figure 4: The precision-recall curve of four human detection datasets(a.TownCenter b.ParkingLot c.PETS09 d.CAVIAR). The red curves show the standard detector results and the blue curves show our results.

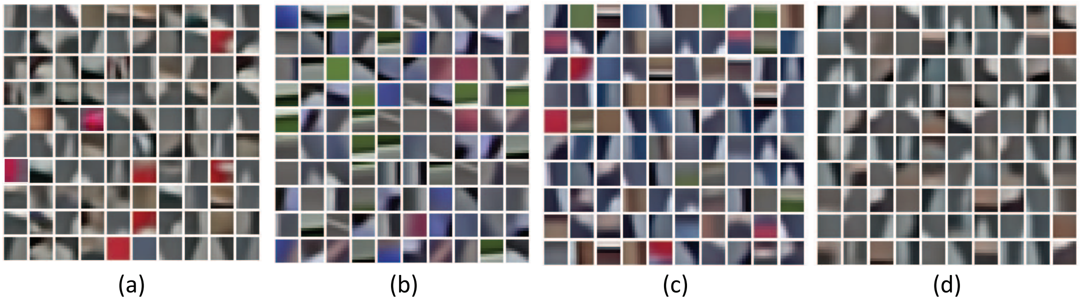

Figure 5: The 100 filters learned discriminatively from four human detection datasets(a.TownCenter b.ParkingLot c.PETS09 d.CAVIAR). The filters are visually different, especially in color, since each is learned from a specific video.[Best viewed in color]

posed method improves the generic offline detector's results 3-10% on the four benchmark human detection datasets. Further, we compared our learned feature with the classical hand-designed HOG feature. We use the same re-scoring pipeline and just replace the feature learning algorithm with the HOG feature. The AP is shown in table 1 second row. The average improvement is around 3%.

One of the advantage of our method is that it learns a specific discriminative compact set of features from the data itself, instead of using the combination of different classical hand-designed features. We show the final learned discriminative features from each dataset in figure 5. (a)-(d) are the corresponding learned first level features from TownCenter, ParkingLot, PETS09 and CAVIAR. As we can see that, the four set of features are visually very different from each other. Each captures the specific representative color and edge information in the corresponding dataset. We argue that using the learned features is more efficient and effective, especially in videos. In contrast to boosting, which selects the good features from a pre-defined feature pool, our method dynamically selects and learns the good features from the raw pixels.

4.3. Discriminative Vs. Generative

As described previously, the features learned in a generative manner are usually over-complete, which are good for reconstruction, but are not necessarily effective for recognition. Hence, we propose to directly learn the discriminative features for particular video by adding the discriminative loss function. We train the classifier and the filters at the same time to find out which features are good for recognition. To qualitatively visualize the differences of the two set of features, we show the features generatively learned from TownCenter and CAVIER dataset in figure 6. By observing the original video, we found that color information is more distinguished for detecting a person in the TownCenter than CAVIAR. Comparing figure 5(a) with 6(a), interestingly, the color information is more emphasized by the discriminatively learned features. Comparing figure 5(d) with 6(b), the color information is now more emphasized by the generative learned feature, since most of the negative examples are the colored background and most of the people are wearing dark clothes.

To quantitatively measure the generatively and discriminatively learned features, we compute the Average Precision (shown in table 1 third and fourth row) of the two sets of features on the four dataset. On average, the discriminative learned features are 2% better than the generative learned features. Further, we show that the discriminative learned feature set is more compact than the generative learned feature. We compute the AP by increasing the learned features at each level from 40 to 800, the re-

	Town Center	Parking Lot	PETS 09	CAVIAR
Baseline [7]	86.9	86.4	93.7	91.1
HOG	92.1	92.9	94.5	92.5
Discriminative	**95.4**	**96.9**	**95.5**	**94.3**
Generative	94.3	94.1	93.7	93.4
first level	94.2	95.1	94.5	92.9
second level	93.7	93.2	93.1	91.6
third level	92.1	93.5	92.7	90.8
1+2+3 level	**94.6**	**96.3**	**95.5**	**93.1**

Table 1: The average precision of different methods or experimental setups on four benchmark datasets for human detection. The first row is the results from a generic detector. The second row is using the the same re-scoring process but HOG feature without our feature learning algorithm. The third row is the results of the proposed discriminative(hybrid) features. The fourth row is the generatively learned feature results. Overall, our proposed algorithm improves the detection results of the generic object detector by $3-10\%$ and the HOG features by 3%.The last four row are the detection results using the by-product weights T of our method for re-scoring instead of training SVM on each level.

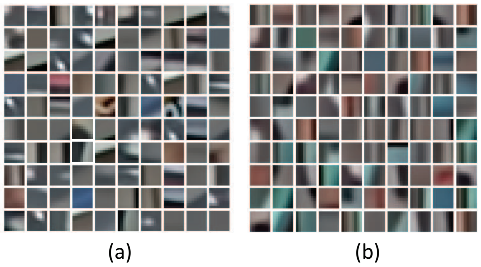

Figure 6: The 100 filters learned generatively from Towncenter(a) and CAVIAR(b) dataset. Compared with the corresponding discriminative filters (a) and (d) in figure5, the generative features are quite different especially in color. [Best viewed in color]

sults are shown in figure 7. Interestingly, we found that the discriminative features reach the highest average precision when 400 features are learned. More features do not improve further in terms of the classification accuracy. Whereas, the generative learned features do need a large set of over-complete features to capture enough discriminative information. This demonstrate that our proposed method can not only improve the classification accuracy, but also boost the computational efficiency.

4.4. Performance at Each Level

As described in subsection 3.3 equation 6, the by-product of our learning algorithm is the weights T, which can be also used as a classifier in our task. In table 1(last four rows), we show the performance of each level and their combination in a late fusion manner, when we only use weights T as a linear classifier. As we can see that the first level plays an important role in terms of classification accuracy. The performance of the second and the third levels are lower than the first level, but the combination of them performs the best.

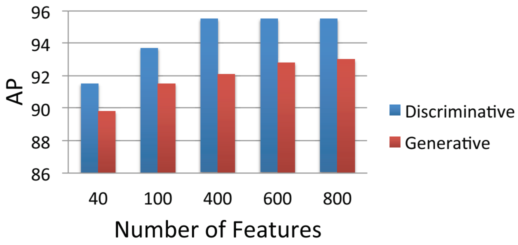

Figure 7: The average precision of discriminative and generative method over different number of feature. Given a target AP, the discriminative one reaches it with less number of features. This means the discriminative one is more compact than the generative one.

	Horse1	Horse2	Horse3
Baseline[7]	51.2	56.5	63.4
HOG	53.1	58.1	66.6
Discriminative(hybrid)	**59.6**	**64.9**	**71.2**
Generative	54.7	61.3	67.1
First level	56.2	62.1	68.9
Second level	56.6	62.9	66.3
Third level	56.1	62.2	67.2
1+2+3 level	**58.7**	**64.1**	**69.4**

Table 2: The average precision of different methods or experimental setups on three horse videos. Overall, our proposed algorithm improves the detection results of the generic object detector by 7% and the HOG by 5-6%.

4.5. Performance on Horse Detection

To demonstrate the generality of our method, we performed further experiments for another object, horse, on the videos collected from YouTube. All the quantitative results are shown in table 2. The generic offline trained horse detector performs averagely 57% on the dataset, whereas, our approach achieves significantly better results than the original detector. Overall, we improve the AP by 7%.

5. Conclusion and Future Work

We propose to learn a feature hierarchies directly from the raw pixels in a particular video to improve a generic detector. We consider the discriminative property of features, and simultaneously learn discriminative and reconstructive features by using both a supervised and an unsupervised objective. Extensive experiments results demonstrate the efficacy of our approach. The future work will be incrementally learn the discriminative set of features instead of a fixed size of feature set.

6. Acknowledgements

The research presented in this paper is supported by the Intelligence Advanced Research Projects Activity (IARPA) via Department of Interior National Business Center contract number D11PC20071. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of IARPA, DoI/NBC, or the U.S. Government.

References

[1] Caviar. `http://homepages.inf.ed.ac.uk/rbf/CAVIAR/`.

[2] M. Andriluka, S. Roth, and B. Schiele. People-tracking-by-detection and people-detection-by-tracking. In *CVPR*, 2008.

[3] B. Benfold and I. Reid. Stable multi-target tracking in real-time surveillance video. In *CVPR*, 2011.

[4] Y. Bengio, A. C. Courville, and P. Vincent. Unsupervised feature learning and deep learning: A review and new perspectives. *CoRR*, 2012.

[5] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *CVPR*, 2005.

[6] P. Dollár, C. Wojek, B. Schiele, and P. Perona. Pedestrian detection: An evaluation of the state of the art. 2011.

[7] P. Felzenszwalb, R. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part based models. In *PAMI*, 2010.

[8] P. Felzenszwalb, D. McAllester, and D. Ramanan. A discriminatively trained, multiscale, deformable part model. In *CVPR*, 2008.

[9] D. M. Gavrila and S. Munder. Multi-cue pedestrian detection and tracking from a moving vehicle. *IJCV*, 2007.

[10] H. Grabner, C. Leistner, and H. Bischof. Semi-supervised on-line boosting for robust tracking. ECCV, 2008.

[11] G. E. Hinton, S. Osindero, and Y. whye Teh. A fast learning algorithm for deep belief nets. *Neural computation*, 2006.

[12] F.-J. Huang and Y. LeCun. Large-scale learning with svm and convolutional nets for generic object categorization. In *CVPR*, 2006.

[13] A. Hyvarinen and P. Hoyer. Emergence of phase- and shift-invariant features by decomposition of natural images into independent feature subspaces. *Neural Computation*, 2000.

[14] O. Javed, S. Ali, and M. Shah. Online detection and classification of moving objects using progressively improving detectors. In *CVPR*, 2005.

[15] K. Kavukcuoglu, P. Sermanet, Y. Boureau, K. Gregor, M. Mathieu, and Y. LeCun. Learning convolutional feature hierachies for visual recognition. In *NIPS*, 2010.

[16] Q. V. Le, R. Monga, M. Devin, G. Corrado, K. Chen, M. Ranzato, J. Dean, and A. Y. Ng. Building high-level features using large scale unsupervised learning. *ICML*, 2012.

[17] H. Lee, R. Raina, A. Teichman, and A. Y. Ng. Exponential family sparse coding with application to self-taught learning. In *IJCAI*, 2009.

[18] A. Levin, P. A. Viola, and Y. Freund. Unsupervised improvement of visual detectors using co-training. In *ICCV*, 2003.

[19] J. Mairal, F. Bach, J. Ponce, G. Sapiro, and A. Zisserman. Discriminative learned dictionaries for local image analysis. In *CVPR*, 2008.

[20] V. Nair and J. J. Clark. An unsupervised, online learning framework for moving object detection. In *Proceedings of the 2004 IEEE computer society conference on Computer vision and pattern recognition*, CVPR, 2004.

[21] M. Ranzato and M. Szummer. Semi-supervised learning of compact document representations with deep networks. In *ICML*, 2008.

[22] C. Rosenberg, M. Hebert, and H. Schneiderman. Semi-supervised self-training of object detection models. In *WACV/MOTION*, 2005.

[23] P. M. Roth, H. Grabner, D. Skoaj, and H. Bischof. On-line conservative learning for person detection. In *In Proc. VS-PETS*, 2005.

[24] P. M. Roth, S. Sternig, H. Grabner, and H. Bischof. Classifier grids for robust adaptive object detection. In *CVPR*, 2009.

[25] G. Shu, A. Dehghan, O. Oreifej, E. Hand, and M. Shah. Part-based multiple-person tracking with partial occlusion handling. In *CVPR*, pages 1815–1821, 2012.

[26] G. W. Taylor, R. Fergus, Y. LeCun, and C. Bregler. Convolutional learning of spatio-temporal features. In *ECCV*, 2010.

[27] A. Vedaldi, V. Gulshan, M. Varma, and A. Zisserman. Multiple kernels for object detection. In *ICCV*, 2009.

[28] P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol. Extracting and composing robust features with denoising autoencoders. In *ICML*, 2008.

[29] P. Viola and M. Jones. Robust real-time object detection. In *IJCV*, 2001.

[30] X. Wang, T. X. Han, and S. Yan. An hog-lbp human detector with partial occlusion handling. In *ICCV*, pages 32–39, 2009.

[31] X. Wang, G. Hua, and T. X. Han. Detection by detections: Non-parametric detector adaptation for a video. In *CVPR*, 2012.

[32] Y. Yang and M. Shah. Complex events detection using data-driven concepts. In *ECCV*, 2012.

[33] M. D. Zeiler, G. W. Taylor, and R. Fergus. Adaptive deconvolutional networks for mid and high level feature learning. In *ICCV*, 2011.

[34] C. Zhang, R. Hamid, and Z. Zhang. Taylor expansion based classifier adaptation: Application to person detection. In *CVPR*, 2008.

[35] G. Zhou, K. Sohn, and H. Lee. Online incremental feature learning with denoising autoencoders. *Journal of Machine Learning Research*, 22, 2012.