Leveraging Inexpensive Supervision Signals for Visual Learning

Technical Report Number:
CMU-RI-TR-17-13

a dissertation presented
by
Senthil Purushwalkam Shiva Prakash
to
The Robotics Institute

in partial fulfillment of the requirements
for the degree of
Master of Science
in the subject of
Robotics

Carnegie Mellon University
Pittsburgh, Pennsylvania
May 2017

Thesis advisor: Abhinav Gupta Senthil Purushwalkam Shiva Prakash

Leveraging Inexpensive Supervision Signals for Visual Learning

Abstract

The success of deep learning based methods for computer vision comes at a cost. Most deep neural network models require a large corpus of annotated data for supervision. The process of obtaining such data is often time consuming and expensive. For example, the process of collecting bounding box annotations takes 26-42 seconds per box. This requirement poses a hindrance for extending these methods to novel domains. In this thesis, we explore techniques for leveraging inexpensive forms of supervision for visual learning. More specifically, we first propose an approach to learn a pose-encoding visual representation from videos of human actions without any human supervision. We show that the learned representation improves performance for pose estimation and action recognition tasks compared to randomly initialized models. Next, we propose an approach to use freely available web data and inexpensive image-level labels to learn object detectors. We show that web data, while highly noisy and biased, can be effectively used to improve localization of objects in the weak-supervision setting.

Contents

Listing of figures

Genius always gives its best at first;
prudence, at last.

-Seneca the Younger

1 Overview

The success of deep learning based methods in Computer Vision has led to algorithms that are now deployable in the wild. For example, deep convolutional neural networks (CNNs) have been successfully deployed for applications like visual search[44], video surveillance[83] and even self-driving cars[46]. These tasks usually require a high-level semantic understanding of visual signals and were far from being an achievable goal a few years ago. The success of these methods is usually attributed to the increase in computational power and the rise of large scale annotated datasets which provide strong supervisory signals. The first approach that demonstrated the successful application of deep learning involved training an 8-layered convolutional neural network using the ImageNet dataset[16].

The ImageNet dataset consists of 1.2million images each of which are annotated with 1 out of 1000 object classes. The labels of these images were hand-annotated by humans. This process is not only time consuming, but also is very expensive. Due to this bottleneck, ImageNet has been one of the few efforts that has generated annotated data at such a large scale. Apart from the image classification task, annotated data has been collected for many other computer vision tasks. For object detection, the MSCOCO[48] consists of about 328,000 images. For these images, bounding box and segmentation mask annotations were collected for 91 kinds of objects. The process of annotating bounding boxes and segmentation masks is clearly more time consuming compared to collecting image-level labels. In [66], it is reported that the time required to annotated a bounding box is 26-42 seconds on average depending on the level of quality required.

While craftily designed deep neural networks trained on these datasets with strong supervision show impressive results, an important question still remains unanswered: is strong supervision really necessary? Most animals do not require supervision at this level to develop a strong visual perception system. This leads to the belief that computer vision models too could infact be trained without supervision of humans. In order to address this question in the deep learning paradigm, numerous works have attempted to learn visual 'representations' without any human annotations[20, 36, 78]. A visual representation is a feature encoding of the visual signal that is discriminative for the task at hand. In order to avoid the expensive collection of annotations, numerous works also address the problem of learning from cheaper and easier forms of supervision.

In this thesis, we address computer vision tasks by making use of no supervision or inexpensive forms of supervision. Specifically, we propose an approach for unsupervised representation learning from videos in Chapter 2. We show that the representation learned can be used for pose estimation and action recognition. In Chapter 3, we propose an approach for weakly supervised object detection that makes use of freely available web-based image search results.

We see in order to move; we move in order to see.

-William Gibson

2

Pose from Action: Unsupervised Learning of Pose Features based on Motion

2.1 Introduction

In recent years, there has been a dramatic change in the field of computer vision. Owing to visual feature learning via convolutional neural networks, we have witnessed major performance gains in different areas including image classification [59, 69], object detection [26–28], 3D scene understanding [80], pose estimation [72] etc. In most cases, visual features are first learned by training for the classification task on the ImageNet dataset followed by fine-tuning the pre-trained network for the task at hand.

While this classification based learning framework has yielded significant gains, it is unclear if this is the right approach to visual feature learning. For example, in case of humans, we do not need millions of category-labeled images/videos to learn visual features. Instead, we can learn a visual representation by observing and actively exploring the dynamic world around us. Furthermore, the manual labeling of images remains a significant bottleneck in exploiting a larger number of images to learn visual representations. As a consequence, there has been rising interest in the area of unsupervised feature learning.

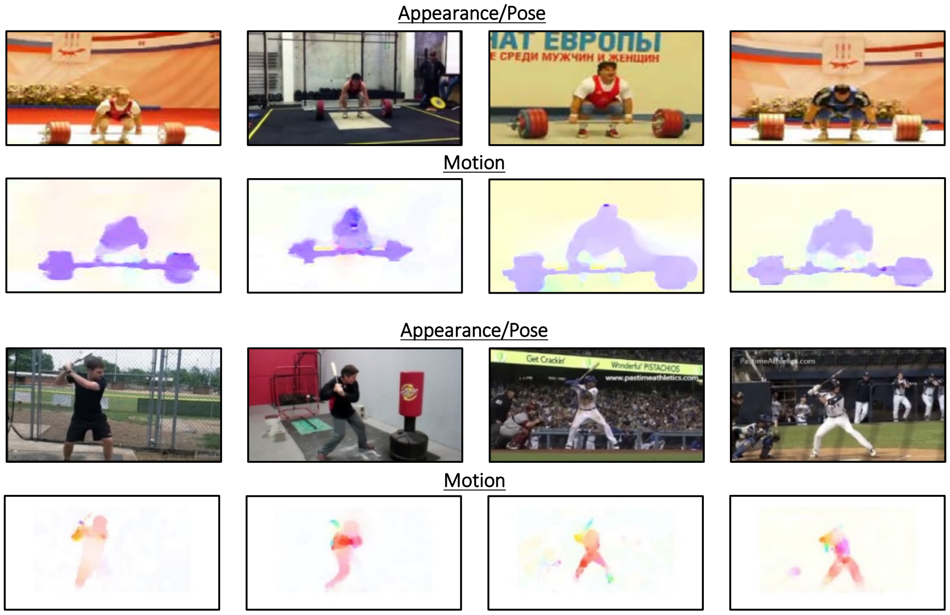
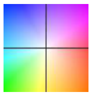


Figure 2.1.1: Similar poses are related to similar motions. Hence motion can be used as a supervisory signal to learn appearance representations. We use the following color coding to visualise the optical flow: 

There are two paradigms for unsupervised feature learning: generative and discriminative. In the generative learning paradigm, we learn a low-dimensional representation that can be used to generate realistic images. These networks use denoising or reconstruction loss with regularization such as sparsity of the learned space. However, the generative learning approaches have been been not been competitive on tasks like object classification or detection.

In the discriminative learning paradigm, the network is trained using standard back-propagation on an auxiliary task for which ground truth can be easily mined in an automated fashion. The hope is that the visual representation learned for this auxiliary task is generalizable and would work for other tasks with simple fine-tuning. Owing to the rise of interest in unsupervised learning, many such auxiliary tasks have been proposed in the recent past. [20] proposed to take pair of patches sample from an image and predict the relative location of the patches, which seems to generalize to suprisingly well to object detection. [1, 36] proposed an approach to take pair of patches and predict the camera motion that caused the change. The ground-truth for this task is obtained via other sensors which measure ego-motion. Finally, [78] presents an approach to sample a pair of patches via tracking and learn a representation which embeds these patches close in the visual representation space (since they are the same object with some transformations).

While [1, 36, 78] use videos for unsupervised learning, they used other sensors or just different viewpoints to train the appearance models. We argue that there is a complementary and stronger signal in videos to supervise the training of these networks: motion patterns. The key inspiration for our proposed method is that similar pairs of poses are associated with similar motion patterns(See Figure 2.1.1). In this paper, we demonstrate how motion patterns in the videos can act as strong supervision to train an appearance representation. We hypothesize that an appearance representation where poses associated to similar motion patterns cluster together could be useful for tasks like Pose Estimation and Action Recognition. We believe that the proposed approach is generic and can be used to learn different kinds of pose-encoding appearance representations based on different kinds of videos. Specifically, in this paper, we choose to work with human action videos since the learnt representations can be semantically associated to human poses. We believe that this idea can provide the missing link in unsupervised learning of visual representations for human actions and human poses.

However, there is still one missing link: how do you compare motion patterns. One way is to use distance metric on hand designed motion features (e.g., 3DHOG, HOF[76]) or the optical flows maps directly. Instead, inspired by the success of the two-stream network[62], we try to jointly learn convolutional features for both the appearance(RGB) and the motion(optical flow) at the same time. Our key idea is to have triplet network where two streams with shared parameters correspond to the first and n^{th}

frame in the video; and the third stream looks at $n - 1$ optical flow maps. All the convolutional streams run in a feedforward manner to produce 4096 dimensional vectors. The three streams are then combined to classify if the RGB frames and optical flow channels correspond to each other *i.e.* does the transformation causes the change in appearance?. Intuitively, solving this task requires the Appearance ConvNet to identify the visual structures in the frame and encode their poses. The Motion ConvNet is expected to efficiently encode the change in pose that the optical flow block represents. We evaluate our trained appearance network by finetuning on the task of pose estimation on the FLIC dataset[70], static image action recognition on PASCAL VOC[23], and action recognition on UCF101[65] and HMDB51[47]. We show that these models perform significantly better than training from random initialisation.

2.2 Related Work

Unsupervised Learning

Training deep learning models in a supervised fashion generally requires a very large labeled training set. This is infeasible and expensive in a lot of cases. This has led to an increase of attention to unsupervised techniques to train these models. Research in unsupervised representation learning can be broadly divided into two categories - generative and discriminative. The approach proposed in this paper belongs to the latter.

Majority of the discriminative approaches involve intelligently formulating a surrogate task which involves learning from an easily available signal. These tasks are designed such that the deep model is forced to learn semantics relevant to us like object labels, human poses, activity labels, etc. In [20], the formulated task involved predicting the relative location of two patches. Automatically cropping pairs of patches from any image makes the 'relative location' signal readily available. The key motivation here is that performing well in this task requires understanding object properties. Hence the Convolutional Neural Network trained to perform this task is shown to perform well on object classification and detection tasks. Similarly, the surrogate task proposed in this paper involves predicting whether a transformation (inferred from optical flow) represents the same transformation as that between a given pair of appearance features.

Unsupervised learning algorithms that learn from videos are extremely valuable since the amount of video data available to us is massive and collecting annotations for them is infeasible. In [78], patches are tracked across frames of videos to generate pairs which are visually dissimilar but semantically same. An unsupervised representation is then learnt by enforcing the similarity on the pair of features extracted for the patches. This structure in the feature space is enforced using a triplet ranking loss which minimises the distance between the pair of features and simultaneously maximises the distance to a feature extracted for a randomly chosen patch. While this approach shows impressive results on a wide range of tasks, it suffers from two drawbacks. First, the constraint explicitly enforced leads to an appearance representation which is invariant to pose, size and shape changes in an object. Second, the spatially and temporally sparse samples of patches do not make use of all the information available in the videos. In contrast, we attempt to learn a representation that encodes the structural changes by making use of densely sampled pairs of frames to capture a large number of variations in poses.

The unsupervised learning approaches which are closely related to our work are video-based approaches which model similarities or differences across frames[6, 33, 36, 51, 52]. A large number of approaches use the idea of temporal coherance to train unsupervised representations. These methods exploit the fact that appearances change slowly between adjacent frames[37].

A recently proposed approach [36] involves learning a representation in which transformations are 'predictable'. The feature representation is learnt by specifically enforcing the constraint that similar ego-centric motions should produce similar transformations in the feature space. This approach requires a dataset of video frames annotated with the corresponding ego-poses and hence is not scalable. In our proposed approach, we eliminate this requirement by jointly learning to infer a representation for the transformation from optical flow maps which are easy to compute.

Action Recognition and Pose Estimation

The task of recognizing human actions from images and videos has received a lot of attention in computer vision [43, 47, 65, 79]. Activity recognition is a challenging computer vision task since recognizing human actions requires perception of the environment, identifying interaction with objects, ***understanding pose changes in humans*** and a variety of other sub-problems. Most successful action recognition methods involve using combinations of appearance, pose and motion information as features [15, 49, 82]. A decade of research in action recognition has led to approaches that show impressive performances on benchmark datasets[21, 45, 56, 75, 77]. The majority of successful algorithms for action classification follow a common pipeline. Appearance or motion features are first extracted either densely or at interest points. This is followed by clustering and generating an encoding. These encoded feature vectors are then classified using various kinds of classifiers. Recently, deep learning based methods have been extended to action recognition[43]. It has been observed that training deep neural networks directly on stacks of video frames is too computationally expensive and does not lead to significant improvements over handcrafted feature based methods[38]. More recent methods operate on individual frames independently since it is observed that this gives similar performance as using a stack of frames [43]. The Two-Stream network[62] is a fully-supervised deep-learning based action recognition method which achieves performances comparable to state-of-the-art. It involves training independent spatial and temporal networks whose classification scores are fused to give the final prediction. Deep learning methods have also been extended to estimating poses in images and videos. The task of pose estimation involves estimating the locations of body parts. [72] uses a deep neural network based regressor to estimate the coordinates of the parts. The model is recursively applied on patches cropped around the previous prediction to obtain better localisation. In [57], a deep convolutional neural network is used to predict heat maps for the location of each body part. The model also uses a spatial fusion technique to capture multi-scale information.

Actions and Poses are very closely related concepts. An action comprises of a sequence of poses in conjunction with interactions with the environment. Videos are a widely available and rich source of actions. As a consequence, they are also the best source for diverse human poses. In [9], a large collection of unlabelled video is searched to augment training data by finding similar poses using the poselet activation vector[49]. To the best of our knowledge, the approach proposed in this paper is the first in attempting to learn pose features from videos using deep networks in an unsupervised fashion.

2.3 Approach

The goal of this paper is to learn an appearance representation that captures pose properties without the use of any human supervision. We achieve this by formulating a surrogate task for which the ground truth labels are readily available or can be mined automatically. In simple terms, given a change in appearance, the task we formulate involves predicting what transformation causes it. For example, in Figure 2.3.1, given the appearance of Frame 1 and Frame13, we can predict that the transformation of 'swinging the bat' caused the change in appearance. In this section, we first develop an intuitive motivation for the surrogate task and then concretely explain how it can be implemented.

Suppose we want to train a model to predict if a Transformation $\mathbf{T}$ causes the change in Appearance $\mathbf{A} \rightarrow \mathbf{A}'$. We would need to have a robust way to encode $\mathbf{A}$, $\mathbf{A}'$ and $\mathbf{T}$ such that they capture all the information required to solve this task. More specifically, given an image, the appearance representation $\mathbf{A}$ needs to localise the object(s) that could undergo a transformation and encode its properties such as shape, size and more importantly, ***pose***. On the other hand, given a motion signal (like optical flow, dense trajectories [75, 77], etc), the transformation representation $\mathbf{T}$ needs to express a robust encoding that is discriminative in the space of transformations.

We propose to learn the appearance representation $\mathbf{A}$ using a convolutional neural network (Appearance ConvNet in Figure 2.3.1). We choose to use optical flow maps as the motion signal in our proposed approach. There are a large variety of existing methods like 3dHOG and HOF [8, 75] which can be used to extract an encoding for the optical flow maps. These methods first extract local descriptors in the volume of optical flow maps, and this is generally followed by a bag-of-words model to generate a feature vector.

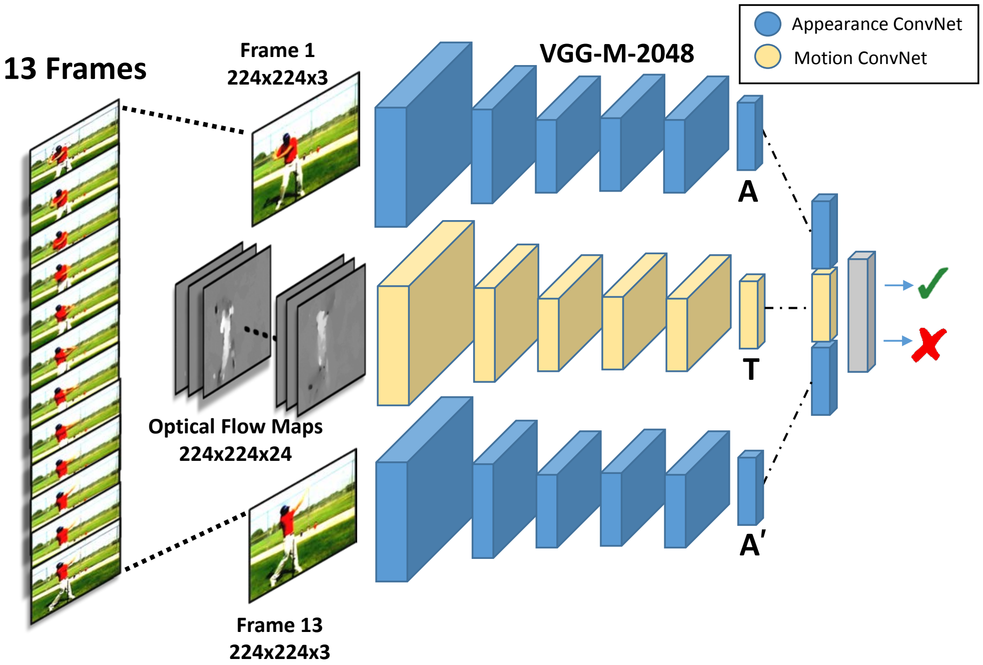


Figure 2.3.1: An overview of our approach. Predicting whether a transformation encoding T causes the change in appearance A→A' requires capturing pose properties.

Instead of using these hand-crafted approaches, we propose to jointly learn the motion representation as a Transformation **T** using a separate convolutional neural network (Motion ConvNet in Figure 2.3.1). The idea of using two independent networks to represent appearance and motion is very similar to the Two-Stream Network [62] which recently achieved accuracies very close to state-of-the-art in action recognition. The Appearance ConvNet takes as input an RGB image and outputs a feature vector. Similarly, the Motion ConvNet takes as input a stack optical flow maps as input and outputs a feature vector.

We propose an unsupervised approach to jointly train the Appearance and Motion ConvNets. The key idea of our approach is that given two appearance features $\mathbf{A}$ and $\mathbf{A}'$, it should be possible to predict whether a Transformation **T** causes the change $\mathbf{A} \rightarrow \mathbf{A}'$. This idea is synchronous with [36], where the notion of ego-motions producing predictable transformations is used to learn an unsupervised model.

Following this intuition, for a video snippet **i**, we extract appearance features for Frame n $(\mathbf{A_i}(n))$ and Frame $n + \Delta n$ $(\mathbf{A_i}(n + \Delta n))$ using the Appearance ConvNet. We then extract motion features for Δn optical flow maps for Frames k to $k + \Delta n$ from a random video snippet j $(\mathbf{T_j}(k, k + \Delta n))$ using the Motion ConvNet. We then use two fully connected layers on top of the three concatenated features to predict whether the transformation $\mathbf{T_j}(k, k + \Delta n)$ could cause the change $\mathbf{A_i}(n) \rightarrow \mathbf{A_i}(n + \Delta n)$ *i.e.*

$$\mathbf{T_j}(k, k + \Delta n) = \mathbf{T_i}(n, n + \Delta n)$$

We randomly (and automatically) sample *i*,*n*,*j*, k and keep Δn fixed. This makes the positive and negative labels readily available *i.e.* the positive examples are the triplet samples where $i = j$ and $n = k$. All the others samples could be treated as negatives, but to account for videos with repetitive actions (like walking), we mine negatives from other videos *i.e.* we do not use samples where $i = j$ and $n \neq k$. Fixing Δn to a constant value is necessary since we need to fix the filter size in the first layer of the Motion ConvNet.

In summary, the joint unsupervised learning pipeline consists of one Motion ConvNet, two instances of the Appearance ConvNet and a two-layer fully connected neural network on top. The parameters of the two Appearance ConvNets are shared since we expect both networks to encode similar properties. Overall the joint system of three neural networks can be treated as one large neural network. This allows us to use standard back propagation to train all the components simultaneously.

Implementation Details

In our experiments, we fix $\Delta n = 12$ **i.e.** we sample pairs of frames which are separated by 12 frames. We follow the VGG-M architecture for the Appearance ConvNet and Motion ConvNet till the FC6 layer. The only difference is the size of Conv1 filters in the Motion ConvNet which has 24 channels instead of 3 to accommodate convolution on 24 optical flow maps (12 in the x-direction and 12 in the y direction). This gives us a 4096-dimensional vector representation for each of $\mathbf{A}$, $\mathbf{A}'$ and $\mathbf{T}$. We then concatenate the three feature vectors to get a 12288 dimensional vector and use a fully connected neural network to perform the binary classification. The first fully-connected layer has 4096 output neurons followed by second fully connected layer with 2 output neurons. A softmax classifier is then used to predict the binary labels.

Patch and Optical Flow Mining

In order to train the binary classification model, we require a large collection of pairs of frames, the correct block of optical flow maps between them and multiple negative samples of optical flow blocks. As the training set, we use a large collection of video which contain humans performing actions. This set is formed by combining the training set videos from the UCF101 [65](split1), HMDB51 [47] (split1) and the ACT[79] datasets. For every pair of consecutive frames we precompute the horizontal and vertical directional optical flow maps using the OpenCV GPU implementation of the TVL1 algorithm[50].

As inputs to the Appearance ConvNet we randomly sample a spatial location and crop 224x224 patches at that location from two frames separated by $\Delta n (= 12)$ frames. For the Motion ConvNet, we sample the 224x224 patches from each of the 12 horizontal and 12 vertical flow maps in between the two sampled frames at the same location, as the positive (label$=$ 1) which gives us a 224x224x24 dimensional array. As the negative examples (label$=$ 0), we randomly sample another 224x224x24 block from a random spatial location in a randomly picked video. During training, we pick the negatives from the same batch in the mini-batch stochastic gradient descent procedure and ensure that negative flow blocks are not picked from the same video as the appearance frames. We also augment the training data by randomly applying a horizontal flip on a (Frame n, Frame $n + \Delta n$, Optical Flow Block) triplet. Since all motion signals also make sense in the reverse direction temporally (they do not necessarily hold any semantic value), we also randomly reverse some triplets *i.e.* (Frame $n + \Delta n$, Frame n, reversed optical flow block).

For the joint training procedure, we use a batchsize of 128 *i.e.* 128 pairs of patches. The SoftMax Loss is used to compute the errors to train the network. We initially set the learning rate to 10^{-3}, momentum to 0.9 and train for 75,000 iterations. We then reduce the learning rate to 10^{-4} and train for 25,000 iterations. At convergence, the joint system performs around 96% on the formulated binary classification task for a held out validation set (note that the baseline performance is 66% since we have two negatives for each positive).

2.4 Experiments

The efficacy of unsupervised feature representation learning methods are generally tested on tasks for which the learnt representation might prove useful. First, the learnt representations are finetuned for the task using either the full labelled dataset (generally trained for a small number of iterations) or a small subset of the training set. Then these finetuned models are tested to provide evidence for the transferable nature of the representation learnt.

We follow a similar strategy and perform an extensive evaluation of our unsupervised model to investigate the transferability of the learned features. In order to build a better understanding of what the models learn, we first perform a qualitative analysis of the models. As explained before, since our unsupervised model is trained on action videos, this leads to an appearance representation (Appearance ConvNet) that is expected to capture pose properties well. Feature representations that capture pose properties are valuable for estimating human poses. Another domain where pose information proves immensely useful [9, 15, 49, 82] is recognizing human actions since any action involves a series of poses. Following this intuition, we test our learned representation for the Pose Estimation and Action Recognition tasks.

We also compare our method to two popular and recent unsupervised representation learning methods which also attempt to learn from videos. The results demonstrate the superiority of our learnt representation for these tasks. The first unsupervised model, proposed by Wang et. al in [78], involves enforcing the constraint that two transformed versions of the same object (different viewpoint, pose, size, etc) needs to represent the same point in the feature space. This leads to a feature representation that is invariant to pose, shape and size changes. The second model, proposed in [37], involves enforcing temporal coherence in the feature space by imposing a prior on the higher order derivatives to be small. This is trained jointly with the classification loss for the supervised task. We compare to this model since it is the most recently introduced unsupervised technique for videos.

2.4.1 Qualitative analysis of learned models

The first layer of a convolutional neural network is often visualised to verify that the network learns meaningful representations. We present the visualisations of the 96 filters in the first convolutional layer of the Appearance ConvNets in Figure 2.4.1. Clearly, the visualisation shows that the filters learn to model gradient like features.

We investigate the pose capturing capability of the learned unsupervised representation in the Appearance ConvNet by visualising closest pairs in the FC6 feature space. We first compute the appearance features for all image in the Leeds Sports Pose(LSP) dataset [40]. We randomly sample images and find the closest image in the rest of the dataset use the Euclidean distance between the appearance features extracted. We present these closest pairs in Figure 2.4.2. From these pairs, it is evident that the Appearance ConvNet is able to match poses reasonably well. This observation suggests that the Appearance ConvNet indeed attempts to capture the pose properties of humans.

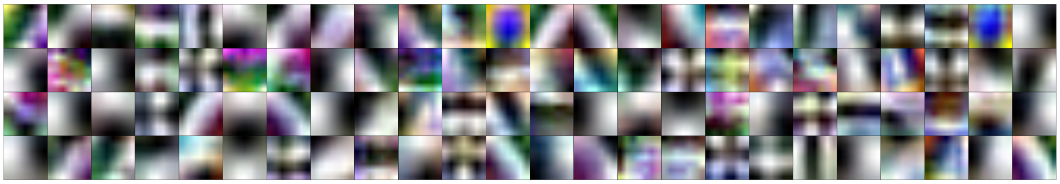


Figure 2.4.1: Visualisations of filters in the first convolution layer in the Appearance ConvNet.

2.4.2 Pose Estimation

The task of estimating human poses from videos and images is an important problem to be solved and has received widespread attention. In its most simple form, the task is defined as correctly localising the joints of the human. Computer vision research focusing on pose estimation has given rise to a large number benchmark which contain videos and images [2, 57, 70] with their annotated joints. We evaluate the efficacy of our

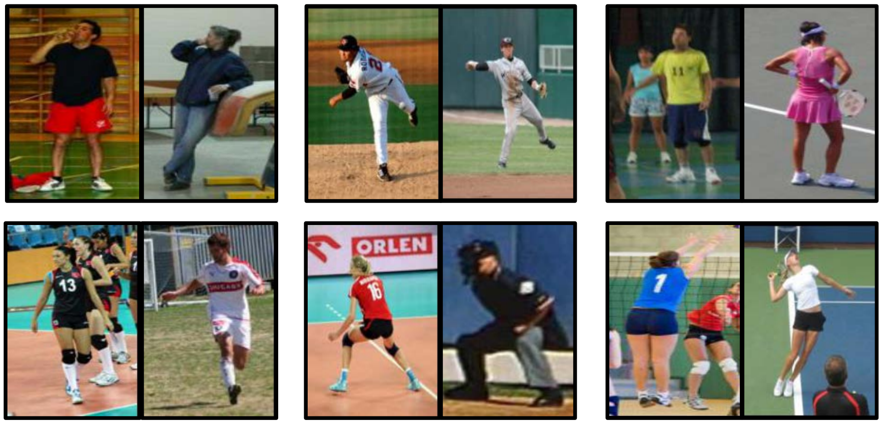


Figure 2.4.2: Closest image pairs in the FC6 feature space of the Appearance ConvNet.

learnt Appearance ConvNet by testing it for estimating human poses in the Frames Labelled in Cinema (FLIC) dataset [70]. This dataset contains 5003 images with the annotated joints collected using crowd-sourcing. The train and test splits contain 3987 and 1016 images respectively.

We design a simple deep learning based pose estimation architecture to allow us the freedom to accommodate other unsupervised models. This also improves interpretability of the results by minimising the interference of complementary factors on the performance. Figure 2.4.3 presents an overview of the architecture we use to perform pose estimation (referred as Pose ConvNet). We copy the VGG-M[7] architecture till the fifth convolution layer (Conv5). This is followed by a deconvolution layer to upscale the feature maps. Then 1x1 convolutions are used to predict heat maps for each body point to be estimated. This network architecture is partly inspired from [71]. The predicted heat maps are 60x60 dimensional. The FLIC dataset contains annotations for the (x, y) coordinates of 9 points on the body (nose, shoulders, elbows, hips and wrists). Hence our architecture uses nine separate 1x1 convolutional filters in the last layer to predict the heat maps for each annotated point.

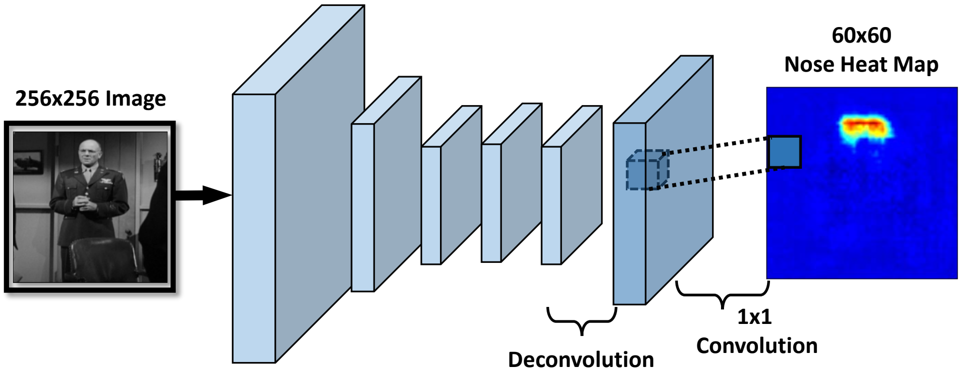


Figure 2.4.3: Architecture of the pose estimation network. First 5 layers copied from VGG-M, followed by a deconvolution layer. A 1x1 convolution layer is then used to predict each output heat map.

Preprocessing

Since the task we are evaluating is pose estimation (and not detection), we first need to crop the images around the annotated human. We do this by expanding the torso ground truth box by a fixed scale on all images. We then rescale all cropped images to 256x256. For each of the new cropped and rescaled images, we generate nine 60x60 ground truth heat maps, one for each of the joints. The heat map values are scaled between [-1,1] such that -1 represents background and +1 represents the presence of a body part. These ground truth heat maps are used to train the convolutional neural network. Since each ground truth heat map has only one positively activated pixel, the data is not sufficient to train the whole neural network. So we augment the data by activating a 3x3 neighbourhood in the heat maps.

Training

We use the Euclidean loss to compute the error signal for each output heat map in the Pose ConvNet. Since we have 9 ground truth heat maps, we have access to 9 error signals. We use standard backpropagation to train the network and average the gradients from all the nine euclidean losses. Training the Pose ConvNet directly using this procedure converges at predicting all pixels as -1 in the heat maps since the number of positive pixels are still very small in the ground truth. In order to overcome this, we reweigh the gradient w.r.t. a positive ground truth pixel by the inverse of number of total number of positive pixels and similarly for the negative pixels. This ensures that the sum of gradients for the positive pixels is equal to the sum of gradients for the negative pixels.

Evaluation

The trained Pose ConvNet maps are used to generate body part heat maps for each of the test images in the FLIC dataset. The highest scoring 20 pixels are identified in each heat map and the location of the centroid of these pixels is used as the prediction for that body part. Various evaluation metrics have been studied in the past for evaluating pose estimations methods [10, 58, 72]. We report accuracies using the Strict Percentage of Correct Parts(PCP) and the Percentage of Detected Joints (PDJ) metrics. We use the code made available by [10] to compute these metrics.

We train four models using the Pose ConvNet architecture to investigate the strength and transferability of our unsupervised representation. We test our unsupervised Appearance ConvNet by copying parameters to the first five convolutional layers of the Pose ConvNet and randomly initialising the last two layers. We then finetune the model on the training data from the FLIC dataset. We follow a similar procedure for the baseline model [78]. We also train an instance of the Pose ConvNet from scratch with random initialisation to compare with our model. The Strict PCP accuracies for these models are presented in Table 2.4.1 and the PDJ accuracies at varying precision values is presented in Table 2.4.2. The Appearance ConvNet beats the accuracy of the randomly initialised baseline by a large margin indicating that the Appearance ConvNet indeed learns a representation useful for Pose Estimation. We also observe a significant increase over the baseline unsupervised model [78] suggesting that the representation learnt by the Appearance ConvNet encodes properties not captured in the baseline. Surprisingly, we observe that when the Pose ConvNet is initialised with a model trained to perform action classification on the UCF101 dataset, it performs worse than random initialisation. This suggests the invariances learned due to semantic action supervision are not the right invariances for pose-estimation. Therefore, using an unsupervised model leads to unbiased and stronger results. In our experiments, we also observe that using Batch Normalization[35] while training the Pose ConvNet initialised with Appearance ConvNet leads to a very narrow increase in performance (1.5% in PCP).

Table 2.4.1: Results for the Strict PCP Evaluation for Pose Estimation on the FLIC Dataset

	Body Part	
Initialisation	**Upper Arms**	**Lower Arms**
Random	51.9	19.3
Wang et. al Unsupervised[78]	52.8	19.7
UCF101 Action Classification Pretrained	46.7	17.8
Ours	**57.1**	**24.4**
ImageNet Classification Pretrained	65.6	34.3

Table 2.4.2: Results for the PDJ Evaluation for Pose Estimation on the FLIC Dataset

		Elbow				Wrist			
Initialisation	Precision→	0.1	0.2	0.3	0.4	0.1	0.2	0.3	0.4
Random		20.0	47.3	63.9	74.8	17.2	36.1	49.6	60.8
UCF101 Pretrained		18.5	44.8	61.0	71.1	16.5	34.8	45.2	53.2
Wang et. al Unsupervised[78]		23.0	48.3	66.5	**77.6**	19.1	36.6	46.7	55.1
Ours		**28.0**	**54.6**	**68.8**	**77.6**	**20.1**	**40.0**	**51.6**	**60.8**
ImageNet Pretrained		34.8	62.0	74.7	82.1	29.0	48.5	59.3	66.7

2.4.3 Action Recognition

For the task of action recognition, we use the UCF101 and HMDB51 datasets. We test on split1 for both datasets since we use the same split to train our unsupervised models. UCF101 consists of 9537 train and 3783 test videos, each of which shows one of 101 actions. The HMDB51 dataset is a considerably smaller dataset which contains 3570 train and 1530 test videos and 51 possible actions. Due to the size of the HMDB51 dataset, overfitting issues are accentuated. Therefore, training deep models from scratch on this dataset is extremely difficult. In [62], the authors suggest multiple data augmentation techniques to alleviate these issues. In our experiments, we witnessed that initialising from our unsupervised model also helps in overcoming this issue to a certain extent which is reflected in the results. We also compare our results to [78] as before.

Similar to the Pose ConvNet, we use the Appearance ConvNet as an initialisation for action recognition to investigate its performance. We use the same architecture as the Appearance ConvNet(VGG-M till FC6) followed by two randomly initialised fully-connected layers at the end to perform classification. The first fully-connected layer has 2048 output neurons, and the second fully-connected has 101 output neurons for classification on UCF101 and 51 output neurons for classification on HMDB51.The softmax classification loss is used to train the action classification network. The input to the network is a random 224x224 crop from any frame in the video. During training, we use a batch size of 256, which gives us 256 crops of dimension 224x224 sampled from random videos. After intialising with the appropriate parameters, we train the whole model for 14k iterations using learning rate as 10^{-3} and for another 6k iterations using learning rate as 10^{-4}.

UCF101 And HMDB51

For testing the network, we uniformly sample 25 frames from the test video. From each of the 25 frames, we sample 224x224 crops from the corners and the center. We also generate flipped versions of each of these samples giving us 250 samples per video. We compute the predictions for each of the samples and average them across all samples for a video to get the final prediction. The classification accuracies on both datasets are reported in Table 2.4.3. We also present the results achieved by [62] for training from scratch and training from a network pretrained on ImageNet for classification. The results reflect improvement over training from random initialisation by significant margins - 12.3% on UCF101 and 7.2% on HMDB51. This clearly indicates that the Appearance ConvNet encodes transferable appearance features which are also useful for action

Table 2.4.3: Results for the Appearance Based action recognition on UCF101 and HMDB51

		Dataset	
Initialisation	Finetuning/Training	UCF101	HMDB51
Random	Full Network	42.5%	15.1%
Wang et. al Unsupervised[78]	Full Network	41.5%	16.9%
Ours	Full Network	**55.4%**	**23.6%**
Ours	Last 2 layers	41.4%	19.1%
ImageNet	Full Network	70.8%	40.5%

recognition. Surprisingly, finetuning just the last 2 fully connected layers also beats training from scratch on HMDB51 and scores comparably on the UCF101 dataset. This further emphasises the transferable nature of the Appearance ConvNet.

2.4.4 Static Image PASCAL Action Classification

For the second baseline model [37], classification accuracies are reported on the Pascal Action Classification dataset. The task involves classifying static images into one of the 10 action classes. The experiment used in [37], involves training the model using just 50 randomly sampled training images while simultaneously enforcing the prior they formulate. To allow fair comparison, we finetune our Appearance ConvNet using 50 randomly sampled images. We train an action classification network similar to the network described above but with 10 output neurons. The results for this experiment are reported in Table 2.4.4. The Appearance ConvNet shows an improvement of 2.5% over [37] on this task.

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Morbi commodo, ipsum sed pharetra gravida, orci magna rhoncus neque, id pulvinar odio lorem non turpis. Nullam sit amet enim. Suspendisse id velit vitae ligula volutpat condimentum. Aliquam erat volutpat. Sed quis velit. Nulla facilisi. Nulla libero. Vivamus pharetra posuere sapien.

Table 2.4.4: Results for action recognition accuracy in static images using just 50 randomly sampled training images from PASCAL VOC2010 dataset (mean over 5 runs)

Method	Accuracy
Random Initialisation (taken from [37])	15.34%
SSFA[37]	20.95%
Appearance ConvNet initialisation	**22.7%**

Nam consectetuer. Sed aliquam, nunc eget euismod ullamcorper, lectus nunc ullamcorper orci, fermentum bibendum enim nibh eget ipsum. Donec porttitor ligula eu dolor. Maecenas vitae nulla consequat libero cursus venenatis. Nam magna enim, accumsan eu, blandit sed, blandit a, eros.

.

3

Web & Turk: Constraint-transfer for Weakly Supervised Object Detection

3.1 Introduction

The collection of large annotated datasets of visual data has been the backbone for the recent success of computer vision algorithms. However, the process of collecting detailed annotations such as bounding boxes is often very expensive and time consuming. For example, annotating bounding boxes in images is known to be extremely time consuming with an average of 26-42 seconds per bounding-box on average [66]. Therefore, in recent years, there has been focus on moving from full-supervision to weaker forms of supervision such as image labels which are significantly cheaper to obtain. These image-level labels can be either annotated by humans (classic weakly-supervised) or

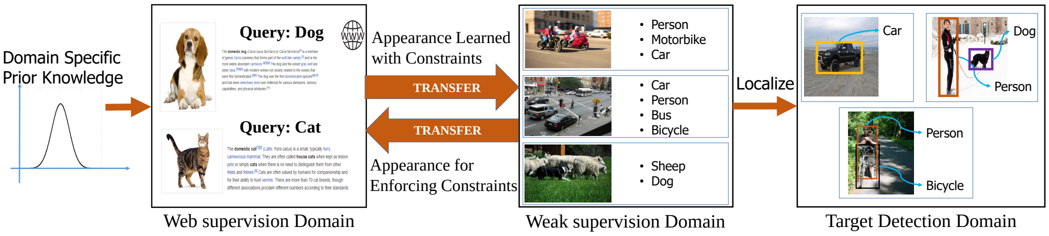


Figure 3.1.1: We propose a constrained-web and constrained-weakly supervised approach for training object detectors. Our approach uses web domain for leveraging the strong domain constraints to bootstrap the learning of detectors. These detectors are then learned to fine-tune on in-domain weakly supervised approaches which in turn improves the appearance constraints in the web domain.

extracted using web signals (web-supervised).

Both weak human-supervision and web-supervision have pros and cons. In weak human-supervision, there is no noise at image level labels. The domain shift between labeling and target domain can be also controlled by hand-picking images from the target domain for labeling. On the other hand, in case of web-supervision there is significant noise in the labeling and the domain shift is huge and often not under control. However, the amount of available data with free labels is enormous and often has some easy examples [11, 61] (object in focus with little or no background clutter). So, if our end-goal is to train the object detectors with least amount of annotations, what should be the right strategy? What is the right data to use for training the detectors?

In this paper, we exploit the advantages of both the domains to tackle the problem of weakly supervised learning. First, we make an observation that in the classic setting WSOD approaches end up learning to localize the most discriminative patches which might be a part of the object, or groups of the same object [4, 5, 29]. For example, for the class label 'dog', the models may localize the head of a dog since it always co-occurs with the 'dog' label. What we need is some extra constraint or knowledge to focus the attention on the right areas for learning.

Since we want to avoid additional human effort, do any such constraints exist that can be inferred without human annotations? Interestingly, we observe that while the easy images from web-supervision do not look realistic (Figure 3.1.2), there is a strong size constraint that can be applied while localizing the object (introduced in Section 3.3.3). We use this constraint to localize objects in web-domain and learn an appearance model for localization under these constraints. The learned appearance model can in turn be used to initialize Weakly Supervised Learning (WSL) in the target domain. On the other hand, the model trained in the weakly supervised setting learns to identify discriminative patches. Since the size constraint does not account for appearance, the WSL based model can in turn be used to create constraints that can also account for 'discriminative-ness'. Using this alternating strategy, we learn to transfer appearance constraints across domains while training for the goal of localizing objects.

In our experiments, we first observe that our proposed constrained learning approach not only outperforms methods using web-supervision, but also surprisingly achieves performances comparable to its weakly-supervised counterpart. We then show that our web-supervised framework can be used to augment training with weak supervision and no additional human effort. To the best of our knowledge, our proposed joint training approach also outperforms previous Weakly Supervised Object Detection (WSOD) methods on PASCAL VOC 2007.

In summary, our contributions are: (a) Learning localization of objects by transferring constraints across domains; (b) Formulating an approach for learning under constraints for WSOD; (c) Presenting a novel approach that combines web and weak-supervision that leads to the state of art performance on VOC2007 without any bounding box labels.

3.2 Related Work

Over the last decade, a lot of research efforts have focused on the problem of WSOD [5, 13, 29, 54, 74, 84]. Most approaches treat the task as a Multiple Instance Learning (MIL) [18, 63] problem, where each image is represented as a bag of object proposals. The underlying concept is that the image-level label indicates that the bag of regions contains at least one positive region corresponding to the label. For each class, all regions from the images not containing the label can be considered negatives. Therefore, detectors are trained by alternating between optimizing the parameters of the detector

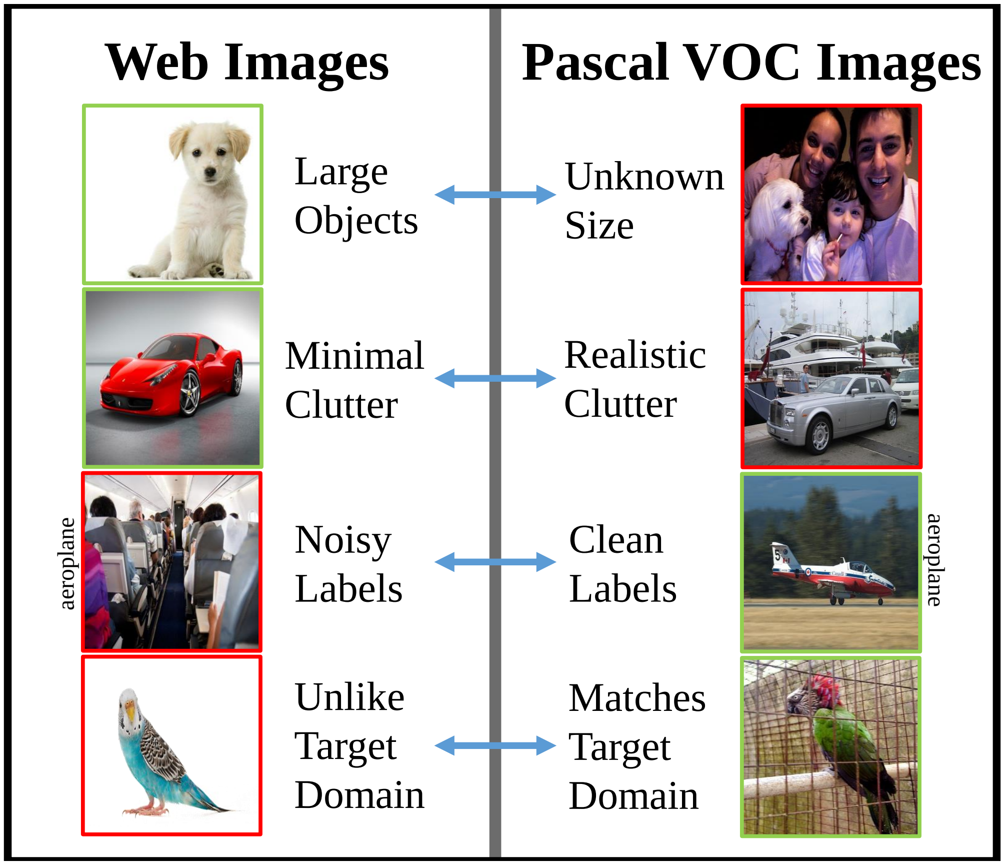


Figure 3.1.2: Learning localization in both domains has pros and cons. We propose an approach to make the best use of both domains.

and picking the best region from each bag. It is observed that MIL based approaches for weakly supervised object detection are extremely susceptible to the initialization. This leads to a lot of research efforts for intelligent initialization [17, 64] or multi-folding [29].

The advances in deep-learning has also inspired MIL based approaches which leverage convolutional neural network architectures as the object detector [55, 68, 85]. In [55], the authors investigate whether convolutional neural networks trained to perform image classification implicitly model the localization of objects. They use fully-convolutional networks in order to preserve the spatial correspondence to the inputs. While they show that such a framework does a good job in point-localization of objects, its performance on bounding box predictions is not as competitive. Recently, a novel architecture called the

Weakly Supervised Deep Detection Networks (WSDDN) [4] was proposed for jointly optimizing classification of region proposals and ranking of regions for each class. This architecture outperforms previous methods for weak supervision by large margins. We use the WSDDN architecture as a part of our proposed framework and explain it in greater detail in Section 2.3.

The key challenge that these approaches still face is that WSOD is an inherently ambiguous task. There are usually numerous patches whose occurrences correlate with the presence of some label. Most approaches end up localizing on one of these possible solutions which is manifested as detection of parts of objects. Numerous works have identified this issue and proposed approaches to overcome this shortcoming. ContextLocNet [42] builds on the WSDDN framework, by adding additional streams to capture the surrounding context for each region. Since we use the WSDDN framework as our base model, we believe that our proposed formulation can also be extended to the ContextLocNet framework. In [61], an estimate of the size of an object is used to guide the weakly supervised learning in an MIL framework and shows impressive results. While this works well, the learning of a regressor for size estimates requires additional annotated data. Our proposed framework also leverages the size bias in web-images but does not require any human supervision.

The ability to train computer vision models automatically on web data would dramatically reduce the need for human effort in annotation. This makes it an appealing research avenue in computer vision. In [12], web-data downloaded from Google and Flickr is used to build a knowledge base by automatically mining patterns in the visual data. [19] also proposes an approach to automatically extract visual knowledge from the web and learns to detect a wide range of variance associated with a query. In [11], an approach is proposed to train object detectors purely using web data. Mining web data has also been used in various other domains in computer vision like action recognition [25, 41, 53, 81]. A common observation among these works is that these approaches do not transfer directly to the domain of natural images. Especially, models that overfit to the domain bias in web images see a drop in performance when deployed in the wild. This is very frequently the case with deep neural networks trained on web-data [11]. In contrast to this, we propose to exploit the domain bias in web-images to guide the weakly supervised learning process.

The issue of transferability across domains not only exists in web images, but is also observed in other domains in computer vision. For example, the domain of rendered images, paintings, clipart etc. cannot be directly used to train models for deployment in the world. Since this is such a commonly encountered impediment, a lot of focus has been given to address this problem of *domain adaptation* [3, 14, 22, 30–32, 60, 67]. For example, Saenko et al. [60] address domain adaptation using metric learning approaches, and Gong et al. [30] propose to use geodesic flow kernel for unsupervised domain adaptation. To utilize web images, Bergamo and Torresani [3] adapt object classifiers learned from web images to target datasets with object-centric images. Sun et al. [67] and Duan et al. [22] propose to train video classifiers by domain transfer from web images. More recently, Tzeng et al. [73] show that convolutional neural network can learn domain-invariance features by introducing domain confusion losses. Gupta et al. [34] propose a cross modal distillation framework to learn visual representations for a new modality from visual representations from existing modalities.

3.3 Approach

We now describe our constraint-transfer approach for weakly supervised learning. First, we explain the preliminaries of weakly supervised learning and the base WSDDN model on which our approach builds upon. We then introduce the constrained setting and also introduce the web-supervised regime where the original constraints hold. Finally, we explain how to transfer the constraints and jointly perform the learning using both weakly and web-supervised data. We will also present the implementation details for facilitating replication of our experiments. The code for all the experiments shall also be made publicly available.

3.3.1 Preliminaries: Weakly-supervised learning

In the weakly supervised setting, we have access to image-level labels for each image indicating the presence or absence of objects of each class. The dataset of images can be

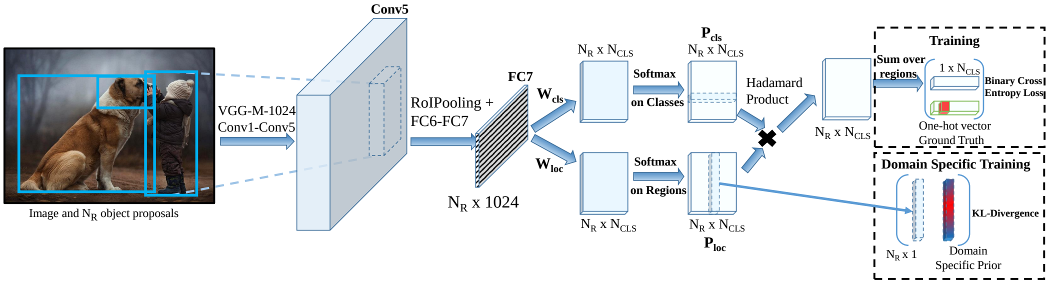


Figure 3.3.1: An overview of the Weakly Supervised Deep Detection Networks (WSDDN) [4] augmented with our domain specific training approach for transferring constraints across domains.

formally represented as a set of image and label vector pairs:
$\mathcal{D} = \{(I^1, y^1), (I^2, y^2)...(I^N, y^N)\}$ where I represents an image and $y \in \{0, 1\}^{N_{cls}}$ represents a one-hot encoding of the object classes present in the image. Generally, since our goal is to learn to localize objects, we represent each image as a set of multiple overlapping candidate regions. For example, in our implementations, we extract region proposals using EdgeBoxes [86]. In [86], low level cues are used to generate a set of candidate boxes for object detection. Therefore, we can represent each image as a bag of region proposals: $I^{(i)} = \{R_1^{(i)}, R_2^{(i)}, ..., R_{N_r^{(i)}}^{(i)}\}$ where $N_r^{(i)}$ is the number of region proposals extracted for the ith image.

In this setting, the task of training an object detector involves learning a scoring function $S(y_k = 1, R_p^{(i)})$ to score the association of a given region $R_p^{(i)}$ with object class k. Intuitively, the goal is to learn to identify discriminative patches whose occurrence correlates with the presence of class k. There have been several approaches in the past that focus on learning this scoring function directly from the image-level labels. Our approach builds upon the WSDDN framework which we now describe.

Weakly-supervised Deep Detection Networks

The base model used in [4] is a Convolutional Neural Network (CNN) with an architecture that overlaps significantly with the popular object detection model Fast-RCNN [27]. In Figure 3.3.1, we present an overview of the network architecture. Each image I is first processed using a series of Convolution, ReLu and Pooling operators leading to a spatial feature map. Given a region proposal R_i in the image, the corresponding region in the feature map can be extracted. These extracted feature maps can be projected to a fixed dimension by performing an adaptive pooling operation. This operation of extracting a feature of fixed dimension for each region proposal is commonly referred to as *RoIPooling* (proposed in [27]). Followed by RoI Pooling, we apply fully connected layers to finally yield a feature vector $\mathcal{F}_{\text{FC}}(I, R_i)$. Note we represent all the parameters in the CNN leading to $\mathcal{F}_{\text{FC}}$ by . In [27], the training is done using bounding box labels. However, in weakly-supervised case, we do not have labels at the level of bounding boxes. Therefore, [4, 42] use a two-stream architecture to train involving one classification stream and a localization stream.

The classification stream, consisting of a linear layer with parameters $\mathbf{W}_{\text{cls}} \in \mathbb{R}^{N_{\text{FC}} \times N_{\text{cls}}}$ and a softmax function, generates a probability distribution over classes for each region proposal as:

$$P_{\text{cls}}(y_k{=}1 \mid I, R_i) = \frac{\exp\left([\mathbf{W}_{\text{cls}}]_{*,\mathbf{k}}{}^{\top} \mathcal{F}_{\text{FC}}(I, R_i) \right)}{\sum_{j=1}^{N_{\text{cls}}} \exp\left([\mathbf{W}_{\text{cls}}]_{*,\mathbf{j}}{}^{\top} \mathcal{F}_{\text{FC}}(I, R_i) \right)}$$

On the other hand, the localization stream also consists of a linear layer with parameters $\mathbf{W}_{\text{loc}} \in \mathbb{R}^{N_{\text{FC}} \times N_{\text{cls}}}$ and a softmax function, but generates a probability distribution over region proposals for each class. Hence it is trying to 'localize' the best region for each class.

$$P_{\text{det}}(R^*{=}R_i \mid I, y_k{=}1) = \frac{\exp\left([\mathbf{W}_{\text{loc}}]_{*,\mathbf{k}}{}^{\top} \mathcal{F}_{\text{FC}}(I, R_i) \right)}{\sum_{j=1}^{N_{\text{r}}} \exp\left([\mathbf{W}_{\text{loc}}]_{*,\mathbf{k}}{}^{\top} \mathcal{F}_{\text{FC}}(I, R_j) \right)}$$

The final prediction score for each region proposal is the element-wise product of the classification and localization streams. We shall ignore the conditioning on I for ease of notation.

$$S(y_k{=}1, R^*{=}R_i) = P_{\text{cls}}(y_k{=}1 \mid R_i) * P_{\text{loc}}(R^*{=}R_i \mid y_k{=}1)$$

We can also obtain the image-wise class label by summing the scores over all regions. Therefore, $\hat{P}(y_k) = \sum_{i=1}^{N_r} \mathsf{S}(y_k{=}1, R^*{=}R_i)$ where $\hat{P}(y_k)$ is the score for the image-level class label. For training the network using backpropagation, we compute the loss between the predicted class labels with ground truth labels using the binary cross entropy loss

($\mathcal{L}_{\text{BCE}}$).

Issues with WSOD

Learning a scoring function without bounding box labels is that this is an inherently ambiguous task. For the occurrence of a class label, there could be multiple (possibly overlapping) patches that can associate and provide a discriminative signal. For example, Figure 3.4.1 (red dashed boxes) shows some localization errors using the WSDDN approach described above. Such errors where parts of object or groups of objects are labeled as whole object is very common in weakly-supervised learning. What we need is some strong prior or constraint to restrict the search space.

3.3.2 A New Domain: Web-supervised

Obtaining constraints in an unrestricted domain like VOC [24] or COCO [48] is extremely difficult. However, there might be other domains, where due to the domain bias, some constraints might exist. In this paper, we propose an approach to leverage the domain bias in web images to identify the extent of objects. Downloading images by querying the web also gives us access to weak labels since the query can be used as the label. Therefore, in our cross-domain setting, we have access to another freely available dataset for weak supervision which we represent as $\mathcal{D}' = \{(\mathcal{I}'^1, y^1), (\mathcal{I}'^2, y^2)...(\mathcal{I}'^N, y^N)\}$. In Figure 3.1.2, we present samples from both domains and present the key pros and cons of each domain for learning localization.

From Figure 3.1.2, it is evident that there is a noticeable difference between the type of images in both domains. In particular, the images from the web possess a characteristic domain bias of cleaner backgrounds and large centered objects. This domain bias has been shown to cause a negative effect on computer vision models trained on data automatically downloaded from the web [11]. We now show how this domain bias can in fact be exploited to guide the weakly supervised learning.

3.3.3 Web Constraints for WSOD

The model presented so far simply tries to localize the regions that maximize the classification and localization scores. In Section 3.4.5, we show qualitatively that this model very frequently localizes on parts/groups of objects. What is missing in these models is a guiding force that helps improve the tightness of a bounding box around an object. In this section, we propose an approach to leverage the domain bias in web images as this guiding force.

We have previously observed that images downloaded by querying the web often consist of a large centered object with minimal clutter in the background (see Figure 3.1.2). We first construct a simple prior based on the size of region proposals to favor larger predictions on web images. For a set of region proposals $\{R'_1, R'_2, \ldots, R'_{N_r}\}$ in a web image image I', we define the prior as:

$$P_{\text{pr}}(R^*{=}R_i) = \frac{\text{area}(R_i)}{\sum_{j=1}^{N_r} \text{area}(R_j)}$$

As explained in Section 3.3.1, the localization stream of WSDDN localizes an object class by generating a probability distribution over the region proposals for each class. Intuitively, we want the localization stream to favor larger objects on web images, in order to avoid localizing on parts of objects. To achieve this, we guide the localization stream by minimizing the distance between the predicted distribution P_{loc} and the prior P_{pr}. We formulate this as a minimization of the KL-divergence between the two distributions:

$$\mathcal{L}_{\text{cons}}(I', y) = \sum_{k=1}^{N_{\text{cls}}} y_k \, \mathrm{D}_{\text{KL}}\left(P_{\text{pr}}(R^*) \;\middle|\middle|\; P_{\text{loc}}(R^* \mid y_k{=}1)\right)$$

The above constraint is fairly strong since it does not account for the 'discriminative-ness' of the appearance of regions. We also observed in our experiments, that training our joint model (described in Section 3.3.4) using the above formulation led to degenerate solutions. This occurs possibly because the model quickly learns to imitate the prior since the target distribution P_{pr} doesn't depend on the appearance of the regions.

In order to retain the appearance information, we need to create a constraint which also accounts for the correlation of a region with a specific class. The model learned for localization using the binary cross entropy loss, possesses a weak estimate of this correlation. More specifically, the predicted distribution P_{loc} ranks the regions based on the association with the class label. Therefore, we first infuse the predicted distribution P_{loc} into the prior and use the new distribution to guide the localization. In this way, we transfer the appearance 'discriminative-ness' from the target domain back to the web-domain. Intuitively, the predicted distribution P_{loc} is first perturbed by incorporating the prior knowledge P_{pr}, and the new distribution is used as the target. This leads to the following formulation:

$$P^{\text{new}}_{\text{pr}}(R^*{=}R_i \mid y_k{=}1) = \frac{P_{\text{loc}}(R^*{=}R_i \mid y_k{=}1) * \text{area}(R_i)}{\sum_{j=1}^{N_r} P_{\text{loc}}(R^*{=}R_j \mid y_k{=}1) * \text{area}(R_j)}$$

$$\mathcal{L}_{\text{cons}}(I', y) = \sum_{k=1}^{N_{\text{cls}}} y_k \, \mathrm{D}_{\text{KL}}\left(P^{\text{new}}_{\text{pr}}(R^* \mid y_k{=}1) \,\middle\|\, P_{\text{loc}}(R^* \mid y_k{=}1)\right)$$

Note that in the computation of the gradient of $\mathcal{L}_{\text{cons}}$ for back-propagation, we treat $P^{\text{new}}_{\text{pr}}(R^* \mid y_k{=}1)$ as a constant target distribution.

Algorithm 1: Joint Constrained Training of WSOD

```
Initialize $\theta, \mathbf{W}_{\text{cls}}, \mathbf{W}_{\text{loc}}$;
Initialize $\mathcal{D}_{\text{BCE}} \subseteq \mathcal{D} \cup \mathcal{D}'$;
Initialize $\mathcal{D}_{\text{cons}} \subseteq \mathcal{D}'$;
while not converged do
    for $i = 1 : BCE_{iter}$ do
        $I, y \leftarrow \text{sample}(\mathcal{D}_{\text{BCE}})$;
        $\theta \leftarrow \theta - \eta * \frac{\partial \mathcal{L}_{\text{BCE}}(I, y)}{\partial \theta}$;
        $\mathbf{W}_{\text{cls}} \leftarrow \mathbf{W}_{\text{cls}} - \eta * \frac{\partial \mathcal{L}_{\text{BCE}}(I, y)}{\partial \mathbf{W}_{\text{cls}}}$;
        $\mathbf{W}_{\text{loc}} \leftarrow \mathbf{W}_{\text{loc}} - \eta * \frac{\partial \mathcal{L}_{\text{BCE}}(I, y)}{\partial \mathbf{W}_{\text{loc}}}$;
    end
    for $i = 1 : CONS_{iter}$ do
        $I, y \leftarrow \text{sample}(\mathcal{D}_{\text{cons}})$;
        $\mathbf{W}_{\text{loc}} \leftarrow \mathbf{W}_{\text{loc}} - \eta * \frac{\partial \mathcal{L}_{\text{cons}}(I, y)}{\partial \mathbf{W}_{\text{loc}}}$;
    end
end
```

Table 3.4.1: Average precision results for detection on Pascal VOC2007 test set using only data downloaded from the web.

Method		plane	bicycle	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbik	man	plant	sheep	sofa	train	TV	mean
$\mathcal{D}_{\text{BCE}}$	Constrained Init																					
a Web		40.6	37.2	28.1	26.2	11.6	51.5	31.7	28.0	11.0	27.6	12.1	15.3	34.9	41.1	10.8	13.7	28.0	32.8	43.3	25.6	27.6
b Web	✔	44.5	36.4	28.7	27.3	11.2	51.9	43.6	36.9	10.9	22.7	12.6	32.5	38.6	44.6	14.7	12.9	26.0	24.8	43.6	26.4	29.5
LEVAN		14	36.2	12.5	10.3	9.2	35.0	35.9	8.4	10.0	17.5	6.5	12.9	30.6	27.5	6.0	1.5	18.8	10.3	23.5	16.4	17.1
Webly Supervised		30.2	41.3	21.7	18.3	9.2	44.3	32.2	25.5	9.8	21.5	10.4	26.7	27.3	42.8	12.6	13.3	20.4	20.9	36.2	22.8	24.4

3.3.4 Joint Constrained Training for WSOD

The goal of this work is to transfer knowledge about localization from the web-domain to the domain of images with weak-supervision. We propose a joint training framework that combines the size constraint on web-domain with the weak-learning from both web and weak-supervision domain. It is important to note that the constraint loss $\mathcal{L}_{\text{cons}}$ makes sense only for images belonging to the web-domain. So jointly optimizing both objectives cannot be achieved by simply optimizing the sum of the loss functions for all the data. Instead, we create two sets of samples with weak supervision $\mathcal{D}_{\text{BCE}} \subseteq \mathcal{D} \cup \mathcal{D}'$ and $\mathcal{D}_{\text{cons}} \subseteq \mathcal{D}'$. The training is done using an alternating optimization by minimizing $\mathcal{L}_{\text{BCE}}$ on $\mathcal{D}_{\text{BCE}}$ and separately minimizing $\mathcal{L}_{\text{cons}}$ on $\mathcal{D}_{\text{cons}}$ using Stochastic Gradient Descent (SGD) in turns. In our experiments, we observed that, while minimizing $\mathcal{L}_{\text{cons}}$, it is important to fix the parameters of the CNN θ except parameters of the localization stream $\mathbf{W}_{\text{loc}}$. We hypothesize that this is necessary since otherwise the network is encouraged to learn low-level domain specific cues to minimize the domain specific loss $\mathcal{L}_{\text{cons}}$. In Algorithm 1, we summarize the strategy we propose for sampling and alternating between the two objectives to update the parameters of the weakly supervised object detector.

Table 3.4.2: Average precision results for detection on Pascal VOC2007 trainval set using only data downloaded from the web.

Method			plane	bicycle	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbik	man	plant	sheep	sofa	train	TV	mean
$\mathcal{D}_{\mathrm{BCE}}$	Constrained	Init																					
a Web			41.7	27.7	30.3	24.9	11.8	46.2	29.0	23.7	3.9	27.0	11.7	15.1	38.0	44.6	10.3	12.8	20.0	30.4	42.8	22.9	25.7
b Web	✓		49.2	27.7	31.3	21.9	11.4	48.0	39.1	39.3	4.6	24.0	12.4	30.9	39.2	47.5	14.9	13.0	19.5	22.7	41.1	22.0	28.0

Table 3.4.3: Average precision results for detection on Pascal VOC2007 test set using different settings of data and initialization

Method			plane	bicycle	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbik	man	plant	sheep	sofa	train	TV	mean
$\mathcal{D}_{\mathrm{BCE}}$	Constrained	Init																					
a Pascal			37.7	44.2	23.2	24.6	11.8	54.0	49.4	37.9	11.1	32.7	20.4	25.3	40.1	48.6	10.3	16.4	34.9	32.6	49.0	46.1	32.5
b Pascal	✓		41.4	45.1	25.9	21.7	13.0	56.8	47.6	43.4	10.1	32.0	35.1	40.8	44.5	47.0	9.7	12.8	34.5	34.4	47.3	47.2	34.5
c Pascal+Web			44.5	42.8	30.1	30.4	9.1	58.9	49.9	37.0	11.5	27.9	25.4	26.9	38.7	51.3	2.8	15.6	34.6	34.3	51.1	44.6	33.4
d Pascal+Web	✓		46.2	46.3	33.6	24.3	11.6	56.7	52.2	46.4	11.4	34.0	19.0	35.2	44.5	50.9	5.2	14.9	34.3	34.5	53.8	46.8	35.1
e Pascal		Web	45.8	47.3	31.2	28.8	13.9	60.2	49.4	36.1	12.5	32.5	32.6	26.9	43.3	53.8	6.8	18.1	34.2	36.6	49.2	52.7	35.6
f Pascal	✓	Web	44.3	43.0	30.6	25.6	13.6	56.3	48.9	45.1	6.6	33.9	31.7	41.2	43.4	52.5	10.7	14.8	32.1	36.5	50.1	52.7	35.7
g Pascal+Web		Web	44.9	43.2	32.4	28.0	13.8	59.2	50.9	37.4	11.5	31.6	32.7	29.0	43.8	52.8	2.7	16.8	34.1	41.6	48.7	48.9	35.2
h Pascal+Web	✓	Web	49.3	42.2	33.1	24.2	12.9	57.5	52.0	52.2	12.1	32.1	33.8	44.5	48.1	52.5	13.1	15.3	31.9	38.4	50.2	46.3	37.1
WSDDN-SSW-S [4, 42]			49.8	50.5	30.1	12.7	11.4	54.2	49.2	20.4	1.5	31.2	27.9	18.6	32.2	49.7	22.9	15.9	25.6	27.4	38.1	41.3	30.5
ContextLocNet [42]			57.1	52.0	31.5	7.6	11.5	55.0	53.1	34.1	1.7	33.1	49.2	42.0	47.3	56.6	15.3	12.8	24.8	48.9	44.4	47.8	36.3
WSOL using Size Estimate AlexNet* [61]																							36.0
WSOL using Size Estimate VGG-16*[61]																							37.2

Table 3.4.4: CorLoc [17] results for detection on Pascal VOC2007 trainval set using different settings of data and initialization.

Method			plane	bicycle	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbik	man	plant	sheep	sofa	train	TV	mean
$\mathcal{D}_{\mathrm{BCE}}$	Constrained	Init																					
a Pascal			60.0	54.7	47.7	44.1	26.5	67.2	69.4	56.0	14.9	65.1	23.7	43.7	65.2	71.8	29.5	37.7	64.6	41.1	63.5	59.7	50.3
b Pascal	✓		62.1	57.9	51.1	38.8	31.3	69.4	72.2	59.2	14.4	63.7	35.8	48.1	66.0	73.5	26.7	37.3	72.2	32.4	66.5	63.7	52.1
c Pascal+Web			62.8	56.2	57.7	41.5	28.8	70.8	68.8	53.2	12.7	63.4	29.0	45.9	66.1	79.8	8.3	34.7	68.8	44.0	69.2	65.8	51.4
d Pascal+Web	✓		70.2	56.2	48.8	45.7	28.6	67.9	73.5	64.0	13.1	64.1	39.1	58.2	68.7	81.4	22.1	37.4	70.8	43.6	63.5	66.2	54.1
e Pascal		Web	66.7	54.8	54.4	44.1	34.0	70.9	72.0	52.9	22.5	63.4	45.0	45.2	67.7	80.3	20.1	39.9	69.5	39.8	66.9	66.7	53.8
f Pascal	✓	Web	68.3	55.3	54.4	41.7	32.4	68.9	71.1	55.4	20.0	68.3	40.7	50.5	67.1	79.4	23.5	38.1	72.2	42.9	67.3	65.9	54.2
g Pascal+web		Web	68.5	50.6	56.9	43.5	31.2	68.0	72.4	52.9	15.0	66.2	45.9	45.7	64.3	80.7	9.4	38.6	66.7	46.0	65.4	66.7	52.7
h Pascal+web	✓	Web	68.5	53.3	54.7	43.9	29.5	67.3	72.5	66.3	20.9	66.2	42.7	55.0	64.8	78.1	15.5	34.9	70.1	40.4	63.9	65.3	53.7
WSDDN-SSW-S [4, 42]			80.4	62.4	53.8	28.2	26.0	68.0	72.5	45.1	9.3	64.4	38.8	35.6	51.4	77.1	37.6	38.1	66.0	31.2	61.6	53.0	50.0
ContextLocNet [42]			83.3	68.6	54.7	23.4	18.3	73.6	74.1	54.1	8.6	65.1	47.1	59.5	67.0	83.5	35.3	39.9	67.0	49.7	63.5	65.2	55.1
WSOL using Size Estimate AlexNet* [61]																							60.9
WSOL using Size Estimate VGG-16*[61]																							64.7

3.4 Experiments

In this section, we first briefly describe the human-annotated dataset PASCAL VOC 2007, the collection of web-data and other implementation details. We then present a quantitative evaluation of our constrained weakly supervised detection framework under

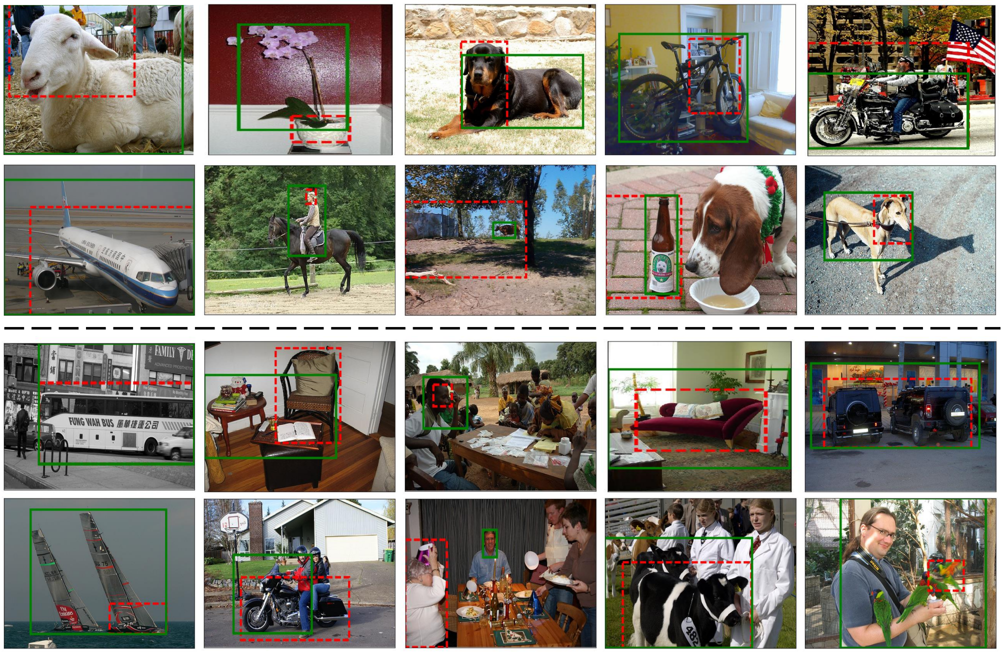


Figure 3.4.1: We present visualizations of predictions made by our best performing model (–) compared to the baseline WSDDN (−−). The top two rows show images where our model successfully localizes the object when the baseline models fail. The images in the bottom two rows represent cases where our model fails to correctly localize the object.

different settings of supervision. We also investigate some initialization strategies which help in improving the performance further. Finally, we perform a qualitative analysis of the predictions from our framework compared to the baseline models.

3.4.1 Datasets

For evaluating our weakly supervised object detector, we use the PASCAL VOC 2007 dataset [24] which contains 20 classes of objects. This is the most commonly used dataset to assess the performance of WSOD methods. While the dataset includes bounding box annotations, we simply use the image-level object class labels for training. In the testing phase, the mean average precision (mAP) metric is used to quantify the performance of object detectors. Weakly supervised object detectors are also evaluated using the

CorLoc [17] metric on the training set. The metric computes the AP on training images given the image-level class labels.

For the web-supervised setting, we download a large collection of images by querying Google image search. In [12], an approach is proposed to expand the list of queries for each class('Category Expansion'). This approach leads to queries like 'porsche' and 'ferrari' for the 'car' category. We follow the list of queries provided in the supplementary material of [12] to download images for each of the 20 PASCAL VOC categories. This gives us a dataset of 147128 images. In order to facilitate replication of our results, the list of queries and class-wise statistics of the dataset will be included in the supplementary material.

3.4.2 Implementation Details

The WSDDN [4] framework was implemented using the Caffe toolbox [39]. Note that we replace the spatial pyramid pooling layer with the RoI Pooling layer which serves the same purpose. We shall make our implementation accessible on the web to allow replication, comparison and benchmarking. In all our experiments, to optimize the binary cross entropy loss, we use SGD with a batch size of 1, learning rate 0.00001, weight decay 0.0005 and momentum 0.9. Note that the batch size of 1 still leads to ~2000 samples of regions. For PASCAL VOC 2007, we use the Edge Boxes made publicly available by [4]. We initialize our models by copying parameters from an ImageNet pretrained model (except when stated otherwise). These hyperparameters are the same as proposed in [4]. The training is done for 30 epochs while dropping the learning rate by a factor of 10 after 15 epochs. In all our experiments, we ensure that the number of iterations of SGD is the same. When images from the web are used (for either $\mathcal{L}_{\text{BCE}}$ or $\mathcal{L}_{\text{cons}}$), we ensure that each class is sampled with equal chance. In the setting where images from the web and PASCAL VOC are combined, at each iteration, we first randomly sample the domain and then sample images from the chosen domain. For optimizing the constraint loss, we use SGD with a batch size of 40, learning rate of 0.001 and keeping the remaining hyperparameters and sampling strategy same as before. We observed that BCE_{iter} in the range 1000-2000 with $\text{CONS}_{\text{iter}} = 100$ works best. These hyperparameters were manually tuned by visualizing the perturbed predicted distributions $P_{\text{pr}}^{\text{new}}$.

3.4.3 Web-supervised object detection

We first evaluate our framework in web-supervised setting. Here we only use web images and exclude all images collected with human annotations of PASCAL VOC. We train the WSDDN using the binary cross entropy loss $\mathcal{L}_{\text{BCE}}$ (refer to as *WSDDN-Web*). Surprisingly, we observe that this base WSDDN model directly outperforms all previous web-supervised object detection methods [12, 19]. Note that the models proposed in [12, 19] do not use an ImageNet based pretraining. In [12], they show that their initialization performs comparably to ImageNet pretraining when finetuned for object detection. So we can expect that the same trends hold considering the large difference in performance.

We also trained the model using our joint training strategy using the alternating optimization (again using only web images). In Table 3.4.1 and 3.4.2, we compare the detection mean average precision (mAP) for the two experiments. We witness an increase in performance of about 2% mAP indicating that the joint optimization with constraint loss contributes significantly to the localization using the same data. We observe that the 'car' and 'dog' classes show that largest gain in performance of 11.9% mAP and 17.2% mAP respectively. Incidentally, these classes have the highest number of images in the collection of web-images. This suggests the improvements might scale to some extent with the variation across web-images per class.

3.4.4 Cross-domain WSOD

We also conducted experiments to verify whether the constraint loss allows us to transfer information about localization to the target domain. For comparison, we first train the WSDDN framework using images from PASCAL VOC 2007. We observe that we achieve a performance of 32.5mAP (which is 0.2mAP lower compared to the publicly released matlab implementation [4]). The joint framework was then trained using PASCAL VOC for the binary cross entropy loss ($\mathcal{L}_{\text{BCE}}$) and the constraint loss on web-images ($\mathcal{L}_{\text{cons}}$). In Table 3.4.3 and Table 3.4.4, we again evaluate the performance of these models using the detection mAP on the 'test' set and CorLoc on the 'trainval' set.

For this experiment, we try all possible settings: (a) $\mathcal{L}_{\text{BCE}}$ on PASCAL Only (baseline WSDDN); (b) $\mathcal{L}_{\text{BCE}}$ on PASCAL, $\mathcal{L}_{\text{cons}}$ on Web; (c) $\mathcal{L}_{\text{BCE}}$ on PASCAL+Web, No $\mathcal{L}_{\text{cons}}$; (d) $\mathcal{L}_{\text{BCE}}$ on PASCAL+Web, $\mathcal{L}_{\text{cons}}$ on Web. As it can be seen using binary cross entropy loss on both PASCAL and Web images along with constraint loss on web images provides best performance. It can also be observed that the constraint loss enhances performance in all settings.

However, we still have not evaluated appearance transfer from Web to PASCAL images. The initialization in all the previous (a)-(d) was the ImageNet model. So, for the next experiment, we transfer appearance information from the web-domain by pretraining the WSDDN framework on web domain and finetune on the target domain. While this strategy is effective, we show that our proposed constraint loss provides a complementary contribution leading to further gains. We repeat the experiments in the settings described so far, but use the *WSDDN-Web* model as initialization instead of an ImageNet pretrained model. The results for these experiments can be found in Table 3.4.3 and Table 3.4.4 (Rows 5-8). Again: (e) $\mathcal{L}_{\text{BCE}}$ on PASCAL Only; (f) $\mathcal{L}_{\text{BCE}}$ on PASCAL, $\mathcal{L}_{\text{cons}}$ on Web; (g) $\mathcal{L}_{\text{BCE}}$ on PASCAL+Web, No $\mathcal{L}_{\text{cons}}$; (h) $\mathcal{L}_{\text{BCE}}$ on PASCAL+Web, $\mathcal{L}_{\text{cons}}$ on Web. We observe that initializing from the WSDDN-Web model gives us further improvements leading to a 4.6% mAP increase over the baseline WSDDN.

We also present results from other previous relevant works for comparison. ContextLocNet [42] is a recently proposed architecture for weakly supervised learning and also builds on the WSDDN framework. The key idea proposed in [42] is to utilize the features of the region surrounding each candidate box in the localization stream. While we outperform the detection mAP of [42], we believe that both ideas are quite complementary in nature. We can use context+web-domain constraints together to further improve weakly supervised learning. WSOL using Size Estimates [61] performed weakly supervised object detection using the size estimate of objects as a cue. For estimation of the size, they train a regressor using object size annotations for PASCAL VOC 2012. In contrast, we do not use any additional human annotations and achieve better detection performance compared to AlexNet and similar performance to VGG16 based models of [61]. While we achieve higher mAP on detection, [61] achieves much higher CorLoc. One possible hypothesis is that, compared to our simple prior, the accurate size cue leads to better localization at the cost of human annotation effort.

3.4.5 Qualitative Analysis

We observe from the quantitative analysis that the proposed joint training framework consistently outperforms training without constraints. We now qualitatively analyze the models trained with and without constrained training. In Figure 3.4.1, we present predictions from both models - base WSDDN trained on PASCAL VOC (red) and the best performing model using joint constrained training (green). It is evident from the predictions that the trained model is better at avoiding localization of parts.

A common failure mode of our framework is the localization of faces for 'person' class (Figure 3.4.1, Column 3, Row 3-4). Upon investigation, we realized that web-images for the 'person' class tend to cover only the face or upper body possibly due to a photography bias. This is also reflected in the lowered CorLoc performance in Table 3.4.4.

4
Conclusion

In this thesis, we presented two approaches to improve computer vision models using inexpensive forms of supervision. The first approach presents an unsupervised algorithm that takes advantage as the motion signal in videos as supervision to train an appearance representation. We train the unsupervised system on action videos in order to force the appearance representation to learn pose features. We demonstrate this property of the feature representation using qualitative results and quantitative results on Pose Estimation in the FLIC dataset, Action Recognition in videos on the UCF101 and HMDB51 datasets and still image action recognition on PASCAL VOC. The finetuning results emphasise the highly transferable nature of the representations learned. We compare to two other video-based unsupervised algorithms and show that our trained representation performs better consistently on these tasks. As a future goal, an interesting direction to pursue would be extending this method to generic videos. In the second part of the thesis, we present an approach to combine weakly-supervised learning from two-domains using constraint-transfer. Current WSOD approaches struggle to find right localization due to parts and groups of object being more discriminative. This paper exploits the benefits of two domains: web-supervised and weakly-supervised VOC. Specifically, it uses web-images which have minimal background clutter and strong size bias to learn an appearance model for objects. We use this model as initialization to perform joint weakly-supervised learning in both web-domain and PASCAL together. The weakly-supervised model is in turn used to improve the target for the constraint step by accounting for 'discriminative-ness'. We show that such an alternating training procedure using easy images can improve the performance on hard PASCAL images themselves. Our approach provides significant boost over the baseline WSDDN by almost 4.6% mAP and also reduces the gap between web-supervised and weakly supervised methods.

References

[1] Pulkit Agrawal, Joao Carreira, and Jitendra Malik. Learning to see by moving. 2015.

[2] Mykhaylo Andriluka, Leonid Pishchulin, Peter Gehler, and Bernt Schiele. 2d human pose estimation: New benchmark and state of the art analysis. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3686–3693. IEEE, 2014.

[3] Alessandro Bergamo and Lorenzo Torresani. Exploiting weakly-labeled web images to improve object classification: a domain adaptation approach. In *Advances in Neural Information Processing Systems (NIPS)*, pages 181–189, 2010.

[4] Hakan Bilen and Andrea Vedaldi. Weakly supervised deep detection networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2846–2854, 2016.

[5] Hakan Bilen, Marco Pedersoli, and Tinne Tuytelaars. Weakly supervised object detection with convex clustering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1081–1089, 2015.

[6] Charles F Cadieu and Bruno A Olshausen. Learning intermediate-level representations of form and motion from natural movies. *Neural computation*, 24 (4):827–866, 2012.

[7] Ken Chatfield, Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Return of the devil in the details: Delving deep into convolutional nets. *arXiv preprint arXiv:1405.3531*, 2014.

[8] Rizwan Chaudhry, Arunkumar Ravichandran, Georg Hager, and René Vidal. Histograms of oriented optical flow and binet-cauchy kernels on nonlinear dynamical systems for the recognition of human actions. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 1932–1939. IEEE, 2009.

[9] Chao-Yeh Chen and Kristen Grauman. Watching unlabeled video helps learn new human actions from very few labeled snapshots. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 572–579, 2013.

[10] Xianjie Chen and Alan L Yuille. Articulated pose estimation by a graphical model with image dependent pairwise relations. In *Advances in Neural Information Processing Systems*, pages 1736–1744, 2014.

[11] Xinlei Chen and Abhinav Gupta. Webly supervised learning of convolutional networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1431–1439, 2015.

[12] Xinlei Chen, Abhinav Shrivastava, and Abhinav Gupta. Neil: Extracting visual knowledge from web data. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1409–1416, 2013.

[13] David J Crandall and Daniel P Huttenlocher. Weakly supervised learning of part-based spatial models for visual object recognition. In *European Conference on Computer Vision*, pages 16–29. Springer, 2006.

[14] Hal Daume III and Daniel Marcu. Domain adaptation for statistical classifiers. *jair*, 26(1):101–126, 2006.

[15] Vincent Delaitre, Josef Sivic, and Ivan Laptev. Learning person-object interactions for action recognition in still images. In *Advances in neural information processing systems*, pages 1503–1511, 2011.

[16] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 248–255. IEEE, 2009.

[17] Thomas Deselaers, Bogdan Alexe, and Vittorio Ferrari. Weakly supervised localization and learning with generic knowledge. *International journal of computer vision*, 100(3):275–293, 2012.

[18] Thomas G Dietterich, Richard H Lathrop, and Tomás Lozano-Pérez. Solving the multiple instance problem with axis-parallel rectangles. *Artificial intelligence*, 89(1):31–71, 1997.

[19] Santosh K Divvala, Ali Farhadi, and Carlos Guestrin. Learning everything about anything: Webly-supervised visual concept learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3270–3277, 2014.

[20] Carl Doersch, Abhinav Gupta, and Alexei A Efros. Unsupervised visual representation learning by context prediction. In *ICCV*, 2015.

[21] Piotr Dollár, Vincent Rabaud, Garrison Cottrell, and Serge Belongie. Behavior recognition via sparse spatio-temporal features. In *Visual Surveillance and Performance Evaluation of Tracking and Surveillance, 2005. 2nd Joint IEEE International Workshop on*, pages 65–72. IEEE, 2005.

[22] Lixin Duan, Dong Xu, Ivor W Tsang, and Jiebo Luo. Visual event recognition in videos by learning from web data. *Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 34(9):1667–1680, 2012.

[23] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2010 (VOC2010) Results. http://www.pascal-network.org/challenges/VOC/voc2010/workshop/index.html.

[24] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. *IJCV*, 88(2):303–338, June 2010.

[25] Chuang Gan, Ting Yao, Kuiyuan Yang, Yi Yang, and Tao Mei. You lead, we exceed: Labor-free video concept learning by jointly exploiting web videos and images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 923–932, 2016.

[26] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. *ArXiv e-prints*, November 2013.

[27] Ross Girshick. Fast r-cnn. *arXiv preprint arXiv:1504.08083*, 2015.

[28] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2014.

[29] Ramazan Gokberk Cinbis, Jakob Verbeek, and Cordelia Schmid. Multi-fold mil training for weakly supervised object localization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2409–2416, 2014.

[30] Boqing Gong, Yuan Shi, Fei Sha, and Kristen Grauman. Geodesic flow kernel for unsupervised domain adaptation. In *CVPR*, 2012.

[31] Boqing Gong, Kristen Grauman, and Fei Sha. Learning kernels for unsupervised domain adaptation with applications to visual object recognition. *International Journal of Computer Vision (IJCV)*, 109(1):3–27, 2014.

[32] Raghuraman Gopalan, Ruonan Li, and Rama Chellappa. Domain adaptation for object recognition: An unsupervised approach. In *Proceedings of IEEE International Conference on Computer Vision (ICCV)*, pages 999–1006, 2011.

[33] Ross Goroshin, Joan Bruna, Jonathan Tompson, David Eigen, and Yann LeCun. Unsupervised learning of spatiotemporally coherent metrics. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4086–4093, 2015.

[34] Saurabh Gupta, Judy Hoffman, and Jitendra Malik. Cross modal distillation for supervision transfer. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[35] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.

[36] Dinesh Jayaraman and Kristen Grauman. Learning image representations equivariant to ego-motion. *arXiv preprint arXiv:1505.02206*, 2015.

[37] Dinesh Jayaraman and Kristen Grauman. Slow and steady feature analysis: Higher order temporal coherence in video. *arXiv preprint arXiv:1506.04714*, 2015.

[38] Shuiwang Ji, Wei Xu, Ming Yang, and Kai Yu. 3d convolutional neural networks for human action recognition. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 35(1):221–231, 2013.

[39] Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell. Caffe: Convolutional architecture for fast feature embedding. In *Proceedings of the ACM International Conference on Multimedia*, pages 675–678. ACM, 2014.

[40] Sam Johnson and Mark Everingham. Clustered pose and nonlinear appearance models for human pose estimation. In *Proceedings of the British Machine Vision Conference*, 2010. doi:10.5244/C.24.12.

[41] Armand Joulin, Laurens van der Maaten, Allan Jabri, and Nicolas Vasilache. Learning visual features from large weakly supervised data. In *European Conference on Computer Vision*, pages 67–84. Springer, 2016.

[42] Vadim Kantorov, Maxime Oquab, Minsu Cho, and Ivan Laptev. Contextlocnet: Context-aware deep network models for weakly supervised localization. In *European Conference on Computer Vision*, pages 350–365. Springer, 2016.

[43] Andrej Karpathy, George Toderici, Sanketh Shetty, Thomas Leung, Rahul Sukthankar, and Li Fei-Fei. Large-scale video classification with convolutional neural networks. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 1725–1732, 2014.

[44] Qifa Ke, Jia Deng, Simon Baker, and Michael Isard. Image retrieval using discriminative visual features, January 5 2016. US Patent 9,229,956.

[45] Alexander Klaser, Marcin Marszałek, and Cordelia Schmid. A spatio-temporal descriptor based on 3d-gradients. In *BMVC 2008-19th British Machine Vision Conference*, pages 275–1. British Machine Vision Association, 2008.

[46] Markus Kuderer, Shilpa Gulati, and Wolfram Burgard. Learning driving styles for autonomous vehicles from demonstration. In *Robotics and Automation (ICRA), 2015 IEEE International Conference on*, pages 2641–2646. IEEE, 2015.

[47] Hildegard Kuehne, Hueihan Jhuang, Estíbaliz Garrote, Tomaso Poggio, and Thomas Serre. Hmdb: a large video database for human motion recognition. In *ICCV,*, 2011.

[48] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Doll⊠r, and C. Lawrence Zitnick. Microsoft COCO: Common objects in context, 2014.

[49] Subhransu Maji, Lubomir Bourdev, and Jitendra Malik. Action recognition from a distributed representation of pose and appearance. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 3177–3184. IEEE, 2011.

[50] Julien Marzat, Yann Dumortier, and Andre Ducrot. Real-time dense and accurate parallel optical flow using cuda. 2009.

[51] Vincent Michalski, Roland Memisevic, and Kishore Konda. Modeling deep temporal dependencies with recurrent grammar cells"". In *Advances in neural information processing systems*, pages 1925–1933, 2014.

[52] Hossein Mobahi, Ronan Collobert, and Jason Weston. Deep learning from temporal coherence in video. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 737–744. ACM, 2009.

[53] David Novotny, Diane Larlus, and Andrea Vedaldi. Learning the structure of objects from web supervision. In *Computer Vision–ECCV 2016 Workshops*, pages 218–235. Springer, 2016.

[54] Maxime Oquab, Léon Bottou, Ivan Laptev, Josef Sivic, et al. Weakly supervised object recognition with convolutional neural networks. In *Proc. of NIPS*, 2014.

[55] Maxime Oquab, Léon Bottou, Ivan Laptev, and Josef Sivic. Is object localization for free?-weakly-supervised learning with convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 685–694, 2015.

[56] Xiaojiang Peng, Changqing Zou, Yu Qiao, and Qiang Peng. Action recognition with stacked fisher vectors. In *Computer Vision–ECCV 2014*, pages 581–595. Springer, 2014.

[57] Tomas Pfister, James Charles, and Andrew Zisserman. Flowing convnets for human pose estimation in videos. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1913–1921, 2015.

[58] Leonid Pishchulin, Arjun Jain, Mykhaylo Andriluka, Thorsten Thormählen, and Bernt Schiele. Articulated people detection and pose estimation: Reshaping the future. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 3178–3185. IEEE, 2012.

[59] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *arXiv preprint arXiv:1409.0575*, 2014.

[60] Kate Saenko, Brian Kulis, Mario Fritz, and Trevor Darrell. Adapting visual category models to new domains. In *Proceedings of European Conference on Computer Vision (ECCV)*, pages 213–226, 2010.

[61] Miaojing Shi and Vittorio Ferrari. Weakly supervised object localization using size estimates. In *European Conference on Computer Vision*, pages 105–121. Springer, 2016.

[62] Karen Simonyan and Andrew Zisserman. Two-stream convolutional networks for action recognition in videos. In *NIPS*, 2014.

[63] Parthipan Siva and Tao Xiang. Weakly supervised object detector learning with model drift detection. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 343–350. IEEE, 2011.

[64] Parthipan Siva, Chris Russell, Tao Xiang, and Lourdes Agapito. Looking beyond the image: Unsupervised learning for object saliency and detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3238–3245, 2013.

[65] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402*, 2012.

[66] Hao Su, Jia Deng, and Li Fei-Fei. Crowdsourcing annotations for visual object detection. In *Workshops at the Twenty-Sixth AAAI Conference on Artificial Intelligence*, volume 1, 2012.

[67] Chen Sun, Sanketh Shetty, Rahul Sukthankar, and Ram Nevatia. Temporal localization of fine-grained actions in videos by domain transfer from web images. In *ACM Multimedia*, 2015.

[68] Chen Sun, Manohar Paluri, Ronan Collobert, Ram Nevatia, and Lubomir D. Bourdev. Pronet: Learning to propose object-specific boxes for cascaded neural networks. *CVPR*, 2016.

[69] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. *arXiv preprint arXiv:1409.4842*, 2014.

[70] Jonathan Tompson, Arjun Jain, Yann Lecun, and Christoph Bregler. Joint training of a convolutional network and a graphical model for human pose estimation. *NIPS*, 2014.

[71] Jonathan Tompson, Ross Goroshin, Arjun Jain, Yann LeCun, and Christoph Bregler. Efficient object localization using convolutional networks. In *ICCV*, 2015.

[72] Alexander Toshev and Christian Szegedy. Deeppose: Human pose estimation via deep neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1653–1660, 2014.

[73] Eric Tzeng, Judy Hoffman, Trevor Darrell, and Kate Saenko. Simultaneous deep transfer across domains and tasks. In *ICCV*, 2015.

[74] Chong Wang, Weiqiang Ren, Kaiqi Huang, and Tieniu Tan. Weakly supervised object localization with latent category learning. In *European Conference on Computer Vision*, pages 431–445. Springer, 2014.

[75] Heng Wang and Cordelia Schmid. Action recognition with improved trajectories. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3551–3558, 2013.

[76] Heng Wang, Muhammad Muneeb Ullah, Alexander Klaser, Ivan Laptev, and Cordelia Schmid. Evaluation of local spatio-temporal features for action recognition. In *BMVC*. BMVA Press, 2009.

[77] Heng Wang, Alexander Kläser, Cordelia Schmid, and Cheng-Lin Liu. Action recognition by dense trajectories. In *CVPR*. IEEE, 2011.

[78] Xiaolong Wang and Abhinav Gupta. Unsupervised learning of visual representations using videos. In *ICCV*, 2015.

[79] Xiaolong Wang, Ali Farhadi, and Abhinav Gupta. Actions ~ transformations. *CoRR*, abs/1512.00795, 2015.

[80] Xiaolong Wang, David Fouhey, and Abhinav Gupta. Designing deep networks for surface normal estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 539–547, 2015.

[81] Zhe Xu, Shaoli Huang, Ya Zhang, and Dacheng Tao. Webly-supervised fine-grained visual categorization via deep domain adaptation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2016.

[82] Weilong Yang, Yang Wang, and Greg Mori. Recognizing human actions from still images with latent poses. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 2030–2037. IEEE, 2010.

[83] Cong Zhang, Hongsheng Li, Xiaogang Wang, and Xiaokang Yang. Cross-scene crowd counting via deep convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 833–841, 2015.

[84] Yimeng Zhang and Tsuhan Chen. Weakly supervised object recognition and localization with invariant high order features. In *BMVC*, pages 1–11, 2010.

[85] B. Zhou, A. Khosla, Lapedriza. A., A. Oliva, and A. Torralba. Learning Deep Features for Discriminative Localization. *CVPR*, 2016.

[86] C Lawrence Zitnick and Piotr Dollár. Edge boxes: Locating object proposals from edges. In *European Conference on Computer Vision*, pages 391–405. Springer, 2014.