NYC3DCars: A Dataset of 3D Vehicles in Geographic Context Supplemental Material

Kevin Matzen
Cornell University

Noah Snavely
Cornell University

In this supplemental material we present additional details about our paper. Section 1 provides a more thorough discussion of the Weak-Label Structural SVM used for our viewpoint-aware detector. Section 2 details how we compute the 3D pose of each vehicle hypothesis in the geographic context rescoring stage of our pipeline. Finally, Section 3 provides a few examples illustrating the complexity of our NYC3DCars dataset. Please refer to the included video for a recorded annotation session, as well as additional illustrations of our dataset and evaluation.

1. Weak-Label Structural SVM

This section provides a more detailed overview of the Weak-Label Structural SVM (WL-SSVM) as used in our baseline viewpoint-aware vehicle detectors. We begin with a definition of the WL-SSVM, as a recap of [3], followed by a discussion of its high-level goals. Next, we describe the optimization procedure used in training. Finally, we provide implementation-specific details. For an authoritative account of the WL-SSVM, please see Girshick *et al*., Sections 3 and 4 [3].

Definition. Let $\mathcal{X}$ be an input space, $\mathcal{Y}$ be a training label space, and $\mathcal{S}$ be a prediction space, where $\mathcal{Y}$ is not necessarily equal to $\mathcal{S}$, as is the case with weakly labeled data (i.e., where certain types of annotations may be missing; in our case, some training data may not come equipped with viewpoint annotations, for instance.) We begin with a classification function mapping inputs in $\mathcal{X}$ to predictions in $\mathcal{S}$:

$$f(x) = \arg\max_{s \in \mathcal{S}(x)} \mathbf{w} \cdot \mathbf{\Phi}(x, s) \tag{1}$$

where $\mathbf{\Phi}(x, s)$ is a joint feature map dependent on both the input, x, and the prediction, s. $\mathbf{w}$ is a vector of model parameters, and the goal of training is to learn these parameters.

The WL-SSVM is defined by the training problem

$$\min_{\mathbf{w}} \frac{1}{2}\|\mathbf{w}\|^2 + C \sum_{i=1}^{n} L'(\mathbf{w}, x_i, y_i) \tag{2}$$

where

$$L'(\mathbf{w}, x, y) = \underbrace{\max_{s \in \mathcal{S}(x)} [\mathbf{w} \cdot \mathbf{\Phi}(x, s) + L_{\text{margin}}(y, s)]}_{3a} - \underbrace{\max_{s \in \mathcal{S}(x)} [\mathbf{w} \cdot \mathbf{\Phi}(x, s) - L_{\text{output}}(y, s)]}_{3b} \tag{3}$$

The goal of this objective function is to encourage the maximizer of 3a to be a bad prediction and to encourage the maximizer of 3b to be a good prediction so that the score associated with 3a is pulled down below the score associated with 3b. This notion of "goodness" is task-specific. For example, as we will show in the next section, there are choices for these two loss functions that are strict in their definition of goodness, in that the prediction must match the label exactly. However, one can also encode some leniency into these loss function, which is necessary when working with weakly-labeled data.

Discussion. In order to provide some intuition about this general objective function, we first describe how to represent the special cases of the Structural SVM and then the Latent Structural SVM using the WL-SSVM framework. Then we will revisit our viewpoint-aware loss functions, $L_{\text{margin}}(y, s)$ and $L_{\text{output}}(y, s)$, defined in the paper, and discuss how they impact our optimization problem.

As a reminder, our training labels are defined as $y = (y^l, y^b, y^v) \in \mathcal{Y}$ where $y^l \in \{+1, -1\}$ indicates a positive/negative example, $y^b \in \mathbb{R}^4$ is a 2D bounding box, and $y^v \in \{1, \ldots, K, \varnothing\}$ is a viewpoint class. Our predictions are similarly defined as $s = (s^l, s^b, s^v) \in \mathcal{S}$ where $s^l \in \{+1, -1\}$, $s^b \in \mathbb{R}^4$, and $s^v \in \{1, \ldots, K\}$. Our viewpoint-aware loss function, $L_{\text{view},l,\tau}(y, s)$, is defined

in the paper as

$$L_{l,\tau}(y,s) = \begin{cases} l & y^l = -1 \wedge s^l = +1 \\ 0 & y^l = -1 \wedge s^l = -1 \\ l & y^l = +1 \wedge \text{overlap}(y^b, s^b) < \tau \\ 0 & y^l = +1 \wedge \text{overlap}(y^b, s^b) \geq \tau \end{cases} \tag{4}$$

$$L_{\text{view},l,\tau}(y,s) = \begin{cases} \frac{1}{2}\left[l + L_{l,\tau}(y,s)\right] & y^v \neq s^v \wedge y^v \neq \varnothing \\ \frac{1}{2}L_{l,\tau}(y,s) & y^v = s^v \wedge y^v \neq \varnothing \\ L_{l,\tau}(y,s) & y^v = \varnothing \end{cases} \tag{5}$$

where $\text{overlap}(y^b, s^b) = \frac{\|y^b \cap s^b\|}{\|y^b \cup s^b\|}$. We use $L_{\text{margin}}(y,s) = L_{\text{view},1,0.5}(y,s)$ and $L_{\text{output}}(y,s) = L_{\text{view},\infty,0.7}(y,s)$. This choice of $L_{\text{output}}(y,s)$ means the maximizer of 3b must have a correct viewpoint prediction and must have a bounding box overlap of more than 0.7. The choice of $L_{\text{margin}}(y,s)$ means the maximizer of 3a might have a viewpoint misclassification or poor localization and when this is the case, the score of this prediction be pulled down below the score of 3b's maximizer.

The Structural SVM (SSVM) training objective function is defined in [4] as:

$$\min_{\mathbf{w}} \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{n} L'(\mathbf{w}, x_i, y_i) \tag{6}$$

where

$$L'(\mathbf{w},x,y) = \max_{\hat{y}\in\mathcal{Y}} \left[\mathbf{w}\cdot\mathbf{\Phi}(x,\hat{y}) + L(y,\hat{y})\right] - \mathbf{w}\cdot\mathbf{\Phi}(x,y) \tag{7}$$

and $L : \mathcal{Y}\times\mathcal{Y} \to \mathbb{R}_+$ is a loss function measuring compatibility between training labels and predicted labels. Note that the label and prediction spaces are equal in the traditional formulation of the SSVM, that is $\mathcal{S} = \mathcal{Y}$. It is easy to encode this in the WL-SSVM framework by taking

$$L_{\text{margin}}(y,\hat{y}) = L(y,\hat{y}) \tag{8}$$

$$L_{\text{output}}(y,\hat{y}) = \begin{cases} 0 & y = \hat{y} \\ \infty & y \neq \hat{y} \end{cases} \tag{9}$$

In short, the choice of $L_{\text{output}}(y,\hat{y})$ forces 3b to choose only assignments with predictions that equal the training labels.

However, the SSVM has two shortcomings. First, our label space does not equal our prediction space. Some training labels do not include viewpoint, whereas we require all predictions to include viewpoint. Second, for one particular label, there might be many compatible predictions. In our case, the goal is to build a classifier where the *car* or *not car* prediction is primary and the viewpoint prediction is secondary. This is subtly different from posing the problem of viewpoint-aware vehicle detection as a multiclass problem where each view is a separate class, trained independently using opposing viewpoints as negative examples.

Another special case of the WL-SSVM is the Latent Structural SVM (LSSVM). This problem is defined by the training objective

$$\min_{\mathbf{w}} \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{n} L'(\mathbf{w}, x_i, y_i) \tag{10}$$

where

$$\begin{aligned} L'(\mathbf{w},x,y) = &\max_{(\hat{y},\hat{z})\in\mathcal{Y}\times\mathcal{Z}} \left[\mathbf{w}\cdot\mathbf{\Phi}(x,(\hat{y},\hat{z})) + L(y,(\hat{y},\hat{z}))\right] \\ &- \max_{\hat{z}\in\mathcal{Z}} \left[\mathbf{w}\cdot\mathbf{\Phi}(x,(y,\hat{z}))\right] \end{aligned} \tag{11}$$

and $L : \mathcal{Y}\times(\mathcal{Y}\times\mathcal{Z}) \to \mathbb{R}_+$. Similar to the SSVM, the LSSVM can be encoded as a WL-SSVM with

$$L_{\text{margin}}(y,(\hat{y},\hat{z})) = L(y,(\hat{y},\hat{z})) \tag{12}$$

$$L_{\text{output}}(y,(\hat{y},\hat{z})) = \begin{cases} 0 & y = \hat{y} \\ \infty & y \neq \hat{y} \end{cases} \tag{13}$$

This formulation addresses one of the aforementioned problems, namely that we are now able to apply a loss using latent variables as opposed to just the binary label variables. However, this strict L_{output} still prevents us from considering cases where the prediction doesn't match the label exactly as being a good prediction.

The WL-SSVM loss functions above overcome both issues. First, L_{output} is not only dependent on the class prediction, y^l, but also the secondary predictions, y^b and y^v. Second, it supports $\mathcal{Y} \neq \mathcal{S}$.

Implementation. The implementation of the system used in our paper is based off of the **voc-release5** system [2]. In particular, we made modifications to the loss adjustment functions and the loss pyramid. We used all default settings except for the number of loss-adjusted detections used in the convex step of the concave-convex optimization procedure. Please see [3] for the details of this procedure. We used the value recommended in [3], $M = 1$, of loss-adjusted detections mined from positive training examples. When $M = 0$, this system reduces to using the Latent SVM of [1] which is what we use to report LSVM results. In our case, both LSVM and WL-SSVM training procedures are identical until a final round of training is performed. This final round consists of one positive relabeling phase followed by five iterations of negative data mining. The difference is that the LSVM used $M = 0$ and the WL-SSVM used $M = 1$ in this final round. Therefore, both methods receive the same number of training iterations.

2. Vehicle Pose Estimation

The 3D vehicle pose estimation stage used in our geographic context rescoring algorithm (Section 5 of the paper) requires solving for four pose parameters, given a 2D detection and a exemplar car 3D model. A 3D vehicle pose, $\vec{v}$, is parameterized as $\vec{v} = (v_\phi, v_\lambda, v_\theta, v_e)$ where (v_ϕ, v_λ) denotes the geodetic coordinates (latitude and longitude) of the vehicle on the Earth's surface, v_θ is the rotation of the vehicle about the scene's local up vector, and v_e is the elevation of the vehicle with respect to our world coordinate system's vertical datum. This section describes how we convert a 2D bounding box and viewpoint class, produced by our baseline detector, to a set of hypothesized 3D vehicle poses.

Orientation. We start by computing orientation, v_θ. The viewpoint class (discretized azumithal angle) given by the 2D detection is combined with the camera's rotation about the scene's up vector (i.e., the camera heading) to obtain the rotation of the vehicle in world coordinates.

3D position. The second step is to jointly solve for suitable values of (v_ϕ, v_λ) and v_e (i.e., latitude, longitude, and elevation). The goal is to place the 3D vehicle in a location such that the projection of its 3D bounding cuboid tightly fits the 2D bounding box on the image plane. Because this projection may not match the aspect ratio of the 2D bounding box exactly, we estimate two positions, one where the top and bottom of the projection of the (vehicle-aligned) 3D bounding cuboid touch the top and bottom of the detection frustum, and one where the left and right edges of the projection of the 3D bounding cuboid touch the left and right sides of the detection frustum. These two positions, illustrated in Figure 1, are then averaged together to yield a final position. We estimate one such translation for each 3D vehicle class exemplar (each of which has a different 3D bounding cuboid). The exemplars we use include a sedan, pickup truck, minivan, SUV, Jeep, and a hatchback.

3. Dataset Examples

To illustrate the content of NYC3DCars, we show a selection of photographs from the dataset and several examples of 3D annotations. These examples demonstrate the diversity of the dataset, as well as the utility of our annotation interface.

Figure 2 shows several photos in our dataset, along with annotations showing the camera orientation as solve for by structure from motion (SfM). Figures 3, 4, and 5 include the input photograph with 2D bounding boxes for each annotated vehicle as well as a map depicting the camera frustum. Additionally, several example vehicles with 3D models are presented for each photograph. Finally, Figure 6 shows examples of cropped, annotated vehicles from various photos.

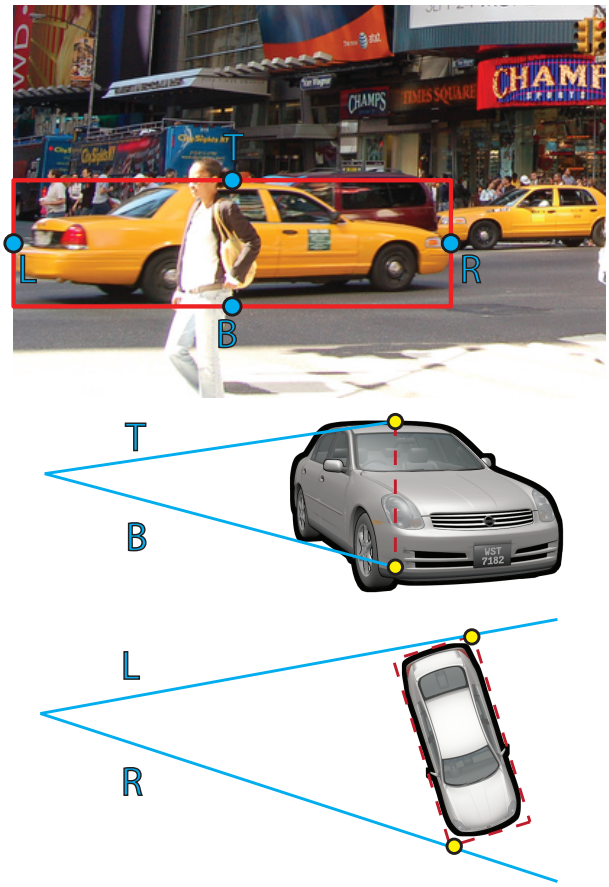

Figure 1. **Vehicle 3D pose estimation.** Top: A detection's bounding box. The rays labeled T, B, L, and R pass through the centers of the 2D bounding box sides. Middle: T and B intersect the centers of the top and bottom faces of the 3D bounding cuboid. Bottom: L and R intersect the extremal points of the rotated 3D bounding cuboid. These two poses are averaged together to get the final pose. The 3D exemplar illustrated here is the "sedan" exemplar.

References

[1] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part-based models. *PAMI*, 32, 2010.

[2] R. B. Girshick, P. F. Felzenszwalb, and D. McAllester. Discriminatively trained deformable part models, release 5. http://people.cs.uchicago.edu/~rbg/latent-release5/.

[3] R. B. Girshick, P. F. Felzenszwalb, and D. A. McAllester. Object detection with grammar models. In *NIPS*, 2011.

[4] C.-N. J. Yu and T. Joachims. Learning structural svms with latent variables. In *ICML*, 2009.

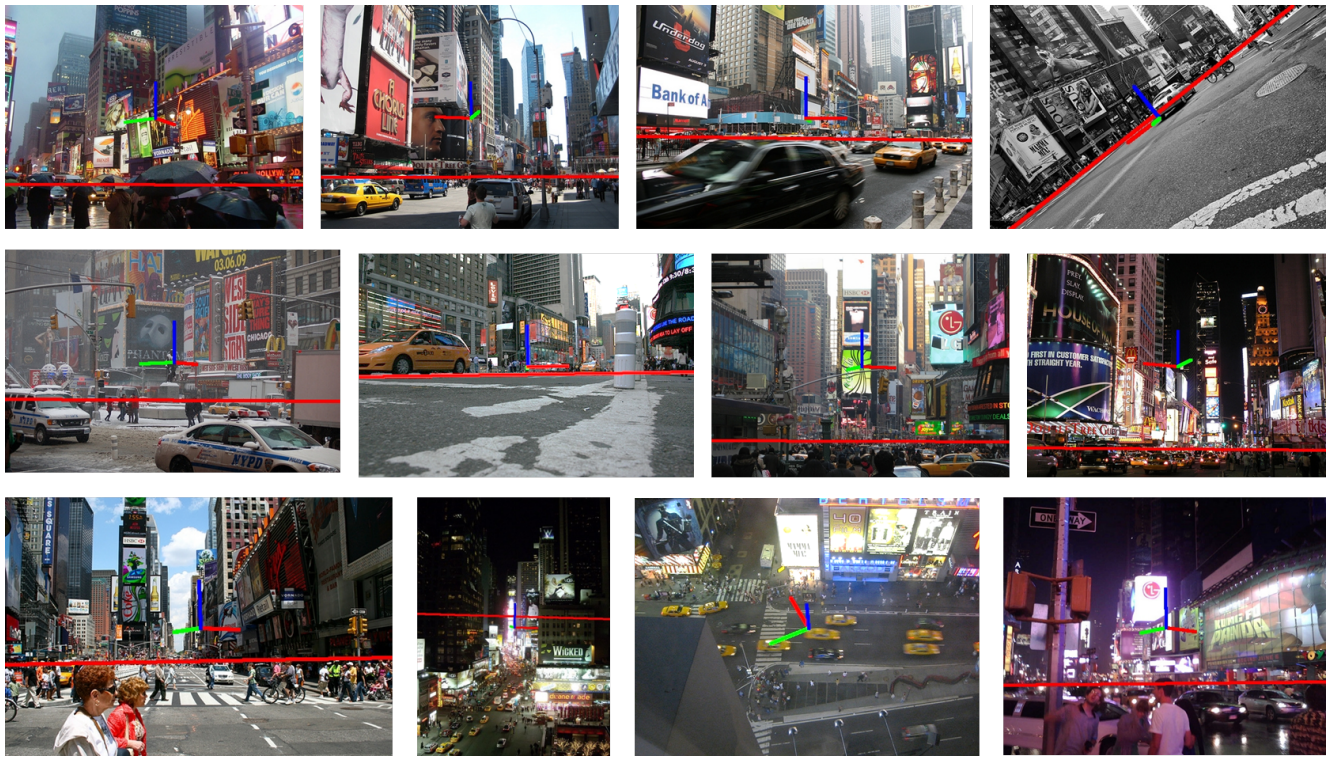

Figure 2. **Example photos from NYC3DCars.** The images in our dataset represent a wide variety of viewpoints (e.g., top-down vs. ground-level), times (e.g., day vs. night), weather conditions (sunny vs. rainy vs. snowy), and levels of clutter and occlusion (e.g., crowded photos). Here, each photo is shown annotated with viewpoint information derived from SfM, including a red line showing the horizon, and a set of axes showing global orientation (red: east, green: north, blue: up).

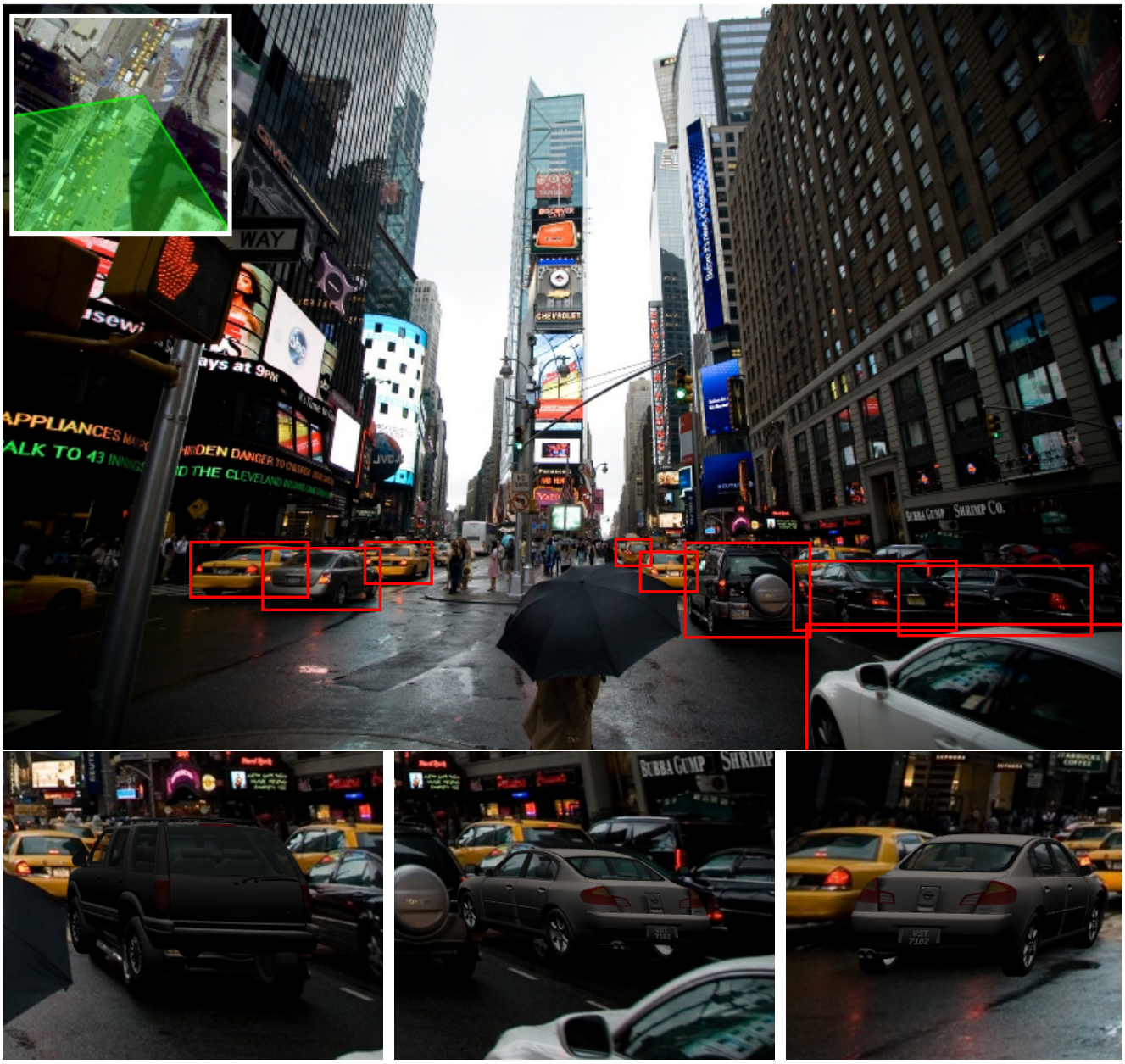

Figure 3. **Example 1.** A wide-angle daytime photograph. The photograph is annotated with bounding boxes around each labeled vehicle, and several 3D ground truth labeled car poses are shown underneath. The estimated pose of the camera is shown in the overlay at the top left. Note that even with a large degree of perspective distortion, the labeler was able to accurately annotate the photograph, because we use the recovered camera intrinsics in the 3D annotation tool.

Figure 4. **Example 2.** A daytime photograph exhibiting a wide variety of occlusion patterns including vehicle-vehicle occlusion and vehicle-pedestrian occlusion.

Figure 5. **Example 3.** Nighttime photograph facing south. The photographer is standing on the bleachers in order to capture this elevated shot. Note that the unannotated vehicles on the left are limousines (labelers were requested not to annotate such vehicles).

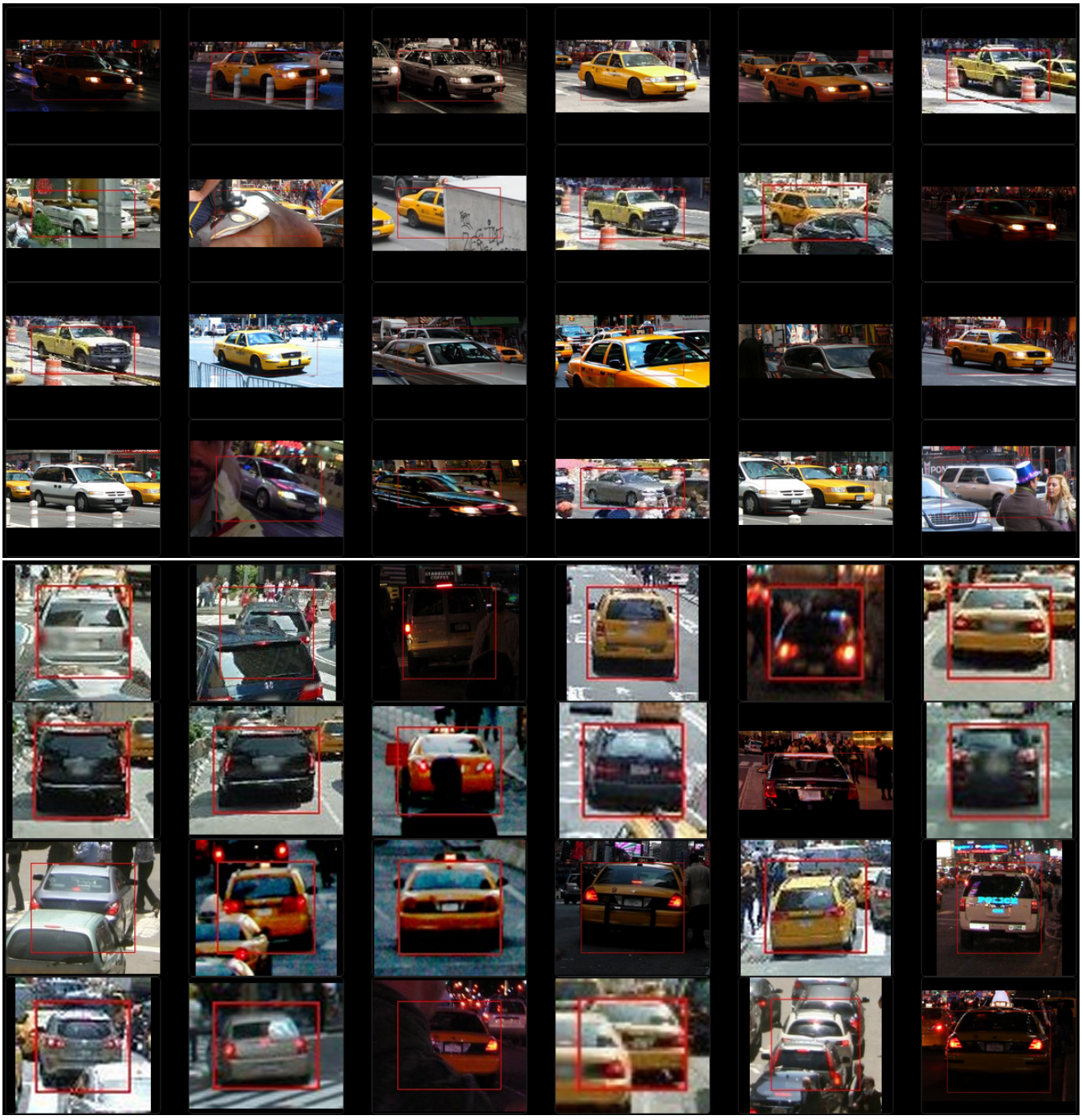

Figure 6. **Annotations queried by viewpoint.** The most closely matching 24 annotations are shown, based on viewpoint similarity to two query viewpoints. Top: Vehicles viewed from the front right. Bottom: Vehicles viewed from behind.