Multiperspective Automotive Labeling

Luke Jacobs
Electrical Engineering
University of Illinois at Urbana Champaign
Champaign, IL, USA
Email: lukedj2@illinois.edu

Akhil Kodumuri
Computer Engineering
University of Illinois at Urbana Champaign
Champaign, IL, USA
Email: akhilvk2@illinois.edu

Jim James
Metea Valley High School
Aurora, IL, USA
Email: jimtjames@gmail.com

Seongha Park and Yongho Kim
Mathematics and Computer Science
Argonne National Laboratory
Lemont, IL, USA
Email: {seongha.park, yongho.kim}@anl.gov

***Abstract*—Supervised machine learning techniques inherently rely on datasets to be trained. With image datasets traditionally being annotated by humans, many advancements in image annotation tools have been made to ensure creation of rich datasets with accurate labels. Nevertheless, users still find it challenging to create and use their own datasets with labels that reflect their problem domain. We propose a streamlined labeling process that aligns multiperspective images and allows a transition from a labeled perspective to other perspectives. The main goal of this work is to reduce the human effort required for labeling vehicle images under favorable conditions where the image perspectives are correlated and one or more perspectives are known. A case study is described and analyzed to show the effectiveness of the process, as well as constraints and limitations when applied to other cases.**

***Keywords*-training dataset, multiperspective images, supervised machine learning, automotive labeling, supervised learning**

I. Introduction

Over the past decade, deep learning is proving to be an effective tool in addressing challenges related to image classification and object detection. In this context, enormous improvements in deep neural networks have been made, with well-described and pretrained models available for use by the research and development community. Several of the pretrained deep networks are trained against numerous datasets, including COCO [1], ImageNet [2], and the Stanford Car dataset [3] which contain up to a million labeled images in more than a hundred different classes.

The performance of deep learning networks is inherently dependent on the suitability, quality, and quantity of the datasets on which the networks are trained. What to learn is more important than how to learn. Regardless of how deep networks are structured, insufficient or incorrectly labeled datasets result in lower-quality models. One of the issues frequently encountered when using machine learning methods for analyzing and inferring image or video data of vehicles, pedestrians, and other public activities is the lack of properly labeled training datasets. Because of logistics, privacy, and practicality reasons, vision systems in urban public areas usually are deployed only above human eye level, at the height of 18-20 ft [4]. These systems are deployed predominantly on existing street infrastructure such as street light and traffic signal poles, where there is already significant competition for mounting real estate. The view from this elevation is considerably different from the view from human eye level; and for automobile-related machine-learning problems and applications (e.g., car make and model classification), having a well-labeled dataset with the top-down viewing angle is important. Unfortunately, to date, the two most popular car datasets, Stanford [3] and CompCars [5], are considerably lacking in images from nontraditional viewpoints (e.g., views from the top).

The lack of sufficiently-curated and accurately labeled datasets makes it challenging to train models for various public-way use cases such as traffic flow analysis, vehicle type, and make and model identification. Yang et al. [5] highlight the importance of having datasets comprising images from multiple camera perspectives for automobile-related machine-learning problems and applications. To solve the perspective gap in the publicly available datasets, in this paper we describe a method and a set of tools to curate datasets that contain images of automobiles from multiple camera perspectives. The approach here takes inspiration from the "teacher-student" model introduced by Kukreja et al. [6]. The teacher in the model teaches one or more students training on different viewpoints based on what the teacher recognizes from its viewpoint. The technique presented in this paper uses a supervised learning approach, with the aim of generating more training data with reduced human involvement and sufficient autonomy.

We propose an offline dataset creation method, using images extracted from multiple previously recorded video streams. These video streams capture the vehicles within a predefined region on a roadway from multiple perspectives,

one of which a learning model has been trained to identify. Machine learning models trained to detect automobiles and their features from the known perspective are next used to generate labels for image frames from various viewpoints. In the context of automobiles on roadways, the best-known perspectives are the front and side views captured from the human eye level as seen from the images in the multiple publicly available datasets referenced earlier. There are several challenges in capturing these video datasets, splitting the videos into frames, extracting and pairing the time synchronized frames from the various perspectives, and picking the best label for the images. The main focus of this paper is hence to propose a streamlined process and provide tools that can help generate the training datasets. The process is divided into three phases: multiperspective image acquisition, frame synchronization, and label assignment. Figure 1 illustrates the first two phases in general.

The rest of the paper is organized as follows. Section II describes existing automated labeling methods and how they complement and motivated the proposed approach. Section III illustrates the image acquisition process that crops images with objects within the cropped images. Section IV describes how the cropped images are aligned, and Section V describes the labeling process. Section VI details the result of the process with examples, and Section VII discusses challenges of the process. Section VIII concludes the work and describes future directions.

II. Related Work

Image annotation in machine learning denotes assigning labels or tags to images. This metadata is an important part of the image-dataset and is fundamental to the training process. To start, the unlabeled images are acquired from a variety of sources. Because of their nature, some of them often come with *weak* labels – merely a title of the video or a few words describing the image. Many research studies have used this information to label images and videos [7]–[9]. The approach of finding the most relevant word only applies to images that come with some metadata, and is not useful when the images are captured directly from cameras. For word-matching, latent space analysis, probabilistic latent space analysis, or finding nearest neighbors based on rank or distance are possible approaches for image auto-annotation [10]–[14]. Such approaches separate textual context from the weakly labeled image and process the textual context to retrieve the words most relevant to the image. They rely on natural language processing to identify the most applicable terms for labeling from the semantic structure of the text surrounding the images and videos. In essence, these approaches assume that the weakly labeled images are visually in a similar context, which is very different from our approach of capturing a region of interest from video clips and annotating images without being given relevant words. In image segmentation, the concept of image annotation is applied such that each segment on an image is matched with one or more word representations [15], [16].

Recent studies have adopted deep learning methods to compare similarity between labeled images and unlabeled images [8], [17], [18]. The approaches explore feature space similarity of labeled and weakly labeled images, inspect the relevance of the descriptor of the images, and select the most relevant label from a list of descriptors. In constructing an image synset from weakly labeled images, the feature extraction component of deep learning is actively exploited. This process can reduce the semantic gap of weakly labeled images [19]. In addition, deep learning techniques explore correspondences between labels and images, and use weakly labeled images for auto-annotation. Still the methods are needed to utilizing the mixture of weakly labeled and unlabeled images to collect possibly relevant words and use deep learning method to find similarities between the images.

Luo et al. [20] introduced another technique that does not rely on weak labels that come with the source. Instead, they create labels using the label from a source that is known to work well. In their approach, a pretrained deep network model capable of detecting vehicles from a high-resolution camera image is used to create labels. The camera is physically aligned with a mobile light detection and ranging (LiDAR), and the 3D-point views of images generated by LiDAR are then annotated from the labels obtained from the pretrained model. This approach is significantly more flexible than using weak labels because weak labels provide only static information, and the dynamics of the scene may change. Moreover, a better model can be used to improve the performance of the image annotation. This approach is similar to our approach in terms of creating labels of images captured from a camera using a pretrained deep learning model and annotating the images collected from different perspectives or sensors. In our approach, we are collecting images of a target from multiple perspective cameras, identifying the target from one perspective, and labeling other images from different perspectives.

III. Image Acquisition

Similar to the technique described by Luo et al. [20], in this paper we propose a data-processing and labeling pipeline for the annotation of images obtained from videos shot from multiple perspectives, where a minimum of one perspective is previously known and identifiable by our models. In the particular automobile labeling context described here, the perspectives are aligned such that they capture the moving vehicles in a common physical space or zone, but from multiple angles. The known and trained perspective is next used to generate a label that is applied to the other images. To our best knowledge, our proposed approach is the first to auto-annotate unlabeled images from multiperspective raw videos.

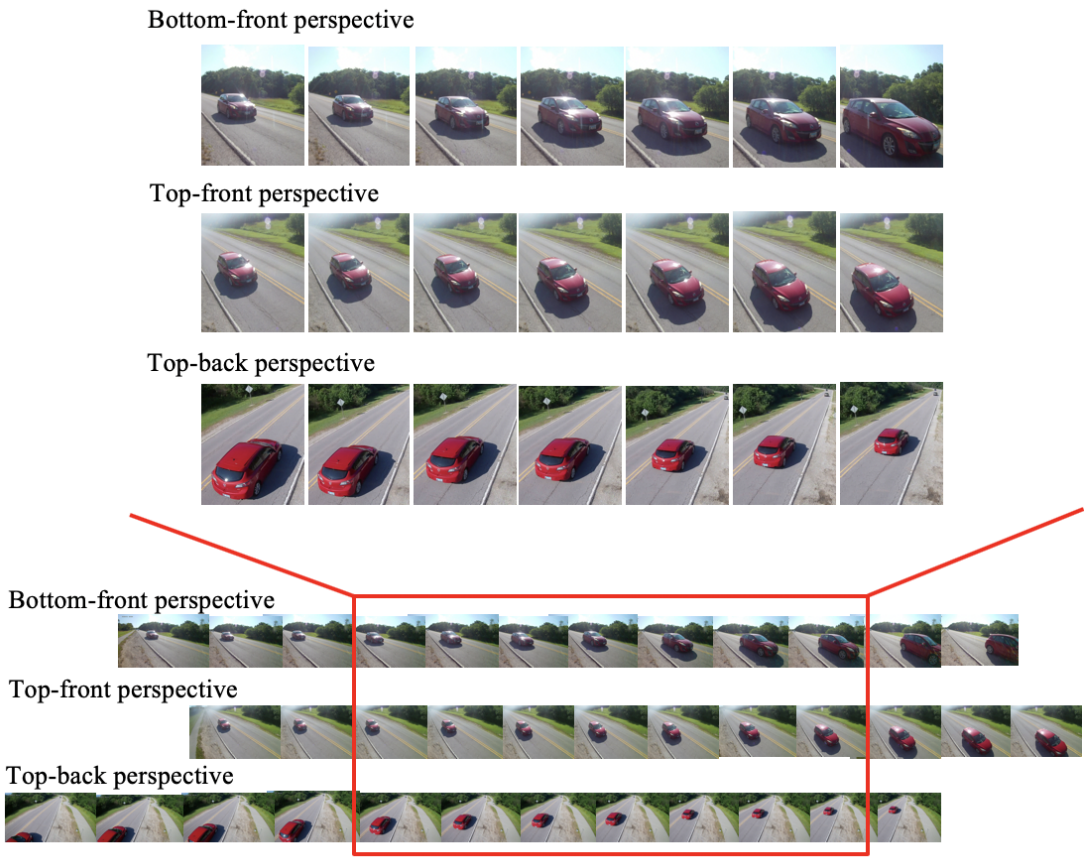


Figure 1: Illustration of generating a dataset from multiperspective videos.

To obtain the images to build our training dataset, we utilized three cameras to capture videos from three different perspectives: *top-front*, *top-back*, and *bottom-front*, as shown in Figure 2. The bottom-front view is the known view. Each camera is angled toward the roadway such that any vehicle passing by will be seen by all the cameras.

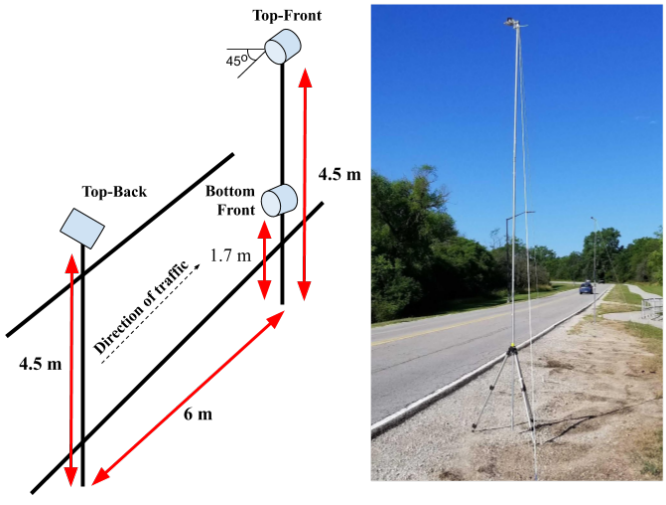


Figure 2: Illustration of the camera setup with the actual installation shown on the right. Each camera represents each perspective.

A video camera capturing a vehicle approaching or moving away from it will produce videos where the size, shape, and view of the vehicle changes as the vehicle moves. Consequently, the recorded features of the vehicle, including logo of the manufacturer, model name, color, and other physical attributes, will vary over the duration that the vehicle is in view of the camera. In our technique, in addition to obtaining the colocated images from multiple perspectives, the image acquisition process extracts the frames that capture the attributes of the vehicle in sufficient detail.

A. Data Collection

To capture the images discussed in this paper, the camera setup was installed on a single-lane roadway with a speed limit of 35 miles per hour. The cameras mounted on each of the poles recorded 4K resolution videos at 24 frames per second (FPS) on a sunny day. The slight glare in a couple of the perspectives was due to the sun throughout the

one-hour recording process. We do not foresee an inherent limit in the number of cameras and hence perspectives that are captured. However, there are some constraints and challenges in extending this setup more broadly. In this setup, we expect the vehicles to drive through the zone at a relatively constant speed, without abrupt acceleration or deceleration; vehicles abruptly stopping will lead to temporary loss of synchronization. These challenges are discussed further in Section VI.

B. Region of Interest

In addition to the vehicles we are interested in, the videos captured by the cameras often included distracting objects in their view. These include objects such as traffic signs, lane markings, trees, poles, manhole covers, and vehicles in other lanes, which are ill-suited for our datasets because they are likely to be blurry, distorted, and intermittently occluded. Hence, within the view of every camera, a region has to be chosen that captures the vehicle in the "common-view" zone and excludes as many undesirable objects and artifacts as possible. A universal region does not exist; however, the following points should be considered when locating the region of interest (ROI) for each camera perspective,

- **Camera's view of the "common-view" zone:** The cameras should be mounted and angled such that vehicles moving through the "common-view" zone are in view for at least a dozen video frames. The zone should also be within sufficient distance from the camera, that images have good spatial resolution, allowing the physical features of the vehicle to be captured in distinguishable detail. This will enable devising a ROI, that is tightly bounded and hence largely free of static features (road signs, lane markings, cracks and crevices, etc.).
- **Vehicle dynamics:** The ROI should be restricted to stay in the boundaries of one lane of traffic. This restriction is to prevent any detection of vehicles appearing in the other lane or in the opposite traffic.
- **Minimum size:** In order to determine the size of the ROI, the size, shape, and average speed of the target vehicles, along with the camera's frame rate, image resolution, and view, needs to be considered. Ideally, the ROI should be large enough that the vehicles are captured in the best possible view from the camera's perspective for at least a dozen video frames.

In the videos we collected at 24 FPS, each vehicle was visible in the camera's view for an average of 3 seconds. A vehicle is deemed *clearly visible* only if features such as vehicle shape, make logo, and model name are clearly distinguishable. Only frames containing clearly visible vehicles are useful for the purpose of the datasets. On tightening the requirements for useful frames, a majority of the vehicles were captured in useful detail for about 1 second, amounting to 24 frames. For each pass of the vehicle, we used 7 frames from each of the camera viewpoints to help label and build the dataset, leaving sufficient room to adjust the ROI and tightly control what static objects were captured in the images. (Refer to subsection E for the extended rationale behind this decision.) The ROIs can be of any shape, but for this work a rectangular region was used. Also, the ROI for the different perspectives varied in size and location within the frame of view from the camera.

A selected ROI might include regions where nontarget objects appear. To eliminate these false triggers and focus only on the vehicles, we set another zoomed area within the ROI. This zoomed area acts as a tripwire so that only frames where vehicles drive through this area will be considered as frames of interest. This secondary zoomed region is smaller than the full ROI. The full ROI is still the region that is used to extract the vehicle from the frame; the detection region is used only to improve accuracy by reducing duplicate detection.

C. Base Image

Once the ROI, which is a small static box within the view of a camera, is selected based on visual human inspection of the videos, the 7 frames of the vehicle are extracted. This task can be accomplished in several ways. We use a computationally less intensive "absolute difference" method (Subsection III-D) to identify the useful frames for the vehicles. In our case, three base images are created, one for each of the different perspectives, and for each of the three base images, three zoomed regions are also created. The created base images will be compared with the frames of the video and used to identify the frames with activities.

The base image captures only the static background (e.g., roads, markings, and other static objects) and does not contain any vehicles or shadows in the scene. This is achieved by saving a frame from the video where there is no vehicle activity. In the image acquisition process, we include a mechanism that updates the base image as the environmental conditions change, and also opportunistically when there is no activity between vehicle encounters. This approach reduces the occurrence of false positives in the "absolute difference" step due to environmental changes such as passing of dense clouds, movement of the sun and shadows, and precipitation, which would otherwise be identified as vehicle activity.

D. Absolute Difference

In this step, the pixel difference between a frame in the video and the base image is calculated. This pixel difference provides a metric for the difference between the two images, and an estimate of whether there is an object within the ROI in a certain frame in the video stream. Figure 3 shows the image subtraction step. As with the base image, the frame being compared is blurred before subtraction; and as shown in Figure 3c, the difference appears as a

(a) Base image (b) Car image (c) Absolute difference (d) Thresholded difference

Figure 3: Image subtraction between base image and the frame from a video.

numerical value on the color channels of the image. Since the base image consists solely of the empty road with constant environmental conditions, a significant change to the image would mean that an object is within the frame. The object would not necessarily be the target object, a vehicle, but that is highly likely because the zoomed region is confined within the road. First, the captured base images are blurred to minimize noise and high-frequency artifacts. Figure 3d shows a single-channel image of the difference indicating any pixel when summation of the pixel values on the channels from Figure 3c is greater than a threshold. This threshold is called the pixel difference threshold. The pixel difference threshold is calculated simply by finding the best appearance of the object from the image while adjusting the threshold value.

E. Finding and Collecting Peak Frames

After computing a binary difference image, we count the number of white pixels on the zoomed region of the binary image (see Figure 3d). The white pixels in the binary image indicate a significant difference between the base image and the video frame. We then divide the number of the pixels by the total number of pixels on the corresponding zoomed region in the base image. This quotient indicates the percent difference between the base image and the frame in the video. A threshold is applied to the quotient to remove frames that are different because of intrinsic camera noise, small objects of non-interest and other artifacts, leaving only frames with significant changes. Figure 5 shows an example of finding peak frames.

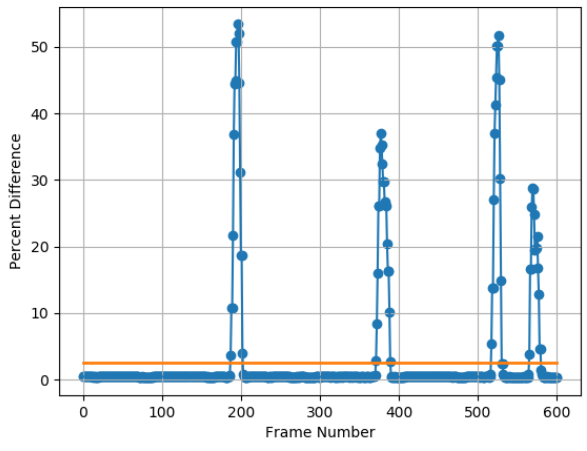


Figure 4: Percent difference of the frames. There are four vehicle encounters. The peak frames indicate an object in the region of interest.

The "absolute difference" step allows us to easily identify frames of interest. However, not all frames with the vehicles are needed, since in some frames which are flagged as containing a vehicle, only parts of the vehicle are visible, like the very front or back. These frames should not be added to the dataset, so we decided to collect and save only the 7 best frames of each vehicle encounter. The reason we chose to select 7 frames for each encounter was that it fit our specific setup. The camera used in our experiment captured footage at 24 frames per second. Since a car was only visible in frame for about a second, this meant that there were 24 images of the car entering and leaving the frame. Of the 24 images, 7 images were of the car within the region of interest. This is why 7 frames were chosen for our experiment. The number of vehicle frames to select depends on the field of view of the camera, the frame rate of the camera, and the speed of the vehicles along the road. A camera with a wider field-of-view is able to record more satisfactory frames of a passing vehicle than a camera with a smaller field-of-view. This is because a smaller field-of-view camera is more limited than a wider field-of-view camera, since vehicles will spend less time in its field-of-view than in the wider angle camera. The same idea applies to frame rate. A camera with a higher frame rate will produce more usable frames per vehicle encounter than a lower frame rate camera, because the higher frame rate camera will be able to save more frames which contain the full view of the vehicle while it is within the camera's field-of-view. Ultimately, the user must decide the optimal number of frames to select, because it depends on the specific setup used.

Figure 4 shows the percent difference between the base image and the frames from a video clip of four vehicles. The

(a) Peak - 5 (b) Peak - 4 (c) Peak - 3 (d) Peak - 2 (e) Peak - 1 (f) Peak frame (g) Peak + 1

Figure 5: Example of a car encounter. The percent differences allow finding the peak and the neighboring 6 frames.

horizontal line at around 2.5 percent difference is the threshold to distinguish the vehicles from the background. From the figure, the four encounters can clearly be identified. Each vehicle encounter begins with a percent difference jump over the threshold. Because the vehicles passed through the scene with different head angles and speeds, each encounter has a different peak and duration. Nevertheless, the highest peak frames on each vehicle encounter should contain the best view of the vehicle from that perspective. The highest point of the difference in Figure 4 represents the peak frame. The rest of the six frames are selected based on the direction of the vehicle movement. If the vehicle is leaving the camera, the 5 frames before the peak frame and one frame after the peak frame are selected. On the other hand, if the vehicle is approaching the camera, the 5 frames after and one before the peak frame are selected. Selecting frames in this manner improved the number of satisfactory frames that we were able to extract. The selected 6 frames along with the peak frame are saved to make a list of 7 frames per perspective for every vehicle encounter.

IV. Frame Synchronization

The goal of this effort is to create a set of images for every encounter across all the different perspectives, 21 in our case, following which the set can be labeled by using the most-confident label from the known perspective. The 21 images can then be added to the larger vehicle database. Synchronizing the encounters across the 3 perspectives is not straightforward since the videos are not time-synchronized with a universal clock. To automate the process of synchronizing the videos and accumulating the images from various perspectives together, we propose a technique that uses the time difference of arrival between two successive encounters. In the previous step, for each perspective the 7 consecutive best frames were identified for every vehicle encounter. This process reduces the video into sets of 7 frames, spaced in time. Essentially, the time between the encounters in each video represents no vehicle movement. One can calculate the time difference between arrivals for each perspective by merely calculating the time between when the peak frames are encountered for each vehicle encounter. At a high level, given that all the cameras are capturing the same traffic flow, although from different perspectives, the time difference (which can be mapped to frame difference) between encounters should be similar. By aligning the time difference of arrival across the perspectives, a set of temporally colocated images can be identified.

Another issue commonly faced is the lack of a "common-view" zone. In fact, in our experiments we have seen that the need for such a zone can be relaxed by using more sophisticated alignment techniques provided that the vehicles do not drastically change their speeds between when they are encountered by the different cameras. As shown in Figure 6, we recorded videos from 3 perspectives: bottom-front, top-front, and top-back. In one of our camera capture settings, the bottom-front and top-front perspectives were able see vehicles at a similar time, while the top-back perspective camera could see the vehicles only when they receded from the other two perspective cameras. Even in a perfect world where the videos are time stamped, they cannot be aligned in time to extract the 21 frames that make an encounter set[1].

To overcome this desynchronization between the videos, the proposed algorithm utilizes the number of frames between peak frames. For example, if a car appears in the bottom-front view in nth frame and the next car appears in the same view in $(n+324)$th frame, then the other perspective cameras should have the similar frame difference between the first appearance of the two vehicles, as shown in Figure 6. The slight time delay between the perspectives does not affect the ability of this algorithm to find the same vehicle from multiperspectives since time distance measures the relative time between car encounters, not their relative positions to the cameras. Assuming constant speed, the time elapsed between vehicles encounters should be similar for all cameras, and this fact allows us to align the videos. Later, we discuss problematic situations where this assumption is not met.

[1]The synchronization algorithm utilizes two parameters; "time distance" and "alignment error". To best describe the algorithm, time distance is a measurement of the time elapsed between two cars passing by a virtual point along a road in a video clip. Time-distance is measured in the unit of frames because that is the most fundamental unit of time in video footage.

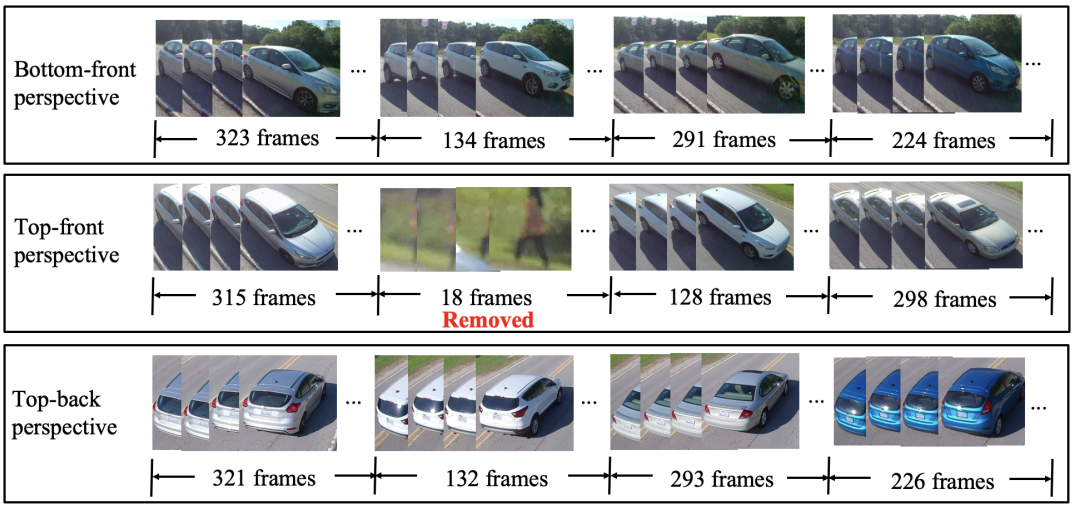


Figure 6: Illustration of synchronization; performed according to the number of frames between the first scene in which each vehicle appears in the frame (time distances). As shown in the figure, the top-front perspective becomes synchronized because of mistakenly identifying a person in the background as a car. After that point, unless resynchronization is applied, this perspective will be out of sync from the other two.

Alignment error is the absolute difference between the lists of time distances of the videos. Our proposed method calculates the alignment error of every possible relative shift between the lists of time distances. The alignment error list is a sequence of the differences between the two lists of time distances. For example, the leftmost time distances of the 3 perspectives illustrated in Figure 6 give an alignment error of 8 between bottom-front and top-front, an error of 6 between top-front and top-back, and an error of 2 between top-back and bottom front views, while the next time-distances of bottom-front and top-front views give alignment error of more than 100. The lower the values in the error list, the more likely the time distances are correctly aligned with the given shift. The algorithm chooses an alignment where the error is continuously below a threshold. In this study, the threshold was set to 10 frames, that is, approximately 0.42 seconds in a 24 FPS video. If the algorithm encounters an alignment error greater than the synchronization error threshold, it will stop matching encounters contiguously and attempt to resynchonize matches between videos.

Ideally, the synchronization needs to be performed once for a set of videos if the time distance lists come with only correctly recorded vehicles. However, the time distances can be miscalculated when one car is recognized twice or a moving object that is not a vehicle is identified as a target such as shown in second time distance of the top-front view on Figure 6. If this misidentification happens to one perspective but not the others, this one additional detection will cause a cascading error. In other words, the synchronization will be off by one car encounter for every match after the error. This type of error is much like a frameshift mutation in DNA.

The issue of recognizing one vehicle twice is solvable by implementing an algorithm to let a detected vehicle pass before continuing to search for new vehicles. This will reduce the chances of a vehicle being detected twice. However, the other issue of identifying non-vehicles cannot be easily avoided, for example, if an animal or bicyclist crosses into view of the camera. In order to solve this problem, the proposed algorithm is to constantly resynchronize the videos. The algorithm finds out when to resynchronize by looking at the alignment error lists between videos. The algorithm stops matching encounters when it finds an error greater than the alignment threshold while it is scanning the alignment errors.

Resynchronization follows the principle that sections of time-distance lists can be offset by a certain amount to account for frameshift errors. The algorithm intelligently skips over miscalculated encounters so that it can get back on track and revert to the correct alignment. The offsets are necessary because every time-distance number downstream from a frameshift error will be misaligned without them.

Resynchronization becomes more complicated when we want to synchronize more than 2 perspectives. In such cases, we need to use the transitive property to match intervals of encounters. For example, if we know that an interval of perspective A maps to an interval of perspective B and that an interval of perspective B maps to an interval of perspective C, we can be certain that the interval of perspective A maps to the interval of perspective C. This property allows the synchronization algorithm to tie together as many encounter sequences as possible between different videos. When working with 3 or more videos, the synchronization

algorithm first tries to sync up every time-distance list with every other time-distance list. It then applies the transitive property to relate together synchronized sub-sections. The transitive property allows the algorithm to match up 3 or more views of the same vehicle.

V. Labeling

The last step of the process involves labeling the frames of vehicles detected and aligned by the prior steps. As claimed in Section I, existing popular deep networks are trained mostly against vehicles viewed from human-eye level, the "bottom-front" view in our setup. The labeling process takes the deep network model to classify the "bottom-front" image of a vehicle and annotates the two perspectives, "top-front" and "top-back", with the same label. In our application, the 21 images representing a vehicle encounter are broken down into 7 sets of 3 images each, one from each perspective. As a result of this step, each set of 3 images is provided a label of the make and model of the vehicle based on the label of the "bottom-front" image inferred by the trained model. The efficacy of this algorithm depends on the accuracy of the trained model and can be improved by relabeling the image sets as the machine-learning models improve.

VI. Performance Results

The techniques and the tools presented in this paper were used to build a dataset using three cameras to capture images from a known viewpoint and two other viewpoints. Table I shows performance of the proposed automotive labeling technique. As the cameras started and stopped recording at different times, they captured a different number of vehicle encounters. In a 70-minute recording on one day, the technique detected about 500-odd encounters in each of the three camera streams. Of these encounters, the technique was able to align and extract 344 unique encounters with 21 images comprising 7 sets of 3 temporally colocated images (one for each perspective).

We have identified a few reasons why the synchronization algorithm skipped over a several encounters, as shown in Table I. First, the time-distance lists it has to work with are not entirely error-free. Our percent difference algorithm sometimes mistakenly counts vehicles twice or is triggered when the camera shakes. Whenever these problems happen, the time-distance list given to the synchronization algorithm takes on error. The synchronization algorithm has the ability to handle these types of errors, but in the process of correcting these errors; it is forced to skip over a few encounters to resynchronize the videos.

Table I: Quantitative result of the labeling process.

Perspective	Total encounters	Encounters successfully aligned with both the other views	Encounters successfully aligned with only one other view
Bottom-front	586	344 (58.7 %)	502 (85.7 %)
Top-front	567	344 (60.7 %)	466 (82.2 %)
Top-back	536	344 (64.2 %)	380 (70.9 %)

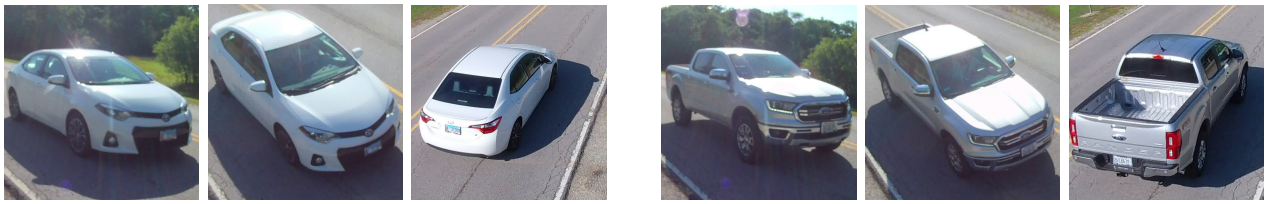
Second, constants used in the synchronization algorithm cause it to be cautious of false positives. We take a conservative approach here and have the algorithm skip over a few encounters to regain synchronization, in order to prevent matching up different vehicles across the various perspectives. The reduction in true positives is a side-effect of our effort to reduce false positives and mismatch.

A pretrained vehicle make and model classification model was selected to label the multiperspective vehicle images we obtained from the process. The "bottom-front" image of each encounter was fed into the model, and the two other perspective images were labeled with the label the model outputs. As a result, of a total of 12,306 images, 7,224 images from three perspectives were labeled. Figure 7 shows some of the labeled images.

VII. Discussion

The environmental conditions encountered during our video imagery collection process were optimal. However, less favorable situations will increase the probability of errors in alignment and selection of peak frames. The following aspects need to be considered when applying the process.

- Selection of peak frames should consider the dynamics of the perspective. We take a constant approach here, aggregating 7 frames for each viewpoint for every vehicle encounter. In certain situations, however, it may be better to take an adaptive approach. For example, when looking at frames from the top-back perspective, it makes more sense to invert the order and take one frame before the peak, then the peak, and 5 after the peak. One general approach to selecting the neighbors of peak frames is to select the neighboring frames with the top six percent differences. This will ensure that the frames that show the greatest region of the car will be selected, because the visibility of the vehicle is tied to the percent difference to the base image. As the vehicle becomes more visible, the percent difference increases.
- Traffic on the road being recorded must be nonuniform. If the traffic is uniform, the time distance of each car encounter will be similar. In this situation, the proposed algorithm may fail to synchronize perspectives because it is no longer able to distinguish the time distances.
- If the cameras are separated by a distance along the road, that is, no "common-view" zone exists, the roadway section between the cameras must be nonbranching and essentially behave like a constant speed FIFO queue of vehicles. The vehicles should not change their order when passing through the views and should not change their speed during the whole series of encounters.



(a) A Toyota car on the perspectives

(b) A Ford truck on the perspectives

Figure 7: Result of the labeling process. Using an "bottom-front" perspective, the other perspectives are labeled.

These assumptions must be handled by the user's specific setup. The most limiting constraint is the behavior of the target road, but there are two specific methods that can be utilized to work around the road constraints. The first adaptive method is lane cropping. Although the proposed labeling method cannot synchronize encounters across multiple lanes of traffic, it is possible to narrow the camera's field-of-view to only consider one lane. This was done in our setup. We specified a region of interest that bounded only one lane, allowing the algorithm to focus only on cars passing one direction. This may not be possible for all camera angles (i.e. a ground-level horizontal view), but it is one possible adaptation that could enable the proposed labeling method to work on a multi-lane highway. The second adaptive method is to setup cameras looking down on the "funnel points" of a road. For example, highway ramps satisfy all the road conditions of the proposed labeling method. Highway ramps are one-lane roads which contain low, non-uniform traffic and vehicles which should not significantly change their respective spacing. Although the proposed algorithm could not synchronize all the traffic of a interstate highway, it could synchronize the traffic *leading into and out of* that highway. Additional work will be required for devising an algorithm that is better suited to synchronizing vehicles regardless of traffic flow.

VIII. Conclusion and Future Work

The proposed labeling process expands the applicability of the "teacher-student" model. It not only allows teacher models to transfer their knowledge onto student models, but it also allows student models to be able to correct themselves. Because the synchronization puts different perspectives of vehicles together, multiple student models each trained on a particular perspective of the vehicle can try to determine the correct label for the vehicle. Democratic approaches such as a consensus model [21] or voting algorithm [22] can facilitate such activity.

One other improvement is in detecting a vehicle within the region of interest by using deep network models. The process currently behaves as a motion sensor that senses any movement in the scene in order to activate the peak-finding algorithm. Any deep network models trained against COCO [1] or another dataset containing cars should detect a car from the region of interest so that false triggers from nontarget objects such as bicycle or person can be ignored. False triggers contribute a significant amount of error to the algorithm, and although it can work around those errors, it has to re-synchronize the videos it is trying to match, which involves skipping over some encounters. The yield of total encounters matched should generally increase as false triggers decrease, which makes better car detection an important future work.

In this paper, we proposed a labeling process that aggregates temporally colocated frames of a target object from multiple video streams. With the help of a pretrained deep network for labeling images, the process bundles the images with the label to help create a new image dataset. For example, in our application, deep networks that are trained against the new dataset will be able to recognize vehicles viewed from different perspectives. The current process described in the paper has several limitations and makes important assumptions, which have been described in detail to help replicate this work on other camera setups.

Acknowledgments

We thank Sean Shahkarami for his discussions and input on the manuscript. This material is based upon work supported in part by U.S. Department of Energy, Office of Science, under contract DE-AC02-06CHI1357, and in part by the Exelon CRADA T03-PH01-PT1397.

References

[1] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *Computer Vision – ECCV 2014*, D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, Eds. Cham: Springer International Publishing, 2014, pp. 740–755.

[2] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A Large-scale Hierarchical Image Database," in *2009 IEEE Conference on Computer Vision and Pattern Recognition*. Ieee, 2009, pp. 248–255.

[3] J. Krause, M. Stark, J. Deng, and L. Fei-Fei, "3D Object Representations for Fine-Grained Categorization," in *The IEEE International Conference on Computer Vision (ICCV) Workshops*, June 2013.

[4] C. E. Catlett, P. H. Beckman, R. Sankaran, and K. K. Galvin, "Array of Things: A Scientific Research Instrument in the Public Way: Platform Design and Early Lessons Learned," in *Proceedings of the 2nd International Workshop on Science of Smart City Operations and Platforms Engineering*, ser. SCOPE 17. New York, NY, USA: Association for Computing Machinery, 2017, p. 2633. [Online]. Available: https://doi.org/10.1145/3063386.3063771

[5] L. Yang, P. Luo, C. Change Loy, and X. Tang, "A Large-Scale Car Dataset for Fine-Grained Categorization and Verification," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

[6] N. Kukreja, A. Shilova, O. Beaumont, J. Huckelheim, N. Ferrier, P. Hovland, and G. Gorman, "Training on the Edge: The Why and the How," in *2019 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW)*, May 2019, pp. 899–903.

[7] H. Aradhye, G. Toderici, and J. Yagnik, "Video2text: Learning to Annotate Video Content," in *2009 IEEE International Conference on Data Mining Workshops*. IEEE, 2009, pp. 144–151.

[8] J. Wu, Y. Yu, C. Huang, and K. Yu, "Deep Multiple Instance Learning for Image Classification and Auto-annotation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 3460–3469.

[9] J. Li and J. Z. Wang, "Real-time Computerized Annotation of Pictures," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 6, pp. 985–1002, 2008.

[10] F. Monay and D. Gatica-Perez, "On Image Auto-annotation with Latent Space Models," in *Proceedings of the Eleventh ACM International Conference on Multimedia*. ACM, 2003, pp. 275–278.

[11] S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman, "Indexing by Latent Semantic Analysis," *Journal of the American society for Information Science*, vol. 41, no. 6, pp. 391–407, 1990.

[12] T. Hofmann, "Unsupervised Learning by Probabilistic Latent Semantic Analysis," *Machine learning*, vol. 42, no. 1-2, pp. 177–196, 2001.

[13] M. Guillaumin, T. Mensink, J. Verbeek, and C. Schmid, "Tagprop: Discriminative Metric Learning in Nearest Neighbor Models for Image Auto-annotation," in *2009 IEEE 12th International Conference on Computer Vision*. IEEE, 2009, pp. 309–316.

[14] J. Weston, S. Bengio, and N. Usunier, "Large Scale Image Annotation: Learning to Rank with Joint Word-image Embeddings," *Machine Learning*, vol. 81, no. 1, pp. 21–35, 2010.

[15] K. Barnard, P. Duygulu, D. Forsyth, N. d. Freitas, D. M. Blei, and M. I. Jordan, "Matching Words and Pictures," *Journal of Machine Learning Research*, vol. 3, no. Feb, pp. 1107–1135, 2003.

[16] M. Guillaumin, D. Küttel, and V. Ferrari, "Imagenet Auto-annotation with Segmentation Propagation," *International Journal of Computer Vision*, vol. 110, no. 3, pp. 328–348, 2014.

[17] Y. Ma, Y. Liu, Q. Xie, and L. Li, "CNN-feature Based Automatic Image Annotation Method," *Multimedia Tools and Applications*, vol. 78, no. 3, pp. 3767–3780, 2019.

[18] R. Kiros and C. Szepesvári, "Deep Representations and Codes for Image Auto-annotation," in *Advances in Neural Information Processing Systems*, 2012, pp. 908–916.

[19] R. Zhao and W. I. Grosky, "Narrowing the Semantic Gap-improved Text-based Web Document Retrieval Using Visual Features," *IEEE Transactions on Multimedia*, vol. 4, no. 2, pp. 189–200, 2002.

[20] H. Luo, C. Wang, and J. Li, "Auto-annotation of 3D Objects via ImageNet," in *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

[21] I. J. Pérez, F. J. Cabrerizo, S. Alonso, and E. Herrera-Viedma, "A New Consensus Model for Group Decision Making Problems With Non-Homogeneous Experts," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 44, no. 4, pp. 494–498, April 2014.

[22] P. K. Chan and S. J. Stolfo, "A Comparative Evaluation of Voting and Meta-learning on Partitioned Data," in *Machine Learning Proceedings 1995*. Elsevier, 1995, pp. 90–98.