Classifying Ego-Vehicle Road Maneuvers from Dashcam Video

Stephen A. Zekany, Ronald G. Dreslinski, and Thomas F. Wenisch
University of Michigan

***Abstract*— A key challenge for self-driving vehicle researchers is to curate massive instrumented vehicle data sets. A common task in their development workflow is to extract video segments that meet particular criteria, such as a particular road scenario or vehicle maneuver. We present a novel approach for detecting vehicle maneuvers from monocular dash-cam video building upon a deep learning visual odometry model (DeepV2D) to estimate frame-accurate ego-vehicle movement. We classify movement sequences against reference maneuvers using dynamic time warping and simple heuristics. We show that using deep learning visual odometry to estimate location is superior to consumer-grade high-resolution GPS for this application. We describe and implement a greedy approach to classify maneuvers and evaluate our approach on non-trivial road maneuvers, finding an overall AUROC value of 0.84.**

I. Introduction

State-of-the art systems for autonomous driving are built using massive data sets for training and evaluating vision and control algorithms [1]. A key challenge for self-driving vehicle researchers is to manage and curate these massive data sets. Video libraries associated with autonomous vehicles rapidly grow to enormous sizes; for example, the publicly available Berkeley DeepDrive data set [30] comprises 100,000 video segments and over 1,100 hours of driving, while proprietary data sets can grow much larger [26].

A common task in the development workflow of autonomous systems is searching such a video archive to extract segments that meet particular criteria. Such searches might be used to find test scenarios to evaluate algorithm performance under unusual circumstances, for example, a segment where strong braking is needed to avoid a collision. Alternatively, when building a training data set for deep learning algorithms, it may be desirable to oversample video segments for situations that arise rarely in practical driving. Existing data management systems are well suited to execute fast searches and queries over relational (tabular) data, but typically cannot do so for unstructured data like video. State-of-the-art video processing systems, like Scanner [18], include optimizations like frame skipping and efficient delta-frame decoding, but still largely resort to brute-force search over frames. Developments in deep learning, such as object detection and semantic segmentation, present new opportunities for video search systems to leverage, but do not yet attain human-level accuracy [5] [7] [12].

In this paper, we develop a processing pipeline that facilitates generating an index to search for vehicle maneuvers from dashboard camera video. The system classifies video frame sequences against a set of reference video clips, provided by the user, that demonstrate the vehicle maneuvers (e.g., right turn, U-turn, driving in reverse) to be indexed. Our system does not rely on GPS, accelerometry, or other metadata to determine the motion of the ego-vehicle. Instead, we utilize an existing deep learning model, DeepV2D [22], to find translational and rotational camera motion between successive frames. We process these pose estimates into a vehicle trajectory, and compare trajectory segments to the reference maneuvers. The system proceeds in two high-level phases: In the first, computationally more expensive phase, we apply the DeepV2D model to reconstruct high-resolution ego-vehicle trajectory from monocular front-facing dashcam videos. In the second phase, we compare the extracted trajectories to the reference maneuvers and label matching video clips. In this way, the reconstructed trajectory data acts as an index for the video library, enabling efficient search for vehicle maneuvers. We evaluate our approach against human-labeled ground truth for seven common road maneuvers, and compare against a classification approach that uses measured GPS rather than reconstructed trajectories.

We make the following specific contributions:

- We describe a technique for searching for vehicle maneuvers using only dashcam video.
- We implement and evaluate our search technique.
- We demonstrate that for this search technique, ego-vehicle trajectory estimated via deep learning is superior to GPS.

II. Prior Work

In this section, we review two similar areas of work: aggressive driving detection and SLAM (simultaneous localization and mapping) from video.

A. Maneuver Classification

A number of studies have explored methods to evaluate *driving style* by analyzing vehicle characteristics and maneuvers. Driving style is typically a characterization of how "aggressive" a specific driver is relative to an aggregate set of reference drivers. This work is of particular interest to safety agencies and insurance companies, who wish to mitigate risk via measurement of individual or aggregate behavior. While our work does not focus on driving style or safety specifically, it is relevant because we utilize similar techniques for measuring vehicle maneuvers. Johnson et al. [9] describe a method to measure aggressiveness for turns and straight-line motion using dynamic time warping with a smartphone. Later work extended this approach with Bayesian classification [3], support vector machines, and random forest analysis [11]. Other work has focused on building

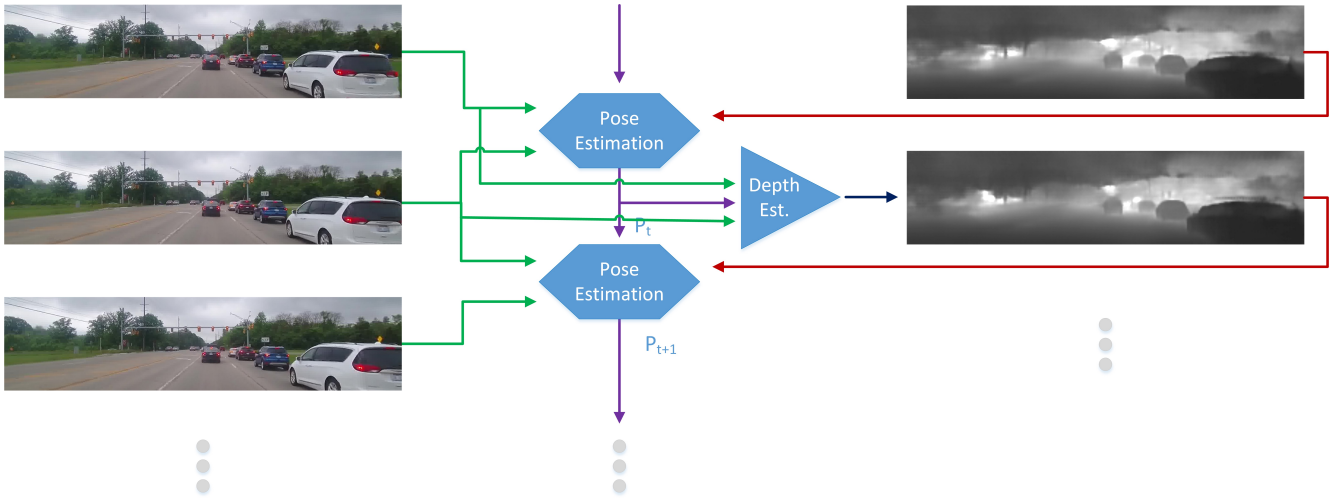

Fig. 1. DeepV2D's pose estimation creates a depth map for each frame, and this depth map then provides the basis for the next frame's pose estimate.

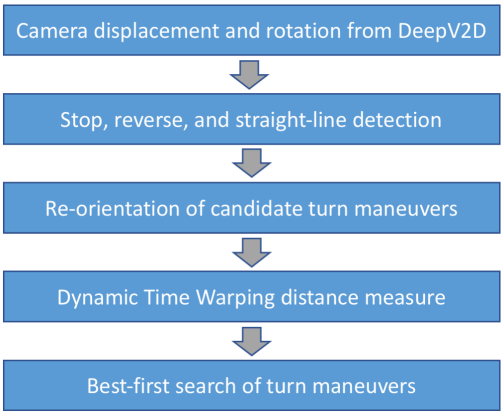

Fig. 2. Our software pipeline obtains output location estimates from DeepV2D, combines these into candidate trajectories, and uses a greedy approach to classify the best fit of a maneuver at any given time.

a driver-centric model based on a series of maneuvers using a sensor array [2] or detection of aggressive driving using deep learning techniques [6]. In contrast to our work, which requires only dashcam video with no additional metadata, this body of work typically utilizes smartphone sensors, such as an accelerometor, GPS, magnetometer, and gyroscope, which must be on-board the vehicle [27], [13].

Other work has been done with additional sensors, which allow for more fine-grained detection of maneuvers; for example, classification of lane changes (cut-ins and overtaking) using hidden Markov models (HMMs) with data collected from radar and LIDAR [14]. Work has also been done using HMMs to predict maneuvers [8].

B. Visual Odometry

Another significant body of work from the computer vision community has focused on the idea of *visual odometry* [17]. Borrowing ideas originating from SLAM, various techniques have focused on efficient ways to extract ego-vehicle motion from video, often for the purpose of mapping a route. ORB-SLAM is a modern monocular SLAM system that outputs camera motion [15]; our work builds on the contributions of this community. Specifically, DeepV2D [22], the deep learning algorithm we use to find camera motion, has operational similarity to ORB-SLAM.

III. Technique

Figure 2 shows an overview of our vehicle maneuver classification pipeline. We first use DeepV2D to extract translational and rotational camera movement at each frame to obtain a time series of X-Y coordinates corresponding to the ego-vehicle's trajectory. We then use simple heuristics to mark stop, reverse, and straight-line motions. We then compare trajectory segments against reference maneuvers using dynamic time warping to compute a distance measure. Finally, we perform a best-first match of turn maneuvers using a greedy approach. We will describe each of these steps in detail in the following sections.

A. DeepV2D

DeepV2D [22] is a deep learning network design for estimating camera motion and depth from video. DeepV2D consists of a Stereo Module, to perform stereo reconstruction from images and a camera motion estimate, and a Motion Module, which uses depth to estimate camera motion. The motion module finds initial estimates for the sequence of frames using a pose regression network to estimate the transformation parameters between images. These initial estimates are then refined in an iterative process using a projective warping function on the differentiable transformation of the input image to produce a warped feature map. The estimated feature map is then compared to the original image feature map and the difference is used to update the pose estimate for the next frame (see Figure 1).

We use a model of the DeepV2D architecture trained via RMSprop [24] and the Kitti dataset [4] (with ground truth motion estimated by ORB-SLAM2 [16]) to infer the motion of the ego-vehicle in our dataset. The input to DeepV2D is

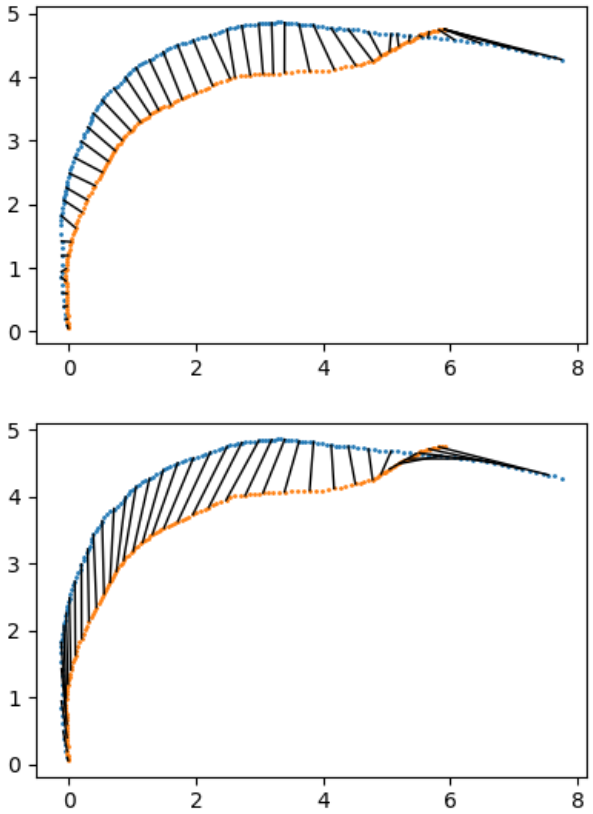

Fig. 3. An example of two right turns (in orange and blue) collected from our experiments. The location tracking is produced by DeepV2D and the maneuvers are automatically reoriented based on preceding motion. Rather than comparing maneuvers using Euclidean distance between discrete locations at a given time (top), we use Dynamic Time Warping (bottom), visualized here with lines connecting a sample of the match points.

a series of five video frames and the output is the estimated depth map and motion between the third and fourth frames. We therefore process each series of five frames from the recorded video library to obtain a motion track at the same framerate as the original video.

B. Dynamic Time Warping

Dynamic time warping [20] (DTW) is a measure of the similarity of two signals that may differ in duration. DTW performs a sequential matching between the signals while ignoring time differences. In effect, it "stretches" ("warps") one signal (or parts of it) to match another and computes the difference between matched values as the distance measure. (Figure 3 shows an example of two left turns). Whereas the original DTW algorithm is of $O(n^2)$ computational complexity (where n is the number of matched points), implementations such as FastDTW can approximate DTW at $O(n)$ complexity [19]. The reduction in computation time, along with comparative simplicity relative to other methods of pattern recognition, has enabled DTW to be widely and efficiently used for applications like speech recognition [10], gesture recognition [23], and detection of vehicle maneuvers [9]. We use DTW to quantify the similarity of vehicle trajectories from DeepV2D against the reference maneuvers.

Maneuver	Heuristic
Left Turn	Forward and horizontal distance traveled must be $\geq 80\%$ of reference maneuver and same direction
Right Turn	Forward and horizontal distance traveled must be $\geq 80\%$ of reference maneuver and same direction
U-Turn	Horizontal distance traveled must be $\geq 80\%$ of reference maneuver
K-Turn	Must contain at least half-second (15 frames) of reverse motion

TABLE I

HEURISTICS TO ACCELERATE CLASSIFICATION

	AUROC Value with Heuristic	AUROC Value without Heuristic
Left Turn	0.90	0.92
Right Turn	0.89	0.88
U-Turn	0.75	0.50
K-Turn	1.0	0.96
Processing time (seconds)	1098	7816

TABLE II

CLASSIFICATION ACCURACY WITH AND WITHOUT HEURISTICS

DeepV2D produces a three-dimensional rotation matrix and translation vector for each sequence of five frames. We found using DeepV2D to produce camera motion estimates at the original 30 frames-per-second of our test video produced the highest-quality motion estimates. We discard the Z component and keep a time series of ego-vehicle X-Y coordinates at each frame. Because of the high temporal resolution of the video, we can reconstruct camera motion estimates at equal or better resolution than is typically available from consumer-grade GPS devices.

C. Endpoint Detection and Candidate Maneuvers

Since we have no information about when a given maneuver may start or end, we perform automatic and simple endpoint detection for each DTW sequence. As the DTW matching process has no a priori information—knowing nothing about the positional track in advance—it must consider the possibility of a maneuver starting at every time step (video frame). We assume the length of a candidate maneuver can be 50% to 150% the length of the reference maneuver in 10% increments, which allows for normal variation in ego-vehicle velocity. We allow the start and end time to be any even-numbered frame in the recording, as we find this has no effect on accuracy. While the processing time to calculate the DTW matching score for an individual reference maneuver is relatively short, performing comparisons of all possible candidate maneuvers is computationally expensive. As a strategy to reduce processing time, we enforce a few simple filtering heuristics on the four types of turns processed with DTW (summarized in Table I). For left and right turns, the

overall motion must match 80% of the candidate maneuver's movement along the x- and y-axes. For U-turns, there must be sufficient x-axis movement, and K-turns (three point turns) require at least a half second (15 frames) of reverse motion. We find these simple heuristics slightly improve maneuver classification accuracy while reducing computational time by an order of magnitude (see Table II).

In addition to the maneuvers detected using dynamic time warping, we detect three additional maneuvers using other statistical techniques: Straight-line motion, no motion (stops), and reverse motion. We cannot use DTW to detect these maneuvers, as DTW requires sufficient variation in the compared signals to have any sensitivity. Using DTW for forward and reverse motion incurs too many false positives (e.g., portions of turns are straight), and stops cannot be represented as x-y motion. To detect reverse motion maneuvers heuristically, we simply find frame sequences where the Y-translation is negative. To tolerate noise and uncertainty, we enforce the condition that 25 of 30 frames in a 1-second sequence must be negative. To find stops, we look for one-second periods where Euclidean distance traveled is below a certain threshold (0.1 meter). To find straight-line forward motion, we use linear regression on 5-second periods of motion with a threshold of $R^2 \geq 0.985$. Examples of stop and straight-line forward maneuvers are shown in Figure 4.

D. Reference Maneuvers and Best-First Search

Our system allows the user to define a reference maneuver for as many categories as desired. DeepV2D extraction is expensive but need only be done once; a user can re-run subsequent analysis with a new set of reference maneuvers. For our evaluation, we define four: left turn, right turn, U-turn, and K-turn (examples are shown in Figure 4). (Note that the K-Turn looks odd because DeepV2D incorrectly estimates rotational speed in reverse, simply because the model isn't trained on reverse motion. However the error is consistent across maneuvers, so classification is still accurate.)

The user is responsible for defining the start and end point of each reference maneuver and selecting the baseline length for candidate maneuvers (which will then be scaled as appropriate). Each maneuver is effectively its own classifier: We compute all candidate DTW distance measures and select non-overlapping maneuvers with the lowest distance measures first. To evaluate quality, we use the area under the receiver operating characteristic curve (AUROC) [29]. The ROC curve is a plot of false positive rate vs. true positive rate, and integrating this curve gives a measure of quality of the classifier without requiring a numeric threshold. While ideally we would hope to have a high true positive rate and low false positive rate, we designed our classifier to increase the true positive rate even at the expense of additional false positives, as human observers can screen out false positives much faster than search for false negatives missed by the classifier.

IV. Evaluation Methodology

To evaluate our analysis flow, we collected dashcam video for processing with DeepV2D while simultaneously collecting high-resolution GPS positional information for a baseline comparison. We chose GPS as a fair comparison for location tracking, as GPS metadata is readily available in a number of consumer devices. We then manually labeled all recorded videos for elementary maneuvers, and selected a random subset of maneuvers from the recorded video to use as reference maneuvers. Finally, we performed cross-validation on five different samples of reference maneuvers to evaluate our best-first matching algorithm.

Our dashcam videos were recorded with a Garmin Dash Cam 55, fixed to the center of the windshield of a Toyota Sequoia SUV at approximately eye-height. This model records GPS location at a rate of 1 Hz, so we also used a GoPro Hero

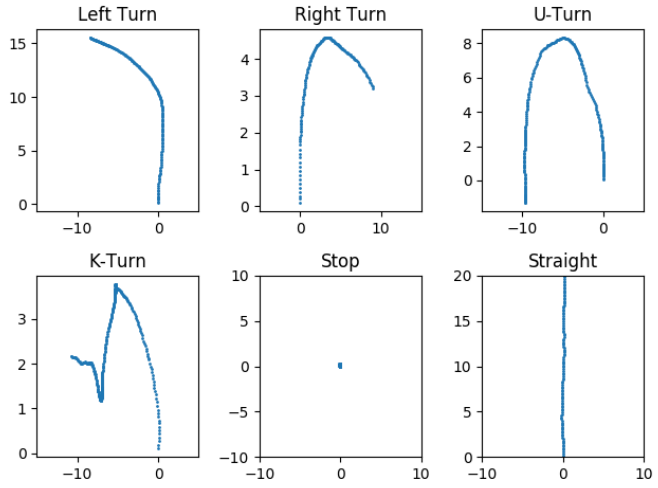

Fig. 4. Sample vehicle maneuver trajectories using location estimates from DeepV2D. X and Y scales are nominally in meters, though DeepV2D overestimates distance in some cases.

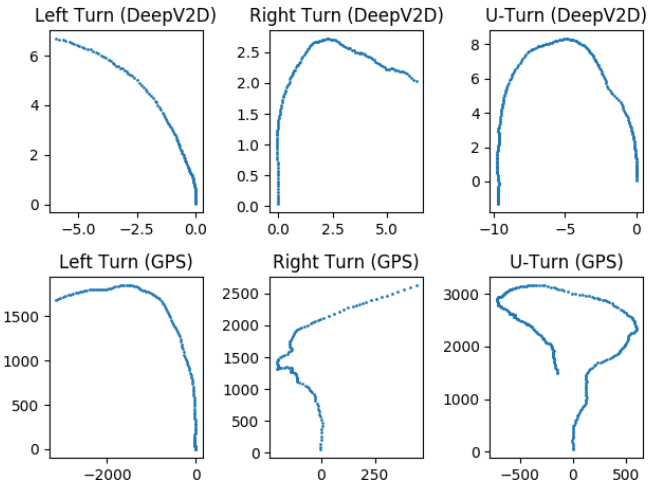

Fig. 5. Vehicle trajectories with motion estimated by DeepV2D compared to concurrent GPS measurements. Whereas neither technique perfectly matches the geometry of the maneuver, the DeepV2D data provides a more consistent match. DeepV2D scale is nominally meters, GPS scale is ten-millionths of degrees of latitude/longitude.

Data Collection Location	Ann Arbor, MI
Total Road Distance	32 miles
Maximum Speed	45 mph
Number of Left Turns	96
Number of Right Turns	76
Number of U-Turns	29
Number of K-Turns	26

TABLE III
DESCRIPTIVE STATISTICS OF OUR TEST DATASET

5 Black to obtain 18 Hz GPS metadata time-aligned with our video. The GoPro was fixed to the roof of the vehicle for better reception, and allowed a 10-minute warmup prior to recording to properly calibrate to the satellite signals.

We evaluate on a combination of commercial and residential streets in Ann Arbor, MI. Table III describes the road and driving conditions. A total of five videos were recorded, with each driving route containing approximately 50-60 maneuvers. These videos were then converted to JPEG frames for processing by DeepV2D, running on an Intel Core i7 workstation with an NVIDIA Titan XP GPU. This workstation ran Tensorflow 1.4 with CUDA 8.0 on Ubuntu 16.04. The higher-resolution GPS data extracted from the GoPro was time-aligned with the Dash Cam 55 video.

Prior to classification, the trajectory data must be reoriented. While DeepV2D motion estimation allows knowledge of the ego-vehicle's orientation at any arbitrary point, GPS does not. Orientation is important, as a maneuver cannot be detected if the positional track is rotated. Therefore, we estimate orientation from GPS data by taking the average direction vector of the half-second period prior to the start time of interest. To allow fair comparison, we use the same half-second reorientation algorithm for DeepV2D trajectories.

An example comparison of the DeepV2D positional data vs. GPS is shown in Figure 5. We find overall that DeepV2D data appears cleaner and more consistent than GPS data. While the GPS data is quite good, the typical GPS accuracy of approximately 3 meters [25] is simply not sufficient for tight maneuvers, as a standard U.S. highway lane is 3.7 meters wide [28] while city roads are narrower. Therefore, a maneuver like a U-turn is difficult to reconstruct accurately if the maneuver is performed on a two-lane road that is at most 8 meters wide. If the GPS device is located in the center of the vehicle, this may further reduce the width of the maneuver relative to the GPS resolution.

We selected ten percent of all maneuvers at random to be reference maneuvers. We manually selected ground truth endpoints for both GPS and DeepV2D data to best capture the given maneuver. We then used five-fold cross-validation with this randomly chosen maneuver set from each of five different videos to evaluate the system. Following the "leave one out" protocol, these reference maneuvers were not included in the overall statistical calculation.

V. RESULTS

We assess the accuracy of each maneuver's classification as follows: A reference maneuver of a particular category is

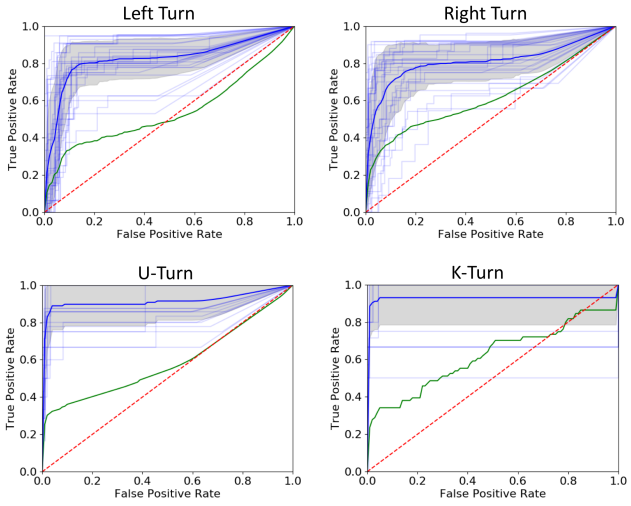

Fig. 6. Non-weighted average ROC of cross-validation DTW classification for DeepV2D motion estimates (blue) and GPS (green).

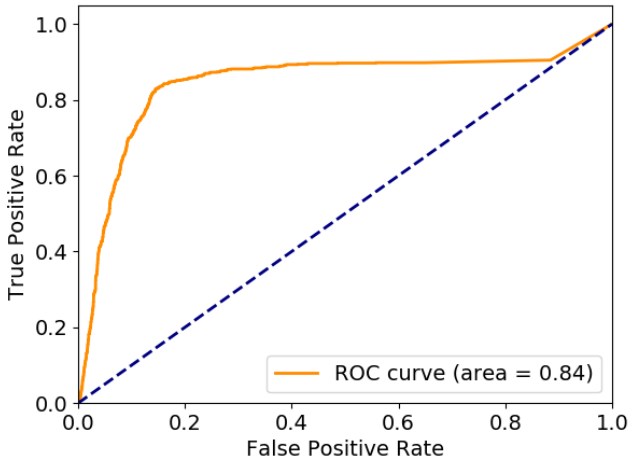

Fig. 7. Overall Receiver Operating Characteristic of greedy classifier.

chosen at random. The classifier then finds the best matches to this reference maneuver throughout the duration of the video, allowing no maneuvers to overlap. These candidate maneuvers are compared to the human-annotated ground truth. If a maneuver is detected within four seconds of the ground truth annotation, it is considered correctly classified; otherwise it is considered a false positive. We then compute the ROC curve using the list of DTW distance measures of detected maneuvers. False negatives (that is, ground-truth maneuvers not detected by the classifier) are included in the ROC calculation with the maximum-detected DTW value (to penalize the classifier for failing to detect these maneuvers).

The non-weighted average ROC curves [21] for each of the four maneuvers detected with DTW in this cross-validation experiment are shown in Figure 6. Each individual ROC curve for a particular maneuver (shown in thin blue lines) is averaged with equal weighting at each step (shown in the

Detected \ Actual	Left Turn	Right Turn	U-Turn	K-Turn	Stop	Reverse	Straight
Left Turn	59%	2%	23%	0%	0%	0%	0%
Right Turn	12%	55%	1%	0%	0%	0%	8%
U-Turn	6%	1%	61%	0%	0%	0%	0%
K-Turn	2%	1%	8%	100%	0%	0%	0%
Stop	0	0%	0%	0%	88%	14%	4%
Reverse	0%	0%	0%	0%	0%	82%	0%
Straight	0%	0%	0%	0%	12%	0%	88%
Forward/ Unclassifiable	20%	40%	7%	0%	0%	5%	0%

TABLE IV

CONFUSION MATRIX OF ACTUAL VS. DETECTED MANEUVERS. ACTUAL IS SHOWN AT TOP AND EACH COLUMN SUMS TO 100%.

Fig. 8. Example of a road curve to the left, which aliases as a left turn.

bold blue line). An interval of one standard deviation above and below the mean is shown in grey.

Figure 6 also shows in green the ROC curve for the same classifier using 18 Hz GPS data. We find the DeepV2D motion estimates consistently outperform our GPS measurements, due to the spatial and temporal resolution advantage of DeepV2D's 30 Hz estimates relative to the 18 Hz 3-meter resolution of the GPS. As seen in Figure 5, the overall jitter of the GPS trajectories means these maneuvers will typically have much higher distance measures and are less well oriented than the DeepV2D trajectories.

We use a greedy approach to classify the seven elementary maneuvers. Stops, reverses, and straight motion have highest priority, as these are the simplest maneuvers and least computationally expensive to detect. The greedy classifier then chooses up to two maneuvers of different types for each period of unlabeled time in the video recording. We allow the system to choose up to two maneuvers because often one maneuver can contain another: e.g., many U-turns have an embedded component that aliases as a left turn. This left turn often has a lower DTW distance measure than the U-turn itself, meaning that if we chose only one maneuver we will miss the correct one. As our goal is to avoid false negatives even at the expense of increasing the rate of false positives, we simply evaluate a greedy classifier allowed to choose up to two maneuvers, as this does not require defining additional heuristics or priorities for each maneuver.

The two-maneuver greedy approach has an overall AUROC of 0.84; the ROC curve is shown in Figure 7. A breakdown of all seven individual maneuvers is provided as a confusion matrix in Table IV. We find that the greedy classifier approach frequently has false positives for left and right turns compared to the labeled ground truth; these arise due to curves in the road. Figure 8 shows an example of a road curving to the left, which is detected by the classifier as a left turn even though a human labeler marked it as forward/unremarkable.

Left turns also have a higher prevalence of aliasing as U-turns and K-turns, as each of these begins with ego-vehicle movement to the left. To better distinguish curves from turns, we must fuse our trajectory-based method with other information sources, such as road segmentation or map data; we are pursuing such multimodal data fusion in ongoing work. As previously mentioned, U-turns often alias with left turns, and K-turns overall are quite accurately detected due to the reverse motion not present in other maneuvers.

The AUROC is higher than the confusion matrix results would indicate because it accounts for the DTW distance metric (confidence) in classifying each maneuver. It is important to note that our classifier uses only one reference maneuver from each maneuver category for classification. It is therefore unrealistic to expect this approach to work perfectly, as the geometry of intersections can vary quite substantially in terms of both size and angle of intersection.

VI. FUTURE WORK

We have demonstrated a mechanism for indexing vehicle maneuvers by trajectory, but further work remains to enable dashcam video search more broadly. First, we wish to extend our vehicle maneuver work from surface roads to highways, allowing detection of maneuvers such as merges and exits. Second, we intend to utilize other modern deep learning techniques to enable new video search capabilities. For example, use of depth estimation to allow detection of lane changes, and object detection and geometric semantics to discover interactions with other vehicles and pedestrians. Finally, we are interested in optimizing the computational efficiency of this end-to-end system and exploring new ways to accelerate it.

VII. Conclusions

We have designed and implemented a system that determines vehicle trajectory state in terms of seven common road maneuvers. Movement is derived via a deep learning model from monocular dashcam video, which produces translational and rotational motion for each frame. These instantaneous movement vectors are processed into candidate maneuvers. Following detection of stops, reverses, and straight-line motion segments, we detect turns via dynamic time warping, computing a distance measure for each maneuver. We evaluate the best choice for each possible maneuver category using a greedy classifier on the distance measures. We test our greedy classifier using cross-validation and find it has an overall AUROC value of 0.84. Finally, we utilize our turn classifiers on high-resolution GPS data collected in parallel with the dashcam video and find that that DeepV2D's estimated motion allows for more accurate classification of vehicle motion.

Acknowledgements We thank German Ros for his advice in the early stages of the project, Zach Teed and Jia Deng for their assistance using DeepV2D, and Thomas Larsen for his technical assistance on the project. This work was funded by the Toyota Research Institute and by NSF Grant IIS-1539011.

References

[1] Mayank Bansal, Alex Krizhevsky, and Abhijit Ogale. Chauffeurnet: Learning to drive by imitating the best and synthesizing the worst. *arXiv preprint arXiv:1812.03079*, 2018.

[2] Hien Dang and Johannes Fürnkranz. Using past maneuver executions for personalization of a driver model. In *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, pages 742–748. IEEE, 2018.

[3] Haluk Eren, Semiha Makinist, Erhan Akin, and Alper Yilmaz. Estimating driving behavior by a smartphone. In *2012 IEEE Intelligent Vehicles Symposium*, pages 234–239. IEEE, 2012.

[4] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[5] Robert Geirhos, David HJ Janssen, Heiko H Schütt, Jonas Rauber, Matthias Bethge, and Felix A Wichmann. Comparing deep neural networks against humans: object recognition when the signal gets weaker. *arXiv preprint arXiv:1706.06969*, 2017.

[6] Jingqiu Guo, Yangzexi Liu, Lanfang Zhang, and Yibing Wang. Driving behaviour style study with a hybrid deep learning framework based on gps data. *Sustainability*, 10(7):2351, 2018.

[7] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017.

[8] Adam Houenou, Philippe Bonnifait, Véronique Cherfaoui, and Wen Yao. Vehicle trajectory prediction based on motion model and maneuver recognition. In *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4363–4369. IEEE, 2013.

[9] Derick A Johnson and Mohan M Trivedi. Driving style recognition using a smartphone as a sensor platform. In *2011 14th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, pages 1609–1615. IEEE, 2011.

[10] B-H Juang. On the hidden markov model and dynamic time warping for speech recognitiona unified view. *AT&T Bell Laboratories Technical Journal*, 63(7):1213–1243, 1984.

[11] Jair Ferreira Júnior, Eduardo Carvalho, Bruno V Ferreira, Cleidson de Souza, Yoshihiko Suhara, Alex Pentland, and Gustavo Pessin. Driver behavior profiling: An investigation with different smartphone sensors and machine learning. *PLoS one*, 12(4):e0174959, 2017.

[12] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015.

[13] Clara Marina Martinez, Mira Heucke, Fei-Yue Wang, Bo Gao, and Dongpu Cao. Driving style recognition for intelligent vehicle control and advanced driver assistance: A survey. *IEEE Transactions on Intelligent Transportation Systems*, 19(3):666–676, 2018.

[14] John Martinsson, Nasser Mohammadiha, and Alexander Schliep. Clustering vehicle maneuver trajectories using mixtures of hidden markov models. In *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, pages 3698–3705. IEEE, 2018.

[15] Raul Mur-Artal, Jose Maria Martinez Montiel, and Juan D Tardos. Orb-slam: a versatile and accurate monocular slam system. *IEEE transactions on robotics*, 31(5):1147–1163, 2015.

[16] Raul Mur-Artal and Juan D Tardós. Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras. *IEEE Transactions on Robotics*, 33(5):1255–1262, 2017.

[17] David Nistér, Oleg Naroditsky, and James Bergen. Visual odometry. In *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004.*, volume 1, pages I–I. Ieee, 2004.

[18] Alex Poms, Will Crichton, Pat Hanrahan, and Kayvon Fatahalian. Scanner: Efficient video analysis at scale. *ACM Transactions on Graphics (TOG)*, 37(4):138, 2018.

[19] Stan Salvador and Philip Chan. Toward accurate dynamic time warping in linear time and space. *Intelligent Data Analysis*, 11(5):561–580, 2007.

[20] D Sankoff and J Kruskal. The symmetric time-warping problem: from continuous to discrete. In *Time Warps, String Edits and Macromolecules: The Theory and Practice of Sequence Comparison*, pages 125–161. Addison Wesley Publishing Company, 1983.

[21] Scikit-learn. Receiver operating characteristic (roc) with cross validation. https://scikit-learn.org/stable/auto_examples/model_selection/plot_roc_crossval.html. Accessed: 2019-04-14.

[22] Zachary Teed and Jia Deng. Deepv2d: Video to depth with differentiable structure from motion, 2018.

[23] Gineke A Ten Holt, Marcel JT Reinders, and EA Hendriks. Multi-dimensional dynamic time warping for gesture recognition. In *Thirteenth annual conference of the Advanced School for Computing and Imaging*, volume 300, page 1, 2007.

[24] Tijmen Tieleman and Geoffrey Hinton. Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. *COURSERA: Neural networks for machine learning*, 4(2):26–31, 2012.

[25] Navigation U.S. Air Force National Coordination Office for Space-Based Positioning and Timing. Gps.gov: Gps accuracy. https://www.gps.gov/systems/gps/performance/accuracy/#how-accurate. Accessed: 2019-04-16.

[26] The Verge. How tesla and waymo are tackling a major problem for self-driving cars: data. https://www.theverge.com/transportation/2018/4/19/17204044/tesla-waymo-self-driving-car-data-simulation. Accessed: 2019-04-16.

[27] Johan Wahlström, Isaac Skog, and Peter Händel. Smartphone-based vehicle telematics: A ten-year anniversary. *IEEE Transactions on Intelligent Transportation Systems*, 18(10):2802–2825, 2017.

[28] Wikipedia. Lane. https://en.wikipedia.org/wiki/Lane. Accessed: 2019-04-15.

[29] Wikipedia. Receiver operating characteristic (roc). https://en.wikipedia.org/wiki/Receiver_operating_characteristic#Area_under_the_curve. Accessed: 2019-04-14.

[30] Fisher Yu, Wenqi Xian, Yingying Chen, Fangchen Liu, Mike Liao, Vashisht Madhavan, and Trevor Darrell. Bdd100k: A diverse driving video database with scalable annotation tooling. *arXiv preprint arXiv:1805.04687*, 2018.