This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.



# Creating Semantic HD Maps From Aerial Imagery and Aggregated Vehicle Telemetry for Autonomous Vehicles

Yijun Wei, *Member, IEEE*, Faria Mahnaz, Orhan Bulan, *Senior Member, IEEE*, Yehenew Mengistu, Sheetal Mahesh, and Michael A. Losh, *Member, IEEE*

***Abstract*— We present a new framework for creating lane-level detailed HD-maps at scale for autonomous vehicles (AVs). In order to overcome scaling challenges of ground survey-based HD-map creation, we propose a number of innovations by leveraging three data sources: high-resolution aerial imagery, aggregated vehicle telemetry, and a navigation map. We first divide map creation problem into several categories based on lane configurations. The road category is predicted in a supervised setting using aerial imagery, which is pre-processed by using aggregated vehicle telemetry without supervision. We utilized the navigation map to guide the process along the road network and used aerial imagery and aggregated vehicle telemetry to extract lane level features at each step. We propose a multi-task convolutional neural network (CNN) to predict road polygons, road-way centerline, number of lanes, lane and road edges using both the aerial imagery and the corresponding aggregated vehicle telemetry. The predicted road features for each image are then stitched along a road segment to construct the road and lane edge polylines, which are then used to predict lane marking and road edge types in a sliding window fashion along the road segment. The extracted features are finally utilized to calculate higher level features for each point in the HD-map. Our experimental results show that the proposed framework works well, offering a flexible solution for creating HD-maps for AVs at scale.**

***Index Terms*— Autonomous vehicles, clustering, convolutional neural networks, HD-maps, machine learning, remote sensing, support vector machines, telemetry.**

## I. Introduction

OVER the last 2-decades, significant progress has been made in AV development. Some of these systems have been already in production as Society of Automotive Engineers (SAE) autonomy Level-2 systems [1], enabling hands-free driving in certain conditions and types of roads (e.g., motorways, highways). One of the key factors for AVs adoption and success in commercial and retail markets is to allow them to work wherever people want to go, on most types of roads. High-Definition (HD) maps are key enablers for safe and efficient AVs operation, so producing them at a comprehensive, even continental scale is paramount. HD-maps provide lane-level detailed information for precise localization and pose estimation, AV maneuver planning, and provide redundant information to the vehicle to compensate for the weakness of on-vehicle sensors in situations where their information is less accurate or unreliable (e.g., temporary occlusion) [2], [3].

A common way to create HD maps is ground survey with mapping vehicles equipped with a suite of sensors such as high density lidars, high resolution cameras, and a high precision GPS (e.g., RTK), among others [4]–[7]. While this methodology can produce highly accurate data, scaling-up a ground survey campaign gets extremely expensive and time-consuming if employed at a national or continental level. Recently, there have been efforts to create HD-maps using alternative data sources [8], [9], [24]–[26] without precise ground survey and complementing it with "crowd-sourcing" data from a ground vehicle fleet to fill in some missing features (e.g., traffic signs and traffic control devices) [10], [27]. These activities provide an alternative approach to create centimeter-level precise maps at regional, national or continental scale.

In this paper, we present a new framework for creating HD-maps at scale using high resolution aerial imagery, aggregated vehicle telemetry and a navigation map. Our proposed methodology includes several innovations that combine these three data sources to extract several HD-map features such as lane edges, road edges, roadway centerline, lane marking type and road edge type. In our system, we first categorize the roads into five different categories: a) divided one-way (e.g., motorway and highway), b) two-way separated with a double yellow marking, c) two-way with a shared center lane, d) unmarked residential, and e) *others*. We formulated road categorization as a classification problem using aerial imagery and aggregated vehicle telemetry, where the latter is used to create a mask to filter out background information from the aerial image. This categorization enables us to create specialized processing for each road type that is more fully automated and better detects failures in the map creation process. We utilized a navigation map to virtually "drive" along the road network, guiding the processing of aerial imagery and aggregated vehicle telemetry

Manuscript received July 16, 2021; revised November 8, 2021; accepted December 31, 2021. The Associate Editor for this article was S. Selpi. *(Corresponding author: Yijun Wei.)*

Yijun Wei, Faria Mahnaz, Orhan Bulan, Yehenew Mengistu, and Sheetal Mahesh are with General Motors Information Technology, Warren, MI 48083 USA (e-mail: yijun.wei@gm.com; faria.mahnaz@gm.com; orhan.bulan@gm.com; yehenew.mengistu@gm.com; sheetal.mahesh@gm.com).

Michael A. Losh is with the General Motors Research and Development Electrical Controls Systems Laboratory, Warren, MI 48083 USA (e-mail: michael.losh@gm.com).

Digital Object Identifier 10.1109/TITS.2022.3140423

1558-0016 © 2022 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.
See https://www.ieee.org/publications/rights/index.html for more information.

Authorized licensed use limited to: Queen's University. Downloaded on February 21,2022 at 05:45:38 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.



to extract lane-level features at each step. For the current road to be mapped and given an aerial image and the corresponding telemetry data, our proposed multi-task network predicts the road polygons and lane and road edges while filtering out other lane and road edges that may be present in the same image. These other lane and road edges in the scene will be detected and incorporated into other mapped roads or ignored if out of scope for the intended map coverage. The estimated lane and road edge polylines, road polygons, and the road centerline are then processed and stitched for each *squiggle* (i.e., small road segment between two intersections in the road network). The lane marking types and road edge types for the predicted squiggle's polylines are then predicted using a classifier in a sliding window fashion. In order to evaluate the performance of the proposed methodology we propose a metric that compares the predicted polylines with the ground truth polylines, obtained by the ground survey, in absolute coordinates. We then leverage predicted polyline features to calculate high level attributes such as curvature and heading to complete the HD map. In our experimental results, we show that the proposed framework works well, offering a flexible and practical solution for creating HD maps at large scale.

Our contributions in this paper are as follows:

- We propose a complete framework for creating semantic HD maps from aerial imagery, aggregated vehicle telemetry and a navigation map. To the best of our knowledge, this is the first work that describes an end-to-end pipeline to create HD maps from these data sources.
- We propose a new neural network architecture to predict road features essential for creating HD-maps using aerial imagery and aggregated vehicle telemetry as primary data sources.
- We propose a road categorization methodology using both aerial imagery and aggregated vehicle telemetry as a way of avoiding errors that can possibly occur in the map-build process.
- We propose and describe a post-processing approach for stitching inference results along short road segments between intersections and also,
- We propose and describe a method to predict lane marking and road edge types for the constructed lane/road edges.
- We propose a new metric to compare predicted road feature polylines with their ground-truth representation.

The remainder of this paper is organized as follows. Section II describes our lane and road edge detection using aerial imagery and aggregated vehicle telemetry. Section III presents our road categorization approach. Section IV presents our lane marking and road edge type detection methods. Our experimental results for the individual sub-processes are presented in Section V. Section VI concludes the paper by summarizing our main findings.

## II. Lane and Road Edge Detection

### A. Data Preparation and Inference

Lane and road edges constitute an important part of HD-maps utilized by several on-vehicle modules in the AV software stack to determine the drivable areas of the roadway.

Fig. 1. Aerial image centered around a node in the road network with vehicle telemetry overlay.

In order to find the lane and road edges, we use two data sources: high resolution aerial imagery and aggregated vehicle telemetry. Additionally, we also utilize a road network, defined by a set of ordered points $N_1, N_2, \ldots, N_i$ along a road of interest (i.e., edge in the road graph) of a navigation map. These points are typically non-evenly spaced in the road graph. We interpolated these ordered points at the desired sample distance (e.g. $d = 10$m) to create evenly spaced points along the road. The distance $d$ between points is selected to allow sufficient overlap between successive images to ensure continuity of the extracted features. Each interpolated node serves as the center point for an aerial image we construct. At first, a given node point with (*lat*, *lon)* values is converted to tile *(x, y)* values at zoom level $z$ using Mercator projection [11]. Based on the relative position of the point in its tile, up to three neighbor tiles are identified and stitched to the first and cropped in order to form a $512 \times 512$ RGB image centered at each $N_i$ point. This ensures the road of interest lies at the center of the image. Finally, we generate a geolocation array with wgs84 coordinates of each pixel of the stitched aerial image using Inverse Mercator projection [11], which allows us to map pixelwise road feature inferences to real world coordinates.

Corresponding to each stitched aerial image constructed along the road network, we also fetch the aggregated vehicle telemetry, which is "crowdsourced" vehicle trajectory information sampled at regular intervals, including vehicle geolocation, speed, heading and elevation. Aggregated trajectories are selected that fall within the aerial image boundaries. The vehicle telemetry points are then matched to pixel locations in the corresponding aerial image. Fig. 1shows aggregated vehicle telemetry data overlaid on top of the aerial image centered around the road of interest.

Using these data sources, we propose a two-stage multitask CNN with multiple branches to detect road polygons, roadway centerline, relevant lane and road edges, and number of lanes in a single inference as shown in Fig. 2. While the first stage of the CNN focuses on detecting features in the scene, the second stage specializes finding features on the road centered. The features detected on the second stage are assigned to the road being processed, which enable us

Authorized licensed use limited to: Queen's University. Downloaded on February 21,2022 at 05:45:38 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.



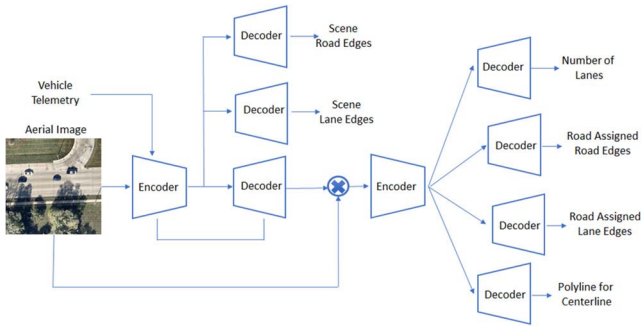



Fig. 2. Two-stage multi-task CNN for detecting lane and road edge features and road level attributes.

to aggregate detected features as we navigate along a road segment in the road network.

Our proposed network consists of multi-branch encoder-decoder architecture [12], where the encoder is used to map high level aerial image and aggregated vehicle telemetry features into an embedding space. The decoder at each branch specializes to learn desired feature maps from the embedding space. In our specific implementation, we utilized ResNet-34 as the shared encoder removing final two fully-connected layers. For the decoders in the first stage, we utilize an architecture similar to LinkNet/E-Net [13], [14] where the layers are linked between encoder and decoder. We also used dilated convolutions at each layer to capture subtle granularity of the lane and road edge features [15]. The second stage of the network starts with extracting the road polygon from the aerial image so that the features extracted at this stage can be assigned to the road being processed to ensure the continuity of the features as we navigate along the road segment. At this stage, the network also predicts the number of lanes and a polyline for the centerline of the road in addition to the lane and road edges to help us mitigate inference errors as we stitch features together.

For calculating the loss for the segmentation branches, we use soft-dice-loss [16] calculated as:

$$loss_{\text{softDice}}\left(\boldsymbol{x}, \hat{\boldsymbol{x}}\right) = 1 - \frac{2\sum \boldsymbol{x}\hat{\boldsymbol{x}}}{\sum \boldsymbol{x}^2 + \sum \hat{\boldsymbol{x}}^2} \tag{1}$$

where $\boldsymbol{x}, \hat{\boldsymbol{x}}$ represent manual annotations and network predictions on the image plane for the segmentation tasks, respectively.

Using soft-dice instead of intersection-over-union (IoU) provided us slightly better results. For the branch estimating the number of lanes, we initially used l2-norm [17] for calculating the loss. This approach, however, was sensitive to outliers that existed in our training set because some of our ground truth data is for roads that had changed between the time aerial images were captured and the road surveyed by lidar-scanning. These outliers caused significant errors during our training and inhibited the network from finding the optimal solution. We addressed this issue by linearizing the l2-norm in regions where the l2-norm is greater than 1.0 [18], which enabled us to smooth out the impacts of large errors due to outliers in the backpropagation, where the loss function is expressed as:

$$loss_{\text{reg}}(\mathbf{x} - \hat{\boldsymbol{x}}) = \begin{cases} ||\boldsymbol{x} - \hat{\boldsymbol{x}}||^2 & if \ \left|x - \hat{x}\right| \leq 1 \\ \left|\left|\boldsymbol{x} - \hat{\boldsymbol{x}}\right|\right| & otherwise \end{cases} \tag{2}$$

where $x, \hat{x}$ represent ground truth and prediction, respectively.

For the branch predicting the centerline polyline, we utilize a geometric loss by applying a function that approximates least squares fitting [19] that minimizes the area between the predicted and ground truth (GT) polylines. Representing the polyline with a parabolic function denoted by $p_0 + p_1 * x + p_2 * x^2$, where $x$ is the independent variable to make the prediction, the geometric loss is calculated as:

$$\begin{aligned} loss_{\text{geometric}}\left(p_0, p_1, p_2\right) = & \frac{\Delta p_2^2 M^5}{5} + \frac{2\Delta p_1 \Delta p_2 M^4}{4} \\ & + \frac{(\Delta p_1^2 + 2\Delta p_0 \Delta p_2) M^3}{3} \\ & + \frac{2\Delta p_0 \Delta p_1 M^2}{2} + \Delta p_0^2 M \end{aligned} \tag{3}$$

where $M$ is the extent of the image and $\Delta p_i = p_i - \hat{p}_i$ shows the difference between the GT and predicted coefficients. The final loss is calculated as the weighted average of the loss from each branch and backpropagated through the network end-to-end. Putting these together, the total loss metric is expressed as:

$$loss_{\text{tot}} = \gamma_1 * loss_{\text{softDice}} + \gamma_2 * loss_{\text{reg}} + \gamma_3 * loss_{\text{geometric}} \tag{4}$$

where $\gamma_1, \gamma_2, \gamma_3$ in the total loss are the hyper-parameters.

### B. Inference Projection

Following inference, the lane and road edge inferences are projected from image coordinates to wgs84 coordinates using the array of corresponding geolocations. To achieve this, small non-overlapping polygons hereby called segments are created between consecutive interpolated points $D_1, D_2, \ldots, D_m$ as;

$$S_i = \text{Segment}\left(D_i, D_{i+1}\right), \quad for \ 0 < i < m - 1 \tag{5}$$

These polygons cover a cross-section of the road for the longitudinal distance between two consecutive points $D_i$ and $D_{i+1}$. The inference outputs from the network for lane and road edges are then binarized and detected pixels are aggregated using their pixelwise (*lat,lon*) values. These aggregated pixels are then projected to each non-overlapping segment. Projected points $\boldsymbol{T}$ at segment $S_i$ are represented as follows;

$$\boldsymbol{T_i} = F\left(S_i, \boldsymbol{P}\right), \quad for \ 0 < i \leq m - 1 \tag{6}$$

where $F$ is a projection function, $S$ is a segment polygon and $\boldsymbol{P}$ is a set of $k$ total aggregated points for the edge, respectively. This allows for generating lane lines with higher confidence by overlapping inferences in a sliding window. Fig. 3illustrates our process for the inference projection along a road segment. 3(a) indicates that the road segment is divided into several pieces using the sliding window, and road edges and lane edges are inferenced on each sliding window. Then all these road and lane edges are stitched together to project to segment polygons in 3(b).

Authorized licensed use limited to: Queen's University. Downloaded on February 21,2022 at 05:45:38 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.



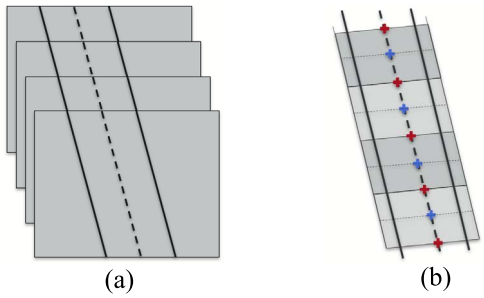



Fig. 3. (a) Lane and road edge inferences in a sliding window. (b) Inference outputs for lane and road edges projected to segment polygons.

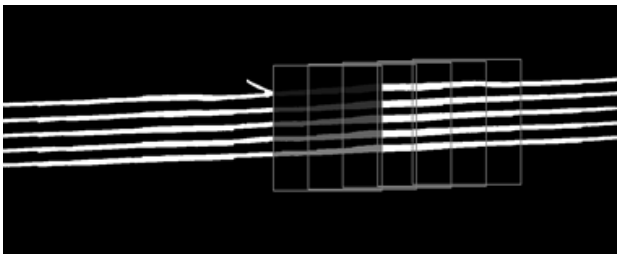


Fig. 4. Clustering applied in a sliding window.

### C. Generating Lane and Road Edges

Lane lines are created per segment polygon from the previous step in a sliding window. Projected points inside current and next polygons are clustered to identify points belonging to the same line. The density-based clustering algorithm *DBSCAN* [20] is used with experimentally selected clustering parameters epsilon and minimum number of points for a cluster. The number of points per cluster and total aggregated inference confidence is used to reject clusters which are less likely to represent a useful road feature. A cubic polynomial is then fitted to points in each cluster. Cubic polynomials are chosen as they insure smooth trajectories along the longitudinal direction of the road. We then sample the polynomials at equally spaced cross-sections along the direction of the edge. The final output from this stage will be uniformly sampled points hereby referred to as knots with line numbers determined by clustering. This step is repeated in a sliding window for the extent of the edge as shown in Fig. 4.

### D. Stitching Lane and Road Edges

Stitching lane lines involves matching lane lines, filling missing gaps and smoothing. Lane lines are matched in a sliding window using a nearest neighbor search. Points on the last cross-section of the current segment are evaluated against the points on the first cross-section of the next segment. Points with a neighboring distance less than a threshold are matched and lines are numbered. The matched points are averaged to smooth the line. This process is repeated for all processing polygons in a sliding window. This process results in a set of continuous lane lines with the following features;

i. *Line numbers*: used to identify unique lines
ii. *Cross-sections*: used to identify ordering of points along the line

Fig. 5. Final lane and road edges output.

iii. *Latitude and longitude*: For identifying the spatial location of each point

While this process ensures continuity of lane lines, missing portions of a line due to occlusion in the aerial image or insufficient feature representation create discontinuities and result in multiple disjoint lines. To minimize this, a gap filling algorithm is utilized. The gap filling algorithm scans through the edge to identify missing gaps or unconnected lines. Cross-sections of the edge before and after the gap are identified. Lane polygons are created for the cross-sections before and after the gap. Next, trajectories are mapped to the lane polygons using telemetry-derived individual trajectories. The lane mapping resulting in the highest number of vehicle trajectories is selected as the nominal path between the lanes. The center of the trajectory is then projected to rightward and leftward to create lane boundaries consistent with the edges detected before and after the gap.

### E. Creating Splines

The final step in the lane and road edge boundary creation is fitting splines to stitched lane lines. We take ordered list of points $P_{i,j}$ where $i$ is the line id and $j$ is the cross-section id. These points are then used as knots to fit a cubic spline using least square spline approximation [21].

The spline is finally evaluated at uniformly spaced points along the line to create a smooth lane boundary as shown in Fig. 5.

## III. Road Categorization

In creation of HD maps, road category information can play an important role in reducing the number of false positives and also in identifying locations with missed detections of road features. For example, if the map creation process knows that the road being processed is an unmarked residential road then it can filter out or flag a lane edge marking feature detection as unlikely. Similarly, for a road with two-way traffic separated by a double yellow stripe, the map creation process can flag a gap where the double-yellow stripe is not detected.

Authorized licensed use limited to: Queen's University. Downloaded on February 21,2022 at 05:45:38 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.



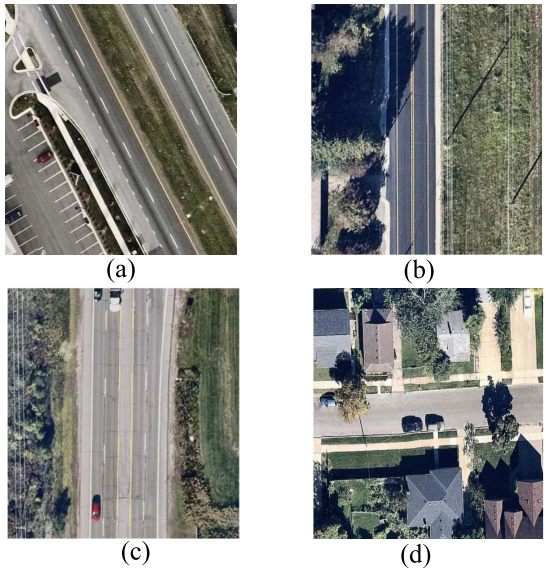



Fig. 6. (a) One-way road, (b) two-way separated with a double/dashed yellow, (c) two-way with a shared lane, and (d) unmarked residential.

Hence, categorizing the roads into meaningful classes can help improve the autonomation of the map creation process and identify possible failures.

We note that navigation maps provide road category information for most roads in the road network based on the speed and the volume of the traffic or its general purpose in a traffic engineering sense. While this categorization can be useful for navigation purposes, it is not based consistently on presence/absence or arrangement of specific visual road features. The same physical presentation of road features (e.g., a two-way road separated by double yellow/dashed yellow stripe) can and is assigned a variety of road categories in navigation maps at various locations, even with, similar traffic load and speed characteristics. Conversely, any given road category is likely to be assigned to roads with some significant differences in road features and basic characteristics. In short, the road category assignment for roads found in today's navigation maps is not consistent and conclusive enough to be utilized directly an HD map without confirmation and possible revision by analysis of additional data sources. Furthermore, depending on the needs of automotive system features using map data, a different categorization scheme may be more appropriate.

In this paper, we categorize the roads into five groups based on their lane configurations. Fig. 6 presents examples for each road type, which include (a) one-way roads, (b) two-way roads separated with a double/dashed yellow line, (c) two way roads with a shared middle lane, (d) unmarked residential roads, and the roads that do not fall in any of the categories above is defined as (e) *other* road types. Since one-way roads tend to be accurately labeled in navigation maps, we specifically focus our attention on categories (b) through (e).

We formulated road categorization as a classification problem using aerial imagery and aggregated vehicle telemetry as primary inputs, where the latter is used to create a mask to emphasize likely roadway area and filter out background information from the aerial image used in road categorization. We note that filtering out background information from aerial images can also be performed using a semantic segmentation network [22], [23] in a supervised fashion and only the segments of the aerial image assigned a semantic type for "road" would proceed to a road type categorization network in a way similar to that proposed in this section. Training a segmentation network, however, requires additional manual annotation to label road boundaries in aerial images, which is inefficient in terms of cost and time. We propose an alternative methodology to pre-process aerial images using aggregated vehicle telemetry data without supervision and show that a notable improvement can be achieved for road categorization compared to using aerial images alone.

In the mask creation process, we first identify a $c \times c$ dimensional square polygon $P_c$, centered at the given image. Then we cluster all aggregated vehicle telemetry data points, $x_1, \ldots, x_n$ related to that image into N different clusters,, using HDBSCAN [20], and create N convex hulls, $h_1, \ldots, h_N$ for each of the clusters.

For each convex hull, we calculate the intersection area $r_i (0 \leq \mathrm{i} \leq \mathrm{N})$, between the center polygon $O_c$ and respective convex hull, i.e.,

$$r_i = h_i \cap O_c \tag{7}$$

We later selected only the clusters if their intersections with the center polygon is greater than a given constant threshold. For each of the selected cluster, a cubic spline [21] is fitted to estimate the centerline of the cluster. The estimated centerline for the cluster is shifted to the left and right perpendicular to the direction of traffic in a way that 95% of the telemetry points in the respective cluster falls between the shifted splines. These shifted splines roughly estimate the left and right road edges for the telemetry points within the cluster. Finally, we stitched the estimated left and right road edges across the clusters to create the polygon for the road of interest.

Once we generate all the polygon masks, we multiply them with their corresponding aerial images to create the input data for road categorization classification model. We then used the masked images to train an image classifier for road categorization.

## IV. Lane Marking and Road Edge Type Prediction

Lane marking type and road edge type are important attributes in HD maps because this information is used in several functions in the AV software stack, such as localization. While finding white markings on a dark road seems simple, in practice several factors can make it challenging to determine lane marking types on various types of road. Difficulties come from shadows, occlusion by vehicles, changes in the road surface texture and material composition, noticeable cracks and seams, patches and sealants applied during road maintenance, presence of dirt or salt over or near markings, varying marking widths, and differing materials and colors used as markings. The same can be said for road edges, which are often obscured by overhead trees or other vegetation, may

Authorized licensed use limited to: Queen's University. Downloaded on February 21,2022 at 05:45:38 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.



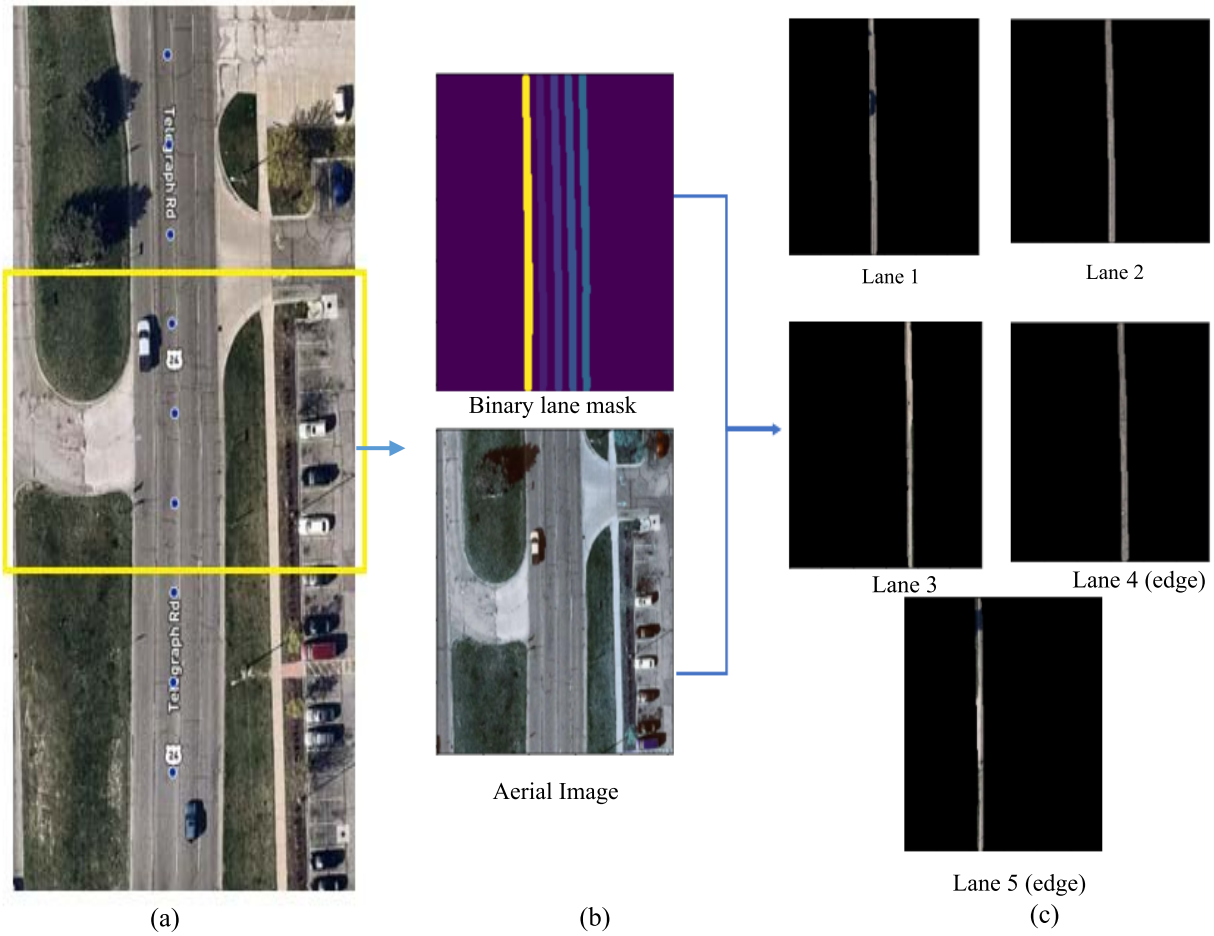



Fig. 7. (a) A given input squiggle with one random sliding window marked in yellow. (b) An aerial image corresponding to that sliding window and masks generated for each lane and road edge lines. (c) Images of each lane and road edge created by multiplying the masks per lane and respective image which will be used as final input for LMT/RET inference module.

have a wide variety of artificial and natural materials beyond the road surface, and/or have three-dimensional characteristics that cast shadows.

A lane marking type and road edge type detection system must be able to classify all manner of lane and road edges in a variety of conditions and challenges as suggested above. In the described system, each lane marking is classified into one of five categories, a) solid single line, b) solid double line, c) dashed-solid (or) solid-dashed line combination, d) dashed single line or e) virtual inferred line (no physical marking or unknown type). Likewise, each road edge is classified into one of three categories: a) barrier, b) curb, or c) pavement. Note that the color type of lane markings are not determined in the proposed detection model because aerial images often have limited color fidelity and colors of markings are largely proscribed by published standards and verifiable by behavior of vehicles observed in aggregated vehicle telemetry.

For Lane Marking Type (LMT) and Road Edge Type (RET) detection, the initial step is to train an image classification model using aerial images of all road types such as highway, primary, secondary, tertiary, and residential with annotated lane marking and road edge types. In this study, we used the ResNet [29] model. To increase performance, transfer learning is applied and Resnet101 model pretrained with ImageNet [32] has been used as base pre-trained model. The model is then further pre-trained and fine-tuned on LMT/RET task. Once the network has converged, the model is converted to an inference model for predicting the type class labels of each lane and road edge line.

As explained in the previous section, the lane and road edge inference module detects the position and shape of edge lines for a given road segment (i.e., *squiggle*) in a sliding window fashion. Each window corresponds to one aerial image and the inference module is applied to each of them for detection. The LMT/RET inference module takes the detected lane and road edge lines and creates binary masks for each of them according to their geolocation coordinates registered to the corresponding aerial image. Multiplying these masks with the original aerial image then generates an image per lane line. The LMT/RET inference model is applied to each of these images to assign a class label for lane marking and road edge types. Fig. 7 illustrates an overview of these data processing steps for a sample sliding window. In the demonstrated case, the module assigned the class Solid Dashed Line to lane line 1 to 3 and Solid Single Line to lane line 4 and 5 as its final output.

Authorized licensed use limited to: Queen's University. Downloaded on February 21,2022 at 05:45:38 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.



Assuming, we have $n$ total lane lines, $L_1, \ldots, L_n$ for a given squiggle, each lane line $L_t$, $1 \leq t \leq n$, has $N_{L_t}$ windows (masks), represented by $W_{L_t,1}, \ldots, W_{L_t,N_{L_t}}$. The polygon $W_{L_t,i}$ is multiplied with the corresponding image to generate an image patch to which the classifier is applied. A class $\hat{C}$ is predicted for the image patch where $\hat{C} \in C$, the total set of classes.

For a given polygon, $W_{L_t,i}$, all the interpolated points from the navigation map, that fall within the polygon are assigned with the class $\hat{C}_{L_t,l}$ predicted for the polygon. Assuming a given interpolated point will appear in 10 overlapped windows, $W_{L_t,j-9}, \ldots, W_{L_t,j}$, each with a predicted class, i.e., $\hat{C}_{L_t,j-9}, \ldots, \hat{C}_{L_t,j}$. The final class is determined as:

$$argmax(\hat{C}_{L_t,j-9}, \ldots, \hat{C}_{L_t,j})$$

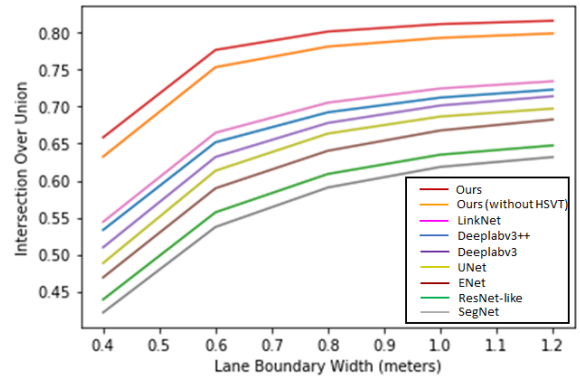
Once the marker type or road edge type class label is assigned to each interpolated point of a road segment (i.e., squiggle), the final step is to apply a smoothing function. The smoothing function makes sure the marking type is consistent throughout the road segment by using a median filtering method.

## V. Experimental Results

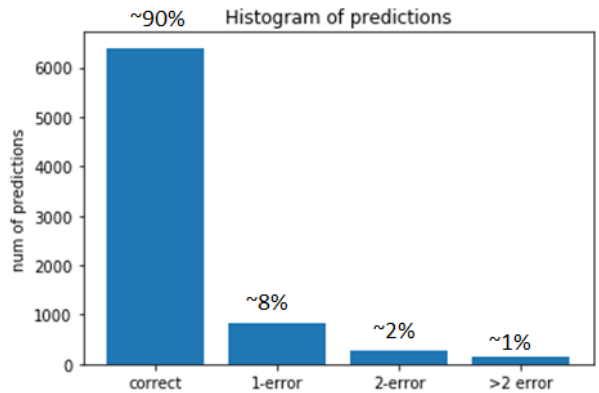
### A. Lane and Road Edge Detection

*1) Evaluating Inference Model:* In this section, we first evaluated the performance of our proposed multitask CNN and benchmarked its performance with several state-of-the-art segmentation networks: UNet [33], Deeplabv3++ [34], Deeplabv3 [35], LinkNet [13], SegNet [36], ResNet-variant, and ENet [14]. For evaluating the performance of the network, we used IoU: $\frac{\text{TP}}{\text{TP+FP+FN}}$ [37] as the metric across several acceptable thickness for lane and road edges. For this purpose, we created a dataset of 75K images randomly selected from a road network of a geofenced area of 96km2. We split the dataset as 70%, 20%, 10% for train, validation and test, respectively. We obtained the ground truth data for the created dataset through lidar scans, which is then transformed into a format that can be consumed by the network. The size of the images created for each selected node from the road network is $512 \times 512$ RGB image centered around the node as described in Sec. II. Each image spans approximately an area of 2500 $m^2$. For each created image, we also generated the corresponding 7 days of aggregated vehicle telemetry data and aligned the telemetry data with the image pixels. Note that the actual number of vehicles for a given week varies based on the geolocation of a given road segment. We have, however, experimentally observed that a week of data gives sufficient data for most of the urban and suburban areas.

For training the network, we used NVIDIA TESLA P100 GPUs to conduct our experiments. We also performed data augmentation, flipping the inputs vertically and horizontally as well as randomly dropping and adding noise up to 10% of the aggregated vehicle telemetry data for each batch. We set the learning rate to 0.0001 initially and gradually decay it as the learning saturates by using an Adam optimizer [38]. We set the batch size to 8 to trade-off memory capacity and stability in training. We trained the networks for 60 epochs to ensure convergence for all networks to have a fair comparison.




(a)




(b)

Fig. 8. a) Evaluation of various networks for detecting lane edges across different lane boundary widths, and b) evaluation of the number of lane predictions.

We trained our proposed network end-to-end and evaluated the performance of the branch for lane edges to make a fair comparison with other segmentation networks. In the training, we used unweighted average of the losses coming from the individual branches.

After training the networks, the performance is evaluated on the test set using IoU across several allowed thickness for lane and road edges ranging from 40cm to 120cm. For each acceptable lane boundary width, we calculated the IoU and reported it for all networks. Fig. 8 (a) depicts the performance for all networks across different acceptable lane boundary widths. As expected, when the acceptance interval for the lane edge boundary gets wider, the performance increases for all networks up to some value after which the performance gets saturated. We also note that using aggregated vehicle telemetry data as an additional data source in addition to aerial imagery improves the performance by approximately 5%.

For our trained model, we also evaluated the performance of the branch that predicts the number of lanes and plotted the histogram of errors as shown in Fig. 8 (b). Our network predicts 90% of the number of lanes correctly while predicting 8% with 1-off error and 2% with 2-off errors in lane counts. The prediction is off more than two lanes in less than 1% of the data.

*2) Evaluating End-to-End Lane and Road Edge Detection:* Next, we evaluated the performance of the end-to-end lane and road edge detection algorithm where the inference results

Authorized licensed use limited to: Queen's University. Downloaded on February 21,2022 at 05:45:38 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.



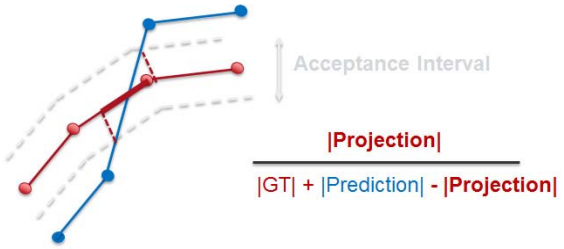
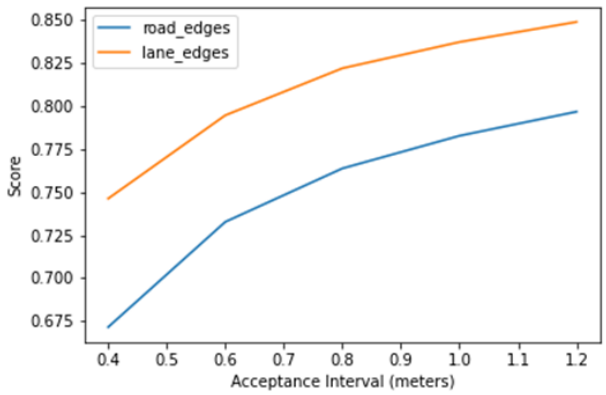
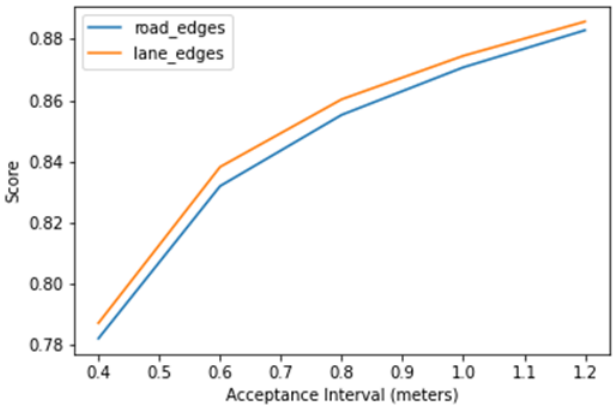



Fig. 9. The proposed metric to compare predicted and ground truth polylines given an acceptance interval around the ground truth polyline.




(a)




(b)

Fig. 10. Evaluation of end-to-end lane and road edge detection pipeline for a) marked roads, and b) unmarked roads.

along continuous road segments are clustered and stitched together as described in Sec. II. Since the output of this process is a polyline for each detected lane and road edge, we cannot directly use IoU as the metric. For this case, we propose a new metric to compare prediction and ground truth polylines as shown in Fig. 9. Given an acceptance interval around the ground truth polyline, the proposed metric captures the ratio of the segments of the predicted polyline in the acceptance interval projected onto the ground truth polyline and the total length of the predicted and ground truth polylines. Since the length of the projection is counted twice in the denominator, it needs to be subtracted from the total length. Hence, the score is calculated as: $\frac{|\text{Projection}|}{|\text{GT}|+|\text{Prediction}|-|\text{Projection}|}$, where the length of the projection serves as the correctly detected GT polyline segment. This ensures that the metric ranges between 0 and 1 and it penalizes both false positives and missed detections.

Fig. 11. Predicted and GT polylines with the calculated scores.

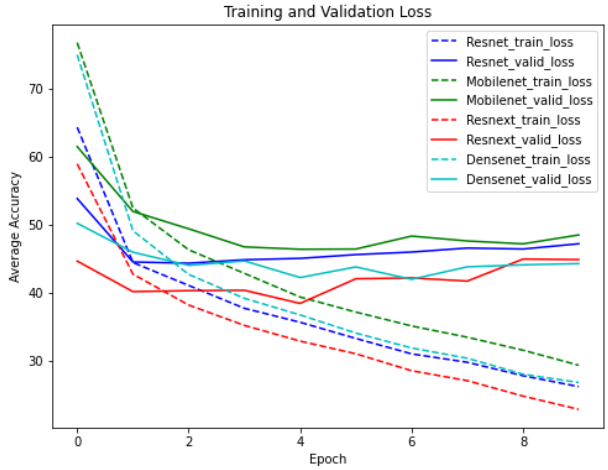



Fig. 12. Training and validation loss for Resnet, Resnext, Densenet, and Mobilenet.

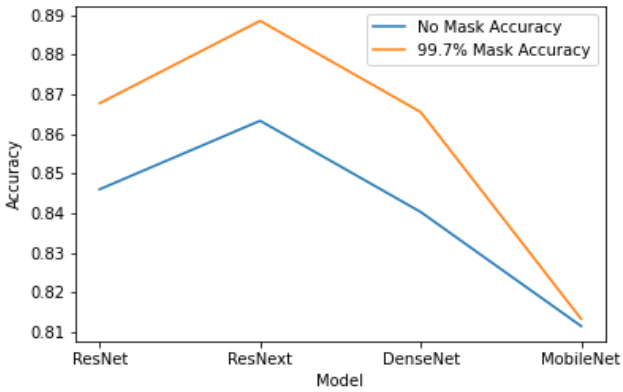



Fig. 13. Test accuracy after 5 epochs for Resnet, Resnext, Densenet, and Mobilenet.

Note that if the predicted polyline perfectly overlaps with the GT polyline, the metric is 1 and it is 0 when the predicted polyline is fully out of the acceptance interval.

We evaluated the performance of our end-to-end lane detection pipeline for both marked and unmarked roads. For this purpose, we create randomly selected 8,000 edges from the road network for both marked and unmarked roads. Using the inference model and stitching algorithms described in Sec. II, we constructed lane and road edges for the selected road segments. We then compared the predicted polylines and GT polylines using the metric shown in Fig. 9for both lane and road edges. We calculated the scores for various acceptance

Authorized licensed use limited to: Queen's University. Downloaded on February 21,2022 at 05:45:38 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.



TABLE I
CATEGORIZATION WITHOUT VEHICLE TELEMETRY MASK

| | | **Predicted:** unmarked residential | **Predicted:** two-way shared lane | **Predicted:** two-way double yellow | **Predicted:** Others |
|---|---|---|---|---|---|
| Resnet | **Actual:** unmarked residential | 0.91 | 0.00 | 0.01 | 0.08 |
| | **Actual:** two-way shared lane | 0.00 | 0.90 | 0.03 | 0.07 |
| | **Actual:** two-way double yellow | 0.02 | 0.03 | 0.74 | 0.21 |
| | **Actual:** Others | 0.10 | 0.02 | 0.05 | 0.83 |
| ResNext | **Actual:** unmarked residential | 0.92 | 0.00 | 0.01 | 0.07 |
| | **Actual:** two-way shared lane | 0.00 | 0.85 | 0.13 | 0.02 |
| | **Actual:** two-way double yellow | 0.03 | 0.00 | 0.83 | 0.12 |
| | **Actual:** Others | 0.08 | 0.02 | 0.09 | 0.81 |
| DenseNet | **Actual:** unmarked residential | 0.84 | 0.01 | 0.03 | 0.12 |
| | **Actual:** two-way shared lane | 0.00 | 0.89 | 0.08 | 0.03 |
| | **Actual:** two-way double yellow | 0.02 | 0.03 | 0.85 | 0.10 |
| | **Actual:** Others | 0.05 | 0.04 | 0.12 | 0.78 |
| MobileNet | **Actual:** unmarked residential | 0.94 | 0.00 | 0.01 | 0.05 |
| | **Actual:** two-way shared lane | 0.01 | 0.83 | 0.13 | 0.03 |
| | **Actual:** two-way double yellow | 0.05 | 0.05 | 0.81 | 0.09 |
| | **Actual:** Others | 0.14 | 0.04 | 0.15 | 0.67 |

intervals and plotted in Fig. 10 (a) and (b). As expected, the score improves as the acceptance interval gets wider and then saturates after some point for both lane and road edges. Note that the overall performance is slightly lower for marked roads since the variation of lane configurations is larger. Also note that the performance for road edges is lower than lane edges for both marked and unmarked roads due to higher variation in visibility and adjacent image details. Fig. 11 shows several examples for GT and predicted polylines along with their calculated scores.

Authorized licensed use limited to: Queen's University. Downloaded on February 21,2022 at 05:45:38 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.



TABLE II
CATEGORIZATION WITH VEHICLE TELEMETRY MASK

| | | **Predicted:** unmarked residential | **Predicted:** two-way shared lane | **Predicted:** two-way double yellow | **Predicted:** Others |
|---|---|---|---|---|---|
| Resnet | **Actual:** unmarked residential | 0.91 | 0.00 | 0.04 | 0.05 |
| | **Actual:** two-way shared lane | 0.00 | 0.93 | 0.05 | 0.02 |
| | **Actual:** two-way double yellow | 0.03 | 0.03 | 0.90 | 0.05 |
| | **Actual:** Others | 0.07 | 0.03 | 0.17 | 0.74 |
| ResNext | **Actual:** unmarked residential | 0.97 | 0.00 | 0.01 | 0.02 |
| | **Actual:** two-way shared lane | 0.01 | 0.89 | 0.07 | 0.04 |
| | **Actual:** two-way double yellow | 0.05 | 0.08 | 0.73 | 0.13 |
| | **Actual:** Others | 0.16 | 0.05 | 0.12 | 0.67 |
| DenseNet | **Actual:** unmarked residential | 0.94 | 0.00 | 0.01 | 0.05 |
| | **Actual:** two-way shared lane | 0.00 | 0.95 | 0.02 | 0.03 |
| | **Actual:** two-way double yellow | 0.02 | 0.07 | 0.82 | 0.08 |
| | **Actual:** Others | 0.10 | 0.04 | 0.11 | 0.75 |
| MobileNet | **Actual:** unmarked residential | 0.97 | 0.00 | 0.01 | 0.02 |
| | **Actual:** two-way shared lane | 0.01 | 0.89 | 0.07 | 0.04 |
| | **Predicted:** two-way double yellow | 0.05 | 0.08 | 0.73 | 0.13 |
| | **Predicted:** Others | 0.16 | 0.05 | 0.12 | 0.67 |

### *B. Road Categorization*

In our study we evaluate the performance of ResNet [29], DenseNet [40], ResNeXt [39], and MobileNet [41] on 9,940 images in four road types, i.e., 2260 images for each road category. The dataset is split at 50%-30%-20% for train-validation-test. The batch size is set as 64, and optimizer is Adam [38]. The learning rate is defined as 0.0001, and training epochs is set as 10. For training, validation, and test dataset, we resize the image to 299 $\times$ 299 for feeding into ResNeXt, and 224 $\times$ 224 for other three networks. We also

Authorized licensed use limited to: Queen's University. Downloaded on February 21,2022 at 05:45:38 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.



TABLE III
CONFUSION MATRIX FOR LMT CLASSIFICATION

| | | **Predicted:** dashed single line | **Predicted:** dashed solid and solid dashed line | **Predicted:** solid double line | **Predicted:** solid single line | **Predicted:** virtual inferred line |
|---|---|---|---|---|---|---|
| Resnet | **Actual:** dashed single line | 0.94 | 0.02 | 0.01 | 0.01 | 0.02 |
| | **Actual:** dashed solid and solid dashed line | 0.03 | 0.93 | 0.04 | 0.01 | 0.00 |
| | **Actual:** solid double line | 0.00 | 0.03 | 0.95 | 0.01 | 0.01 |
| | **Actual:** solid single line | 0.01 | 0.01 | 0.03 | 0.92 | 0.03 |
| | **Actual:** virtual inferred line | 0.02 | 0.01 | 0.03 | 0.05 | 0.90 |
| Incepsion-ResNet-v2 | **Actual:** dashed single line | 0.93 | 0.02 | 0.00 | 0.03 | 0.03 |
| | **Actual:** dashed solid and solid dashed line | 0.04 | 0.90 | 0.03 | 0.02 | 0.01 |
| | **Actual:** solid double line | 0.01 | 0.08 | 0.86 | 0.02 | 0.03 |
| | **Actual:** solid single line | 0.02 | 0.01 | 0.02 | 0.89 | 0.06 |
| | **Actual:** virtual inferred line | 0.04 | 0.01 | 0.01 | 0.04 | 0.90 |

standardize the image with mean and standard deviation for all RGB channels. We also applied data augmentation by flipping images vertically, horizontally, and randomly rotating each image less than 15 degree. We also randomly change the brightness, contrast and saturation of the training images (color jitter).

Transfer learning is used in ResNet, ResNeXt, DenseNet, and MobileNet to train for 10 epochs, and the result is displayed in Fig. 12. We found that after 5 epochs, the validation loss for all four models are not decreasing although the training loss is still decreasing. Therefore, the early stopping criteria is applied to avoid overfitting, and we stopped training the model after 5 epochs.

Then we evaluated the performance of ResNet, ResNeXt, DenseNet, and MobileNet on the test dataset without using the vehicle telemetry mask, and the accuracy is achieved as 84.6%, 84.0%, 85.1%, and 81.2%, respectively. The confusion matrix is shown in Table I. Similarly, we trained the images with using the vehicle telemetry mask for 5 epochs with the accuracy 86.8%, 85.1%, 86.5%, and 81.3%, respectively for ResNet, ResNeXt, DenseNet, and MobileNet, and the confusion matrix is displayed in Table II. As shown in the table, we find that consistent with Table I, the prediction of *unmarked residential* category and *two-way with a shared lane* category perform the best as these categories have the most robust features for the classification. Most of the false negative instances in predicting *unmarked residential* category fell in the *others* category, and most of the false negative instances in predicting the *two-way with a shared lane* class fell into the *two-way separated with a double yellow* class, which may not be surprising since both categories of road have doubled-stripe features in near the center of roadway (one double-solid versus solid-dash and dash-solid pair). The 3$^{rd}$-best predicted category is *two-way separated with a double yellow*. The *others* category performs the worst, with the false negative instances almost evenly distributed to *unmarked residential* and *two-way separated with a double yellow* categories, and having the most false positives of all categories.

Road categorization training and inference was run again with the road area masks derived from aggregated vehicle telemetry as described in Section 3 and compared to training and inferences made without the mask. The test accuracy for the four tested networks is displayed in Fig. 13, and indicates that three of the networks perform better with masked images than without by approximately 2%, the exception being MobileNet, perhaps due to its relatively simple structure compared to the other three models it does not have the capacity to use background information for prediction.

Furthermore, using the mask improves predicting *unmarked residential* and *two-way with a shared lane* categories significantly, for which background information appears to be a major error source, so removing them with the mask helps the network focus more on identifying distinct characteristics such as the lack of markings in the *unmarked residential* case

Authorized licensed use limited to: Queen's University. Downloaded on February 21,2022 at 05:45:38 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.



and the pair of doubled (solid+dash) markings in the *two-way with a shared lane* case. However, using a mask failed to improve the prediction of the *two-way separated with a double yellow* category, which indicates the background information does not provide any useful information for its classification, and the main characteristic is the road itself. Finally, using mask makes the prediction of the *others* category much worse, which implies that background information is very helpful as surrounding context to identifying it accurately.

### C. Lane Marking and Road Edge Type Prediction

In our study we evaluated the performance of ResNet [29] and Inception-ResNet-v2 [30] image classification models for the LMT/RET detection task. All these models are pretrained in the ImageNet [32] dataset, and transfer learning was used to perform the LMT/RET classification tasks. The training dataset was created from aerial images using the method described in Sec. IV. Dataset contains 100,000 images of 5 lane markers type, i.e., 20,000 images for each type of a), b), c), d) and e) in Fig. 14. The dataset is split at 70%-20%-10% for train-validation-test. The batch size is set as 32, and optimizer is Adam [38]. The learning rate is defined as 0.0001, and training epochs is set as 50.

For training, validation, and test dataset, we resize the image to $299 \times 299$ for feeding into both ResNet, and Incepsion-ResNet-v2 networks. We also normalize the image color intensities according to mean and standard deviation [(0.485, 0.229), (0.456, 0.224), (0.406, 0.225)] for each RGB channel respectively. To improve the performance of the network, data augmentation is performed on the training data. We flipped images vertically and horizontally, and randomly rotated each image less than 15 degree. Four TESLA P100 GPUS are used to conduct the experiment. Transfer learning was used in ResNet, and Inception-ResNet-v2 to train for 50 epochs. We found that after 50 epochs, the validation loss for both models are not decreasing although the training loss is still decreasing. Therefore, early stopping criteria to avoid over-fitting was met, and we stopped training the model after 50 epochs. Then we evaluated the performance of ResNet, and Inception-ResNet-v2 networks on test dataset, which achieved 94.6%, and 89.5% respectively. Based on the confusion matrices shown in Table III, we found that prediction of *solid single line* type, *dashed solid and solid dashed line* type, and *dashed single line type* all performed much better than the *solid double line* and *virtual inferred* types.

As can be seen in Fig. 14, both the *dashed solid and solid dashed line* type and *dashed single line* type have a dashed line characteristic, which seems to be a robust feature that is learnt well by both models evaluated and give better performance for these two types. The best type classification performance overall was for *solid double* type with ResNet, owing to this type's distinctive visual pattern, but it also had a few false negatives appearing as *dashed solid and solid dashed* type, perhaps due to some occlusion or blurriness. Similarly, *solid single line* type is also giving good performance but had many false negatives appearing as *solid double* type.

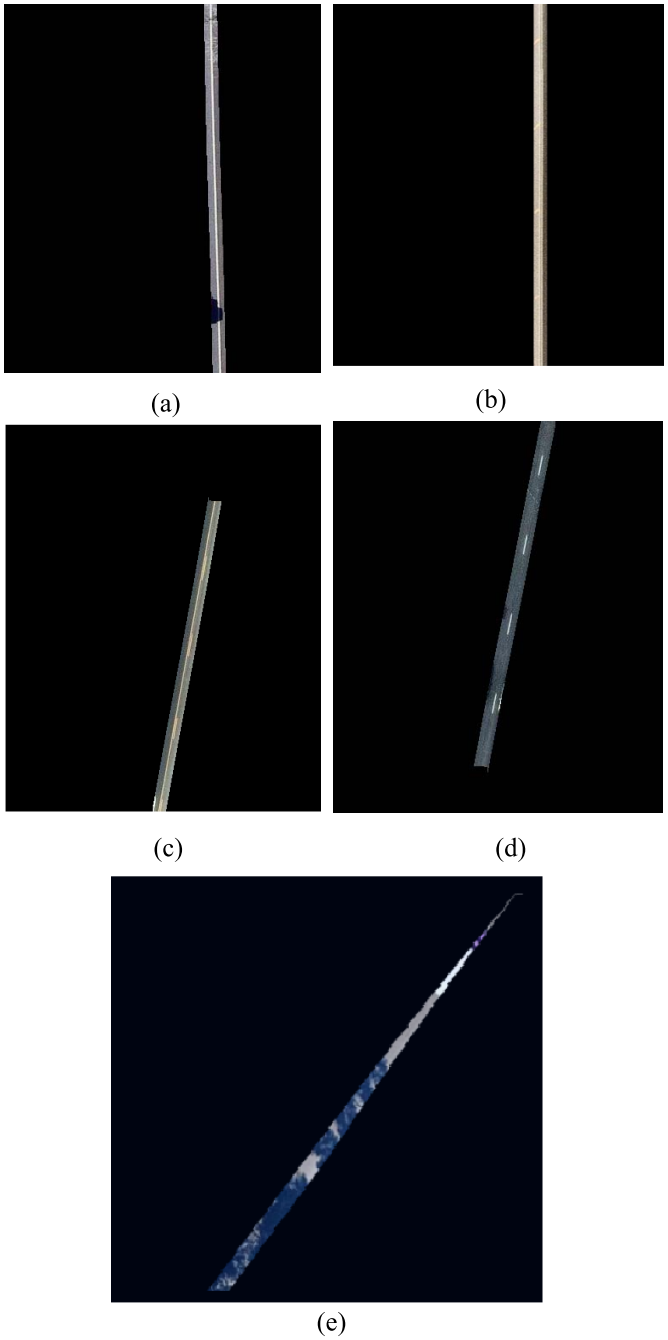



Fig. 14. (a) Solid single line, (b) solid double line, (c) dashed solid and solid dashed line, (d) dashed single line, and (e) virtual inferred line.

Furthermore, most false negative instances noted above are for *virtual inferred* type, which is used for segments of road where lanes of travel exist, but are not demarked by paint stripes or other markings. We can learn from the example in Fig. 15 (e), if the lane marking is not visible fully due to occlusion by trees, vehicles, or the marking color is faded, the lane will be inferred as *virtual inferred* type by the model. Subsequently, the networks performed worse at predicting the *virtual inferred* type than the physical marking types, generating more misclassified results in this case.

Similarly, we trained the ResNet classification model for RET detection task. Road edge dataset was created from aerial

Authorized licensed use limited to: Queen's University. Downloaded on February 21,2022 at 05:45:38 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.



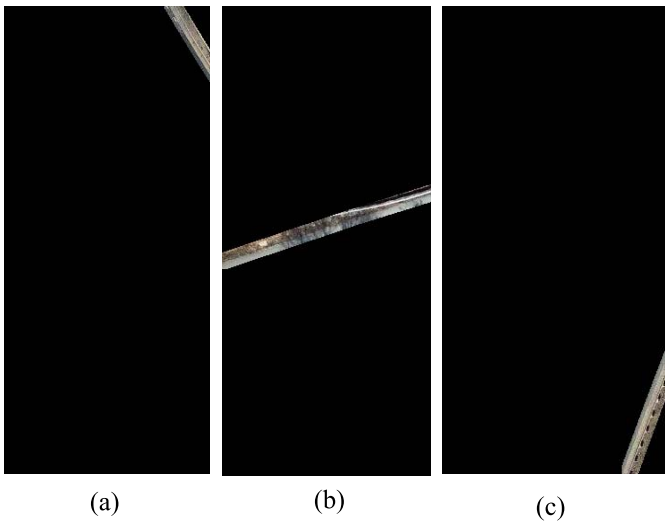



Fig. 15. (a) Curb, (b) paved to non-paved transition, and (c) barrier.

TABLE IV
NORMALIZED CONFUSION MATRIX FOR RET CLASSIFICATION

| | | **Predicted:** Barrier | **Predicted:** Curb | **Predicted:** Paved_to_Non-Paved |
|---|---|---|---|---|
| Resnet | **Actual:** Barrier | 0.89 | 0.08 | 0.03 |
| | **Actual:** Curb | 0.07 | 0.83 | 0.10 |
| | **Actual:** Paved_to_Non-Paved | 0.03 | 0.19 | 0.78 |

images using the same mask creation method described for lane marking types. The dataset contains 12,000 images of 3 road edge type, i.e., 4,000 images for each type of a), b), and c) in Fig. 15. The dataset is split at 70%-20%-10% for train-validation-test. Then we evaluated the performance on test dataset, and the accuracy was achieved as 84.6%. The confusion matrix shown in Table IV. Based on the confusion matrix, we found that prediction of *barrier* and *curb* road edge types performed much better than *paved to non-paved* edge type.

## VI. CONCLUSION

The new approach we present in this paper is an effective solution to create HD-maps at scale for autonomous vehicles, showing that the inherent scaling challenges that exist in ground survey-based HD-map creation can be overcame by utilizing three data sources: high resolution aerial imagery, aggregated vehicle telemetry, and navigation maps. While the first two provide rich information to detect road features, the latter enables us to guide the process along the road network. The proposed multitask CNN extracts multiple road features in a single inference and fusing aggregated vehicle telemetry along with aerial imagery for inference improves the detection performance in extracting road features. Categorizing roads based on lane configurations while using aggregated vehicle telemetry in an unsupervised fashion helps to detect some of the possible errors in automated map creation. Visual inspection of the detected features by our new approach and comparisons with data collected by ground survey vehicles show the effectiveness of the proposed approach and suggests a feasible approach to the largest scale mapping initiatives.

## REFERENCES

[1] Cadillac.com. Accessed: Jan. 22, 2021. [Online]. Available: https://www.cadillac.com/world-of-cadillac/innovation/super-cruise
[2] F. Poggenhans *et al.*, "Lanelet2: A high-definition map framework for the future of automated driving," in *Proc. 21st Int. Conf. Intell. Transp. Syst. (ITSC)*, Nov. 2018, pp. 1672–1679.
[3] J. Jiao, "Machine learning assisted high-definition map creation," in *Proc. IEEE 42nd Annu. Comput. Softw. Appl. Conf. (COMPSAC)*, Jul. 2018, pp. 367–373.
[4] R. V. Carneiro *et al.*, "Mapping road lanes using laser remission and deep neural networks," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2018, pp. 1–8.
[5] C. Guo, K. Kidono, J. Meguro, Y. Kojima, M. Ogawa, and T. Naito, "A low-cost solution for automatic lane-level map generation using conventional in-car sensors," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 8, pp. 2355–2366, Aug. 2016.
[6] G.-P. Gwon, W.-S. Hur, S.-W. Kim, and S.-W. Seo, "Generation of a precise and efficient lane-level road map for intelligent vehicle systems," *IEEE Trans. Veh. Technol.*, vol. 66, no. 6, pp. 4517–4533, Jun. 2017.
[7] P. Lamon *et al.*, "Mapping with an autonomous car," Tech. Rep., 2006.
[8] P. Fischer, S. M. Azimi, R. Roschlaub, and T. Krauß, "Towards HD maps from aerial imagery: Robust lane marking segmentation using country-scale imagery," *ISPRS Int. J. Geo-Inf.*, vol. 7, no. 12, p. 458, Nov. 2018.
[9] Accessed: Oct. 2020. [Online]. Available: https://global.toyota/en/newsroom/corporate/31898884.html
[10] Accessed: Oct. 2020. [Online]. Available: https://www.mapillary.com/
[11] M. Mark, *Rhumb Lines and Map Wars: A Social History of the Mercator Projection*. Chicago, IL, USA: University of Chicago Press, 2010.
[12] H. Zhang, J. Shao, and R. Salakhutdinov, "Deep neural networks with multi-branch architectures are intrinsically less non-convex," in *Proc. AISTATS*, 2019, pp. 1099–1109.
[13] A. Chaurasia and E. Culurciello, "LinkNet: Exploiting encoder representations for efficient semantic segmentation," in *Proc. IEEE Vis. Commun. Image Process. (VCIP)*, Dec. 2017, pp. 1–4.
[14] A. Paszke, A. Chaurasia, S. Kim, and E. Culurciello, "ENet: A deep neural network architecture for real-time semantic segmentation," 2016, *arXiv:1606.02147*.
[15] Y. Wei, H. Xiao, H. Shi, Z. Jie, J. Feng, and T. S. Huang, "Revisiting dilated convolution: A simple approach for weakly- and semi-supervised semantic segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 7268–7277.
[16] L. Fidon *et al.*, "Generalised wasserstein dice score for imbalanced multi-class segmentation using holistic convolutional networks," 2017, *arXiv:1707.00478*.
[17] V. Belagiannis, C. Rupprecht, G. Carneiro, and N. Navab, "Robust optimization for deep regression," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 2830–2838.
[18] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1137–1149, Jun. 2017.
[19] W. Van Gansbeke, B. De Brabandere, D. Neven, M. Proesmans, and L. Van Gool, "End-to-end lane detection through differentiable least-squares fitting," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. Workshop (ICCVW)*, Oct. 2019, pp. 1–9.
[20] M. Ester, H. P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *Proc. KDD*. vol. 96, no. 34, 1996, pp. 1–6.
[21] C. De Boor and J. R. Rice, "Least squares cubic spline approximation I-fixed knots," Tech. Rep., 1968.
[22] C. Henry, S. M. Azimi, and N. Merkle, "Road segmentation in SAR satellite images with deep fully convolutional neural networks," *IEEE Geosci. Remote Sens. Lett.*, vol. 15, no. 12, pp. 1867–1871, Dec. 2018.
[23] A. Ferdowsi, U. Challita, and W. Saad, "Deep learning for reliable mobile edge analytics in intelligent transportation systems: An overview," *IEEE Veh. Technol. Mag.*, vol. 14, no. 1, pp. 62–70, Mar. 2019.

Authorized licensed use limited to: Queen's University. Downloaded on February 21,2022 at 05:45:38 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.




[24] W. Shi, S. Shen, and Y. Liu, "Automatic generation of road network map from massive GPS, vehicle trajectories," in *Proc. 12th Int. IEEE Conf. Intell. Transp. Syst.*, Oct. 2009, pp. 1–6.

[25] E. R. I. A. C. M. Uduwaragoda, A. S. Perera, and S. A. D. Dias, "Generating lane level road data from vehicle trajectories using kernel density estimation," in *Proc. 16th Int. IEEE Conf. Intell. Transp. Syst. (ITSC)*, Oct. 2013.

[26] L. Zheng, B. Li, B. Yang, H. Song, and Z. Lu, "Lane-level road network generation techniques for Lane-level maps of autonomous vehicles: A survey," *Sustainability*, vol. 11, no. 16, p. 4511, Aug. 2019.

[27] L. Tang, X. Yang, Z. Dong, and Q. Li, "CLRIC: Collecting lane-based road information via crowdsourcing," *IEEE Trans. Intell. Transport. Syst.*, vol. 17, no. 9, pp. 2552–2562, Sep. 2016.

[28] C. Cortes and V. Vapnik, "Support-vector networks," *Mach. Learn.*, vol. 20, no. 3, pp. 273–297, Sep. 1997.

[29] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

[30] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. Alemi, "Inception-v4, inception-ResNet and the impact of residual connections on learning," 2016, *arXiv:1602.07261*.

[31] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 2818–2826.

[32] Accessed: Jan. 22, 2021. Image-net.org. [Online]. Available: https://www.image-net.org/

[33] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Medical Image Computing and Computer-Assisted Intervention* (Lecture Notes in Computer Science). Cham, Switzerland: Springer, 2015, pp. 234–241.

[34] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, "Encoder-decoder with atrous separable convolution for semantic image segmentation," in *Computer Vision*. Cham, Switzerland: Springer, 2018, pp. 833–851.

[35] L.-C. Chen, G. Papandreou, F. Schroff, and H. Adam, "Rethinking atrous convolution for semantic image segmentation," 2017, *arXiv:1706.05587*.

[36] V. Badrinarayanan, A. Kendall, and R. Cipolla, "SegNet: A deep convolutional encoder-decoder architecture for image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 12, pp. 2481–2495, Dec. 2015.

[37] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The PASCAL visual object classes (VOC) challenge," *Int. J. Comput. Vis.*, vol. 88, no. 2, pp. 303–338, Jun. 2010.

[38] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*.

[39] S. Xie, R. Girshick, P. Dollar, Z. Tu, and K. He, "Aggregated residual transformations for deep neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 1492–1500.

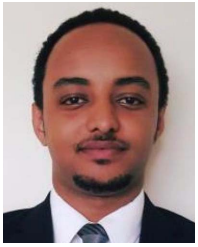
[40] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 4700–4708.

[41] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "MobileNetV2: Inverted residuals and linear bottlenecks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 4510–4520.

**Yijun Wei** (Member, IEEE) received the B.S. degree in mathematics from Sichuan University, Sichuan, China, in 2011, the M.S. degree in mathematics and applied statistics from the University of Michigan, Ann Arbor, MI, USA, in 2013, and the Ph.D. degree in computational science informatics from George Mason University, Fairfax, VA, USA, in 2020. He is currently an Artificial Intelligence/Machine Learning Scientist at General Motors, developing the ML/DL algorithms for HD-map creation for AV programs. Prior to joining General Motors, he was a Research Statistician at the National Institute of Statistical Sciences, Washington, DC, USA. His research interest includes the machine learning/deep learning.

**Faria Mahnaz** received the B.S. and M.S. degrees in mathematics from the University of Dhaka, Bangladesh, in 2010 and 2012, respectively, and the M.S. degree in computer science from Wayne State University, MI, USA, in 2015, where she is currently pursuing the Ph.D. degree in industrial and systems engineering. She joined General Motors as an Artificial Intelligence/Machine Learning Scientist in 2019. She is involved in developing machine learning algorithms for HD mapping, and currently supporting AV programs. Prior to joining General Motors, she was a Data Scientist at Ford Motor Company, MI, USA, supporting company's Global Data Insight and Analytics Team. She has been a member of several research efforts dealing with machine learning throughout her graduate programs and professional career. Her research interests include deep learning, sparse representation learning, and computer vision.

**Orhan Bulan** (Senior Member, IEEE) received the B.S. degree in electrical and electronics engineering from Bilkent University, Ankara, Turkey, in 2006, and the M.S. and Ph.D. degrees in electrical and computer engineering from the University of Rochester, NY, USA, in 2007 and 2012, respectively. He is currently a Principal Artificial Intelligence Computer Scientist at General Motors, leading the ML efforts for HD-map creation for AV programs. Prior to joining General Motors, he was a Research Scientist at the Palo Alto Research Center, a xerox company, in Webster City, NY, USA. He holds over 60 patents and has coauthored over 35 papers in peer-reviewed journals and conference proceedings. He is a frequent reviewer of IEEE TRANSACTIONS ON IMAGE PROCESSING, IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS, IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY, and IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY.



**Yehenew Mengistu** received the B.S. degree in electrical and computer engineering from Bahir Dar University, Bahir Dar, Ethiopia, and the M.S. degree in electrical and computer engineering from Oklahoma State University. His graduate research areas include adaptation of machine learning and computer vision algorithms for wearable and other embedded applications. He currently works as an AI/ML Scientist at General Motors, developing machine learning algorithms for HD mapping, and supporting the company's autonomous software development.

**Sheetal Mahesh** was born in Bengaluru, India. She received the B.E. degree in computer science and engineering from Rashtreeya Vidyalaya (RV) College of Engineering, Bengaluru, in 2014, and the master's degree in science in data science from the University of Southern California, CA, USA, in 2016. She joined General Motors as an Associate AI/ML Scientist in 2017. She works with the Machine Learning Team, primarily focused on building the mapping systems to help in the navigation of AVs.

**Michael A. Losh** (Member, IEEE) received the B.A. degree in physics from Grinnell College and the M.S. degree in electrical and computer engineering from Oakland University. He is a Staff Researcher with General Motors Research and Development Electrical Controls Systems Laboratory, in the Perception, Planning, and Decision Systems Laboratory Group. Through much of his career, he has worked in automotive embedded control software development, including development for several of GM's autonomous concept vehicles, AV research projects, early production AV programs, and a recent focus on mapping and map use. He is a member of IEEE Vehicular Technology Society (VTS). He is currently serving as a reviewer for *ASME Journal of Autonomous Vehicles and Systems*.

Authorized licensed use limited to: Queen's University. Downloaded on February 21,2022 at 05:45:38 UTC from IEEE Xplore. Restrictions apply.