

# Diff-Net: Image Feature Difference based High-Definition Map Change Detection for Autonomous Driving

Lei He<sup>1</sup>, Shengjie Jiang<sup>2</sup>, Xiaoqing Liang<sup>2</sup>, Ning Wang<sup>2</sup> and Shiyu Song<sup>1\*</sup>

**Abstract**—Up-to-date High-Definition (HD) maps are essential for self-driving cars. To achieve constantly updated HD maps, we present a deep neural network (DNN), Diff-Net, to detect changes in them. Compared to traditional methods based on object detectors, the essential design in our work is a parallel feature difference calculation structure that infers map changes by comparing features extracted from the camera and rasterized images. To generate these rasterized images, we project map elements onto images in the camera view, yielding meaningful map representations that can be consumed by a DNN accordingly. As we formulate the change detection task as an object detection problem, we leverage the anchor-based structure that predicts bounding boxes with different change status categories. To the best of our knowledge, the proposed method is the first end-to-end network that tackles the high-definition map change detection task, yielding a single stage solution. Furthermore, rather than relying on single frame input, we introduce a spatio-temporal fusion module that fuses features from history frames into the current, thus improving the overall performance. Finally, we comprehensively validate our method’s effectiveness using freshly collected datasets. Results demonstrate that our Diff-Net achieves better performance than the baseline methods and is ready to be integrated into a map production pipeline maintaining an up-to-date HD map.

## I. INTRODUCTION

Autonomous driving has moved from the realm of science fiction to a practical possibility during the past twenty years. Despite many years of research and development, building reliable solutions that can handle the complexity of the real world remains an open problem. Modern self-driving vehicles primarily rely on detailed pre-built maps, the high-definition (HD) map, which typically contains rich information of the environment, such as topology and location of lanes, crosswalks, traffic lights, and intersections.

They are a great source of prior knowledge and must be maintained adequately by constantly updating them to reflect up-to-date changes in the real world. These changes typically include recently installed or removed traffic signals, portable traffic signals that just moved to new positions, or human errors during map production. In this work, we address the HD map update problem by building an end-to-end learning-based network that detects changes in the HD map, helping our self-driving vehicles access the latest environmental information.

As we aim to detect changes in an HD map, a conventional approach is to leverage object detection algorithms together

with necessary association and difference calculation algorithms. In this way, it derives map element changes given existing map elements and object detection results from on-line camera imagery. The entire processing pipeline involves multiple steps, such as object detection, element association, and difference calculation. However, the apparent problem is that each step above has its optimization objective, making the entire change detection pipeline fail to achieve an overall optimal solution. For example, an object detector typically involves thresholding detection confidence scores and running Non-Maximum Suppression (NMS) to trade-off precision against recall. Falsely detected objects in the early step will quickly lead to change detection failures eventually. Meanwhile, the traditional approach ignores important prior information from the HD map.

In this work, we propose an end-to-end learning-based method to detect map changes directly. More specifically, we use a deep neural network (DNN) to detect missing or redundant elements in an HD map. To incorporate the prior information in HD maps, we project map elements onto images and rasterize them from the camera’s perspective. Both the rasterized and online camera images are furnished into the DNN as input. We then calculate the differences between extracted features from both sources in different scales. These feature differences are propagated, fused, and decoded, finally yielding an end-to-end HD map change detection (HMCD) network. Moreover, since the changes are consistent in a group of consecutive frames over time, it is worth mentioning that we introduce a spatio-temporal feature fusion module to improve its performance further. To fully validate the designs in our proposed method, we construct a large-scale dataset that includes abundant synthesized and real HD map change cases. The synthesized cases help us overcome the vital issue that HD map changes are low probability events in practice, thus allowing us to accomplish network training and performance evaluation.

## II. RELATED WORK

Although HD maps have become an indispensable module in an autonomous driving system in recent years, relatively few attempts specifically focus on the HD map change detection (HMCD) problem in the academic community. Pannen et al. [1] propose a crowd-based method that combines particle filter and boosted classifier to infer the probability of HD map changes. Heo et al. [2] adopt an encoder-decoder architecture driven by adversarial learning, achieving a pixel-level HD map change detector in the camera view.

<sup>1</sup>The authors are with Baidu Autonomous Driving Technology Department, {helei07, songshiyu}@baidu.com.

<sup>2</sup>Work done during internship at Baidu, 2104190011@cugb.edu.cn, xiaoqing.liang@nlpr.ia.ac.cn, 20181222016@nuist.edu.cn.

\*Author to whom correspondence should be addressed, E-mail: songshiyu@baidu.com.

The most related task probably is the scene change detection [3], [4], [5], [6], [7], [8], of which the solutions can be mainly divided into three categories. The first category leverages 3D-to-3D comparisons [9] between a pre-built 3D CAD model and a reconstructed one built by classic multi-view stereo (MVS) methods [10], [11], known to be highly time-consuming methods and only applicable for offline applications. The second approach [12], [13], [14], [15], [7] is to infer the changes of the scene by comparing newly acquired images against the original 3D model. In particular, the probability of changes can be estimated by comparing the voxel color of a 3D voxel-based model against the color of the corresponding image pixels [12], [13]. A relevant alternative [14], [15], [7] is to identify changes by re-projecting a new image onto an old one with the help of the given 3D model and compare the inconsistencies. The third category [3], [16], [17], [5], [18], [19], [20], [21], [8] adopts 2D-to-2D comparisons between images representing old states and current states of a scene. A prior 2D-to-2D image registration step is required.

Besides detecting changes in a scene, our HD map change detection task identifies changed elements in the HD map together with the types of changes. A straightforward method is to recognize map elements in images using a standard object detector, project map elements onto the images, associate the projections with the detections, and finally obtain the corresponding changes through a cross-comparison procedure. Object detection is a classic problem in computer vision. The solutions can be mainly divided into two categories, namely two-stage [22], [23], [24] and one-stage [25], [26], [27] methods.

This work introduces an image feature difference-based HD map change detection method that infers the map changes by adopting the anchor-based one-stage detection method, YOLOv3 [25], as its detection head.

### III. PROBLEM FORMULATION

The HD map change detection (HMCD) task is formulated similar to an object detection problem. The goal is to recognize map change instances of a predefined set of object classes (e.g., traffic lights, signs), describe the locations of detected objects in the image using 2D bounding boxes, and assign correct change categories for them, including *to\_add*, *to\_del*, and *correct*. As their names imply, objects with *to\_add* attributes are the ones missed, *to\_del* are the ones that should be removed, and *correct* are the correct ones in the HD map, respectively. Portable traffic signals are special cases as they are treated as a pair of *to\_del* and *to\_add* bounding boxes illustrating both the old and new locations of the traffic signals. In particular, we focus on the change detection task of traffic signals in this work. Note that our proposed method can be extended to other objects in regular shapes, while the change detection problem of irregularly shaped ones is beyond the scope of this work.

Formally, for an online HMCD method that works with a single image as input, the problem can be formulated as:

$$\mathbf{D}_k = f_\theta(\mathbf{M}, I_k, T_k, K), \quad (1)$$

where  $\mathbf{M}$  is the HD map,  $I_k$  is the  $k$ -th image frame in a video stream,  $T_k$  is a global camera pose typically estimated by a localization system in a self-driving car,  $K$  is the camera intrinsic calibration matrix, and  $\mathbf{D}_k$  is a set of 2D bounding boxes with corresponding change categories predicted by our HMCD predictor  $f_\theta$  with a set of learnable parameters  $\theta$ .

### IV. METHOD

The overall architecture of the proposed Diff-Net is shown in Figure 1. Besides using the original camera image as our model input, we first construct a rasterized image by projecting map elements onto it from the camera perspective. Then, pyramid features in different resolutions are extracted by two parallel CNN-based backbones from both the rasterized and camera images. The key design of our work is to infer map changes by having a series of feature operators that propagate, fuse, differentiate, and decode these features. Meanwhile, we leverage anchor-based object detection techniques, finally inferring map changes from decoded features. The following sections describe them in detail.

#### A. Model Input

The fact that HD maps and camera images are data in different modalities poses considerable challenges in building a meaningful map data representation that a deep neural network can consume as an input. Inspired by recent planning [28], [29], [30], prediction [31], [32], [33] or change detection [2] works, we propose to construct an image from the camera perspective and rasterize it by projecting map elements onto it. Specifically, given a global pose (position and orientation) of the camera, we first query map elements within a region of interest (ROI) in the HD map. Then, they are projected onto the canvas from the camera perspective, with the projected area filled with a homochromatic color for the same type of objects in the HD map. This yields a binary image, if we only focus on a single object category, such as traffic lights, as shown in an example in the upper left corner of Figure 1. Furthermore, it is worth noting that our proposed method can be generalized to multi-category objects by rendering them with different colors. And it has been found that different color selections do not produce a significant effect when images are rasterized similarly in other tasks [33].

#### B. Feature encoding and Difference

Given both the rasterized images embedded with local map information and online camera images, we aim to extract meaningful features from them, yielding two parallel feature extraction pipelines in our implementation. They are shown in pink and green colors in Figure 1. We adopt a 11-layer CNN to increase its receptive field for the rasterized images, where the convolution strides of the 3rd, 6th, and 8-11th layers are 2, and others are 1. The size of the convolution kernel is  $3 \times 3$ , and the number of channels is shown in Figure 1. The ReLU activation functions are used to average input channels in our implementation. It is verified to be sufficient for feature extraction from these relatively clean

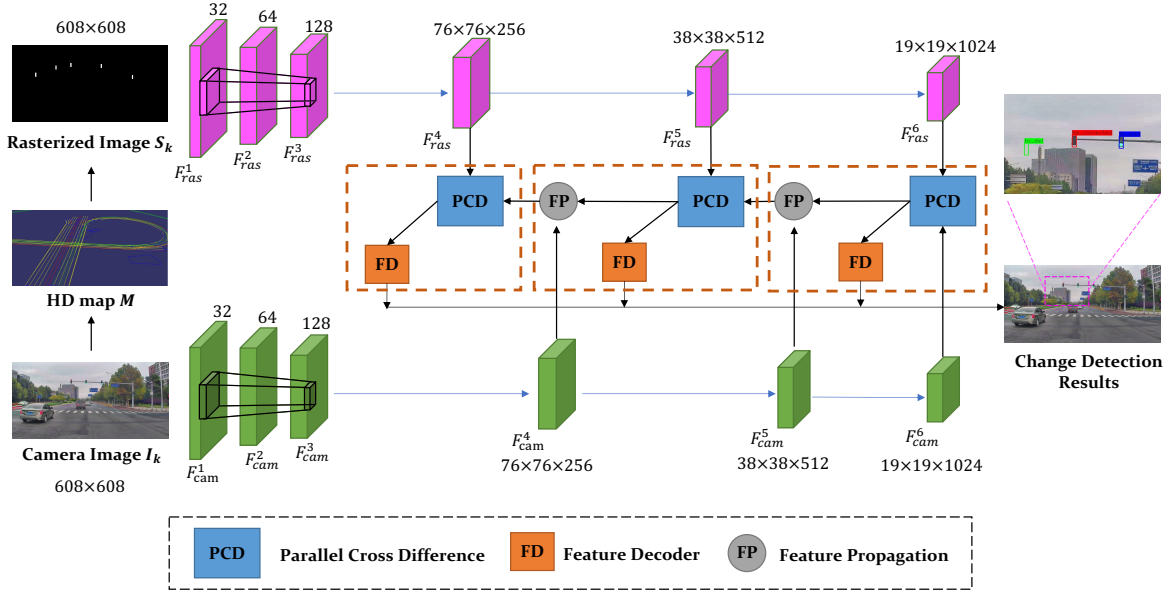


Fig. 1: The overall architecture of the Diff-Net involving three main modules: (a) rasterized images including projected map elements used as one of the inputs to CNN; (b) CNN based pyramid feature extraction layers for two parallel model inputs (pink and green); (c) a series of feature operators that involve feature propagation (FP), parallel cross difference (PCD) calculation, and feature decoding (FD). We downsample the original images from  $1920 \times 1080$  to  $608 \times 608$  as the network input.  $I_k$  is the  $k$ -th camera image.  $M$  is the HD map.  $S_k$  is the rasterized image embedded with map labels.  $F_{ras}^i$  and  $F_{cam}^i$  denote the extracted features from rasterized and camera images at  $i$ -th scale level, respectively. In addition, 32, 64, and 128 are the numbers of feature channels in the first, second, and third scale levels. The blue, green, and red masks in the top-right image represent the ground truth detection results in *correct*, *to\_del*, and *to\_add* change categories, respectively.

images. For feature extraction of online camera images, we use DarkNet-53 from YOLOv3 [25] because of its well-balanced accuracy and inference speed.

As we mentioned earlier, a conventional approach directly cross-compares object detection results against existing map elements to infer possible changes. Note that it is not a trivial problem since missing or redundant elements and localization noises make them not necessarily a group of one-to-one comparisons in most cases. Inspired by this process, we employ a deep neural network (DNN) that transforms comparisons in instance space to feature space, denoted as the parallel cross difference (PCD) network, as shown in Figure 2. Specifically, the PCD module calculates the difference between the two extracted features. Its output features pass through 4 convolution layers and are then processed by a feature decoder (FD), finally producing change detection results. This design leverages deep learning networks' powerful capabilities that they can generalize well in solving complicated problems. Our experiments also demonstrate that the proposed method achieves better performance, as shown in Section V.

Similar to YOLOv3 [25], we also predict bounding boxes at three different scales. As shown in Figures 1 and 2, features outputted by the PCD module at a coarser scale go through a feature propagation (FP) module. They are upsampled to a finer scale and then concatenated with camera features in the finer scale. After another convolution, the

resulted features are passed to the PCD module at a finer scale.

### C. Anchor-based Detection

As mentioned earlier, the output features of the PCD module are processed by a feature decoder (FD) module, which produces final detection bounding boxes. We first perform a  $3 \times 3$  convolution in the FD module to lift the feature channel dimension from  $c/2$  to  $c$ . Then, a  $1 \times 1$  convolution is applied to generate the region proposals, resulting in the final tensor with a channel number  $S \times S \times [3 \times (num\_class + 5)]$ , where  $num\_class$  represents the number of the change categories (3 for *to\_add*, *to\_del* and *correct*), 5 represents the location and confidence of the bounding box, and 3 denotes the number of anchor boxes in one of  $S \times S$  grid cells ( $S = 7$ ).

Similar to YOLOv3 [25], we have two branches for change detection. One outputs the change categories with softmax operations. The other infers elements' geometric locations  $t_x$ ,  $t_y$ ,  $t_w$ , and  $t_h$  with respect to necessary width and height priors  $p_w$  and  $p_h$  (See [25] for details). Finally, the non-maximum suppression (NMS) method from [34] is used to eliminate redundant detections.

### D. Spatio-temporal Fusion

Essentially, similar to object detection in autonomous driving applications, the data is acquired as a video stream instead of sparse images, and detection results in the map change detection task are temporally correlated. Therefore,



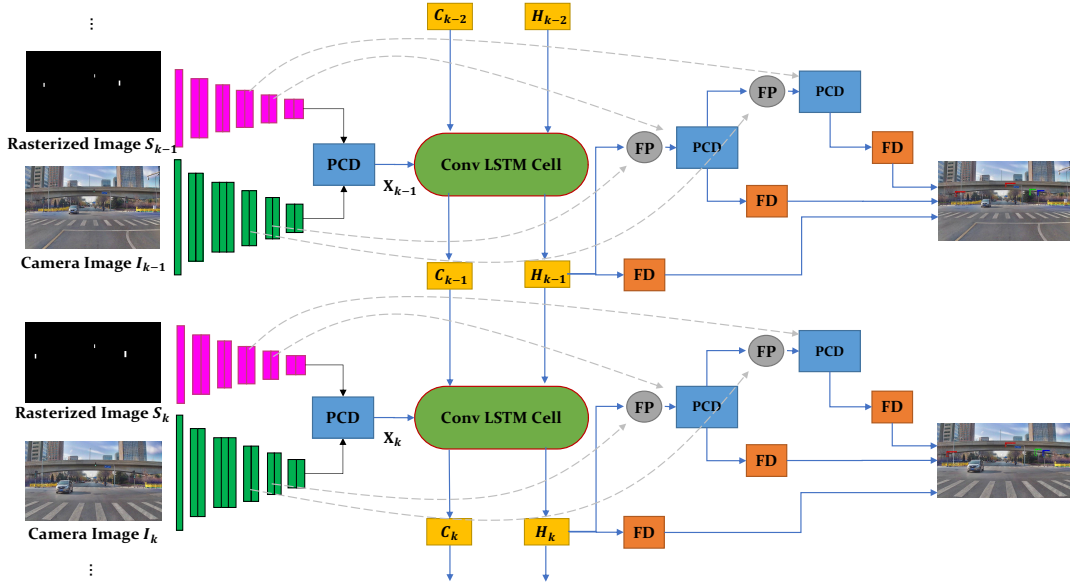


Fig. 3: The illustration of the network architecture with the ConvLSTM structure in the coarsest image resolution scale, which enables the temporal feature fusion between consecutive frames and improves the overall change detection performance.  $C_k$  and  $H_k$  denote the hidden state and the cell state in the ConvLSTM, respectively.

detection score of the  $c$ -th category in the  $i$ -th grid cell, and  $\mathbb{1}_i^{obj}$  denotes if the object appears in the  $i$ -th grid cell.

## V. EXPERIMENTS

### A. Datasets

To the best of our knowledge, there are no public datasets that focus on the HD map change detection task. Therefore, we recruited our self-driving vehicles equipped with standard sensors, such as LiDARs, cameras, IMUs, and GNSS receivers. We divide our data into three datasets, SICD, VSCD, and R-VSCD. In SICD and VSCD datasets, we collected data in Yizhuang District, Beijing city, the capital city of one of the largest developing countries, where environmental changes, for example, road constructions, are common. To fully validate different methods, we propose synthesizing change events as environmental changes are rare. Furthermore, to meet different needs, the SICD dataset contains 205,076 isolated images while the VSCD contains 3,750 short video clips. The R-VSCD dataset includes 44 video clips where actual traffic signal changes happened. Furthermore, it is worth mentioning that the R-VSCD dataset includes data from four different cities in China, Beijing, Cangzhou, Guangzhou, and Shanghai.

### B. Performance Analysis of SICD/VSCD Dataset

As we formulate our problem as a detection task, we adopt the mAP as our primary performance metric. YOLOv3 [25] + D is the conventional method that depends on a standard object detector together with necessary association and difference calculation algorithms. When we calculate the mAP for *to.del* cases, YOLOv3 typically does not output bounding boxes in the area because there is no object. Under such circumstances, we consider bounding boxes before the NMS

| Method            | mAP          |              |
|-------------------|--------------|--------------|
|                   | SICD         | VSCD         |
| Yolov3 [25]+D     | 0.437        | 0.423        |
| Diff-Net          | <b>0.876</b> | 0.678        |
| Diff-Net+ConvLSTM | -            | <b>0.761</b> |

TABLE I: Comparison of the change detection performance using the mAP metric. Our wide improvement over the conventional method, Yolov3 [25] + D, is notable. Meanwhile, the benefits of the ConvLSTM based spatio-temporal fusion module are visible.

| Method            | Top-1 Accuracy |
|-------------------|----------------|
| Yolov3 [25]+D     | 0.558          |
| Diff-Net          | 0.725          |
| Diff-Net+ConvLSTM | <b>0.810</b>   |

TABLE II: Comparison of the change detection performance using the dataset with real HD map changes. The problem is formulated as a classification problem to determine the correct change category. Our proposed method outperforms the baseline method.

step as candidates and take  $1.0 - c$  as the confidence score, where  $c$  is the original confidence score in YOLOv3. Diff-Net+ConvLSTM is our proposed method with the spatio-temporal fusion module. In Table I, we give a quantitative analysis of each method. Note our vast performance improvement over the conventional method in both the SICD and VSCD datasets. The end-to-end learning-based network achieves joint optimization of the change detection task, yielding significantly better performance overall. In terms of video data, the ConvLSTM aided version performs even better and achieves 76.1% mAP.



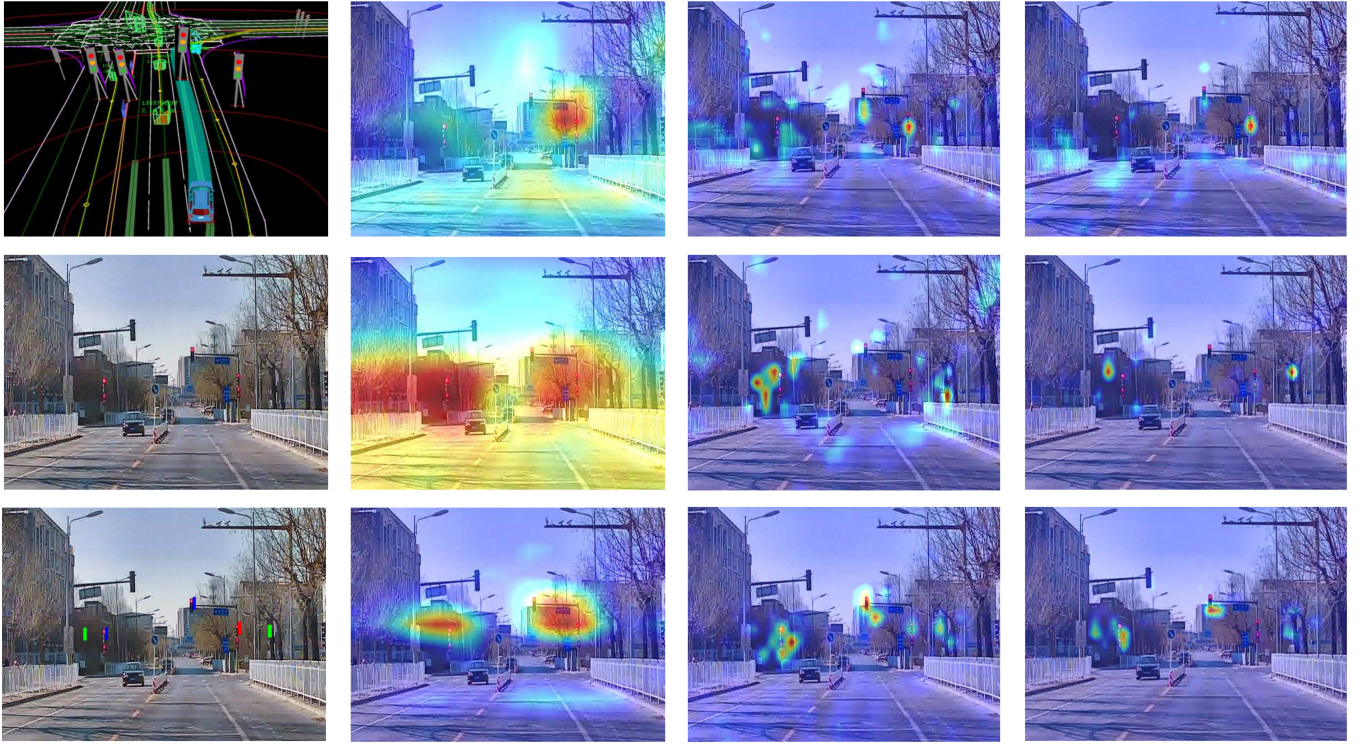


Fig. 4: Visualization of three channels (from top to bottom) of the PCD module’s output features. In the left-most column, from top to bottom, there are the HD map, a camera image, and a camera image with the ground truth of the change detection results. The blue, green, and red masks represent the *correct*, *to\_del*, and *to\_add* change categories, respectively. For the other three columns, the features are visualized as heatmaps. From left to right, we show the features in different image scales in a coarse-to-fine pattern. Note that the features accurately spread in areas of interest in the images.

### C. Performance Analysis of R-VSCD Dataset

As mentioned before, we introduced an R-VSCD dataset where actual traffic signal changes happened. In this experiment, we evaluate the performance of the proposed methods in detecting HD map changes in the real world. Since the number of changes in the R-VSCD dataset is too limited (HD map changes are rare) to produce a meaningful mAP value, we choose to evaluate the top-1 accuracy in this experiment. It is known that there is zero or one change case in each video clip, so the problem becomes a classification problem to determine the correct change category of the video clip, *correct*, *to\_add*, or *to\_del*. More specifically, we run our change detector for all frames in the video clip and determine the outcome using majority voting. The top-1 accuracy results of each method are reported in Table II. The ConvLSTM aided version achieves 81.0% top-1 accuracy and outperforms the baseline method. It is worth mentioning that the R-VSCD includes data from 4 different cities. However, our training data was collected in Beijing, China only. It makes the problem sufficiently challenging since traffic signals look markedly different from city to city.

### D. Feature Visualization

To help us interpret the effectiveness of the PCD module, we visualize three channels (from top to bottom) of the PCD’s final output features  $F_{pcd}^s$  in Figure 4. Note that the

features accurately cover the areas of interest in the images. Notably, no objects exist in the camera image for the *to\_del* changes. This implies that they are compelling features for the HD map change detection task. Also, interestingly, we find that features in a coarser scale focus more on larger objects, while features in a more refined scale are for smaller ones. This strictly follows the purpose of our design.

## VI. CONCLUSION

We have proposed a learning-based HD map change detection network designed for autonomous driving applications. Instead of resolving the problem in several isolated steps, contrary to conventional methods, the proposed method constructs an end-to-end network that infers the map changes directly in a joint framework, yielding substantially more outstanding performance. The excellent performance makes our method ready to be integrated into an HD map updating pipeline and support the regular operation of a self-driving fleet. Three datasets have been introduced in this work to fully validate our method, including synthetic and natural HD map changes. The datasets are to be published along with the future release of the Apollo (<https://apollo.auto/developer.html>). Our future work includes expanding our methods for more map elements and handling irregularly shaped objects.

## REFERENCES

- [1] D. Pannen, M. Liebner, and W. Burgard, "Hd map change detection with a boosted particle filter," in *Proceedings of the International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 2561–2567.
- [2] M. Heo, J. Kim, and S. Kim, "Hd map change detection with cross-domain deep metric learning," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2020, pp. 10 218–10 224.
- [3] R. J. Radke, S. Andra, O. Al-Kofahi, and B. Roysam, "Image change detection algorithms: A systematic survey," *IEEE Transactions on Image Processing (TIP)*, vol. 14, no. 3, pp. 294–307, 2005.
- [4] A. Taneja, L. Ballan, and M. Pollefeys, "Geometric change detection in urban environments using images," *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, vol. 37, no. 11, pp. 2193–2206, 2015.
- [5] K. Sakurada and T. Okatani, "Change detection from a street image pair using cnn features and superpixel segmentation," in *Proceedings of the British Machine Vision Conference (BMVC)*, vol. 61, 2015, pp. 1–12.
- [6] R. Qin, J. Tian, and P. Reinartz, "3D change detection – approaches and applications," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 122, pp. 41–56, 2016.
- [7] E. Palazzolo and C. Stachniss, "Fast image-based geometric change detection given a 3D model," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 6308–6315.
- [8] Y. Lei, D. Peng, P. Zhang, Q. Ke, and H. Li, "Hierarchical paired channel fusion network for street scene change detection," *IEEE Transactions on Image Processing (TIP)*, vol. 30, pp. 55–67, 2020.
- [9] M. Golparvar-Fard, F. Pena-Mora, and S. Savarese, "Monitoring changes of 3D building elements from unordered photo collections," in *Proceedings of the IEEE International Conference on Computer Vision Workshops (ICCVW)*. IEEE, 2011, pp. 249–256.
- [10] S. Agarwal, Y. Furukawa, N. Snavely, I. Simon, B. Curless, S. M. Seitz, and R. Szeliski, "Building rome in a day," *Communications of the ACM*, vol. 54, no. 10, pp. 105–112, 2011.
- [11] Y. Furukawa and C. Hernández, "Multi-View Stereo: A tutorial," *Foundations and Trends® in Computer Graphics and Vision*, vol. 9, no. 1-2, pp. 1–148, 2015.
- [12] T. Pollard and J. L. Mundy, "Change detection in a 3-d world," in *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2007, pp. 1–6.
- [13] A. O. Ulusoy and J. L. Mundy, "Image-based 4-d reconstruction using 3-d change detection," in *Proceedings of the European Conference on Computer Vision (ECCV)*. Springer, 2014, pp. 31–45.
- [14] A. Taneja, L. Ballan, and M. Pollefeys, "Image based detection of geometric changes in urban environments," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*. IEEE, 2011, pp. 2336–2343.
- [15] R. Qin and A. Gruen, "3D change detection at street level using mobile laser scanning point clouds and terrestrial images," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 90, pp. 23–35, 2014.
- [16] K. Sakurada, T. Okatani, and K. Deguchi, "Detecting changes in 3D structure of a scene from multi-view images captured by a vehicle-mounted camera," in *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2013, pp. 137–144.
- [17] I. Eden and D. B. Cooper, "Using 3D line segments for robust and efficient change detection from multiple noisy images," in *Proceedings of the European Conference on Computer Vision (ECCV)*. Springer, 2008, pp. 172–185.
- [18] Y. Zhan, K. Fu, M. Yan, X. Sun, H. Wang, and X. Qiu, "Change detection based on deep siamese convolutional network for optical aerial images," *IEEE Geoscience and Remote Sensing Letters*, vol. 14, no. 10, pp. 1845–1849, 2017.
- [19] E. Guo, X. Fu, J. Zhu, M. Deng, Y. Liu, Q. Zhu, and H. Li, "Learning to measure change: Fully convolutional siamese metric networks for scene change detection," *arXiv preprint arXiv:1810.09111*, 2018.
- [20] P. F. Alcantarilla, S. Stent, G. Ros, R. Arroyo, and R. Gherardi, "Street-view change detection with deconvolutional networks," *Autonomous Robots*, vol. 42, no. 7, pp. 1301–1322, 2018.
- [21] K. Sakurada, M. Shibuya, and W. Wang, "Weakly supervised silhouette-based semantic scene change detection," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 6861–6867.
- [22] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, vol. 39, no. 6, pp. 1137–1149, 2016.
- [23] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2017, pp. 2117–2125.
- [24] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*. IEEE, 2017, pp. 2961–2969.
- [25] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," *arXiv preprint arXiv:1804.02767*, 2018.
- [26] H. Law and J. Deng, "Cornernet: Detecting objects as paired key-points," in *Proceedings of the European Conference on Computer Vision (ECCV)*. Springer, 2018, pp. 734–750.
- [27] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "SSD: Single shot multibox detector," in *Proceedings of the European Conference on Computer Vision (ECCV)*. Springer, 2016, pp. 21–37.
- [28] M. Bansal, A. Krizhevsky, and A. Ogale, "Chauffeurnet: Learning to drive by imitating the best and synthesizing the worst," in *Proceedings of the Robotics: Science and Systems*, 2019.
- [29] A. Bühler, A. Gaidon, A. Cramariuc, R. Ambrus, G. Rosman, and W. Burgard, "Driving through ghosts: Behavioral cloning with false positives," *arXiv preprint arXiv:2008.12969*, 2020.
- [30] S. Hecker, D. Dai, A. Liniger, M. Mahner, and L. Van Gool, "Learning accurate and human-like driving using semantic maps and attention," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2020, pp. 2346–2353.
- [31] S. Casas, W. Luo, and R. Urtasun, "Intentnet: Learning to predict intention from raw sensor data," in *Proceedings of the Conference on Robot Learning*. PMLR, 2018, pp. 947–956.
- [32] Y. Chai, B. Sapp, M. Bansal, and D. Anguelov, "Multipath: Multiple probabilistic anchor trajectory hypotheses for behavior prediction," in *Proceedings of the Conference on Robot Learning*. PMLR, 2020, pp. 86–99.
- [33] N. Djuric, V. Radosavljevic, H. Cui, T. Nguyen, F.-C. Chou, T.-H. Lin, N. Singh, and J. Schneider, "Uncertainty-aware short-term motion prediction of traffic actors for autonomous driving," in *Proceedings of the IEEE Winter Conference on Applications of Computer Vision (WACV)*. IEEE, 2020, pp. 2095–2104.
- [34] N. Bodla, B. Singh, R. Chellappa, and L. S. Davis, "Soft-nms – improving object detection with one line of code," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 5561–5569.
- [35] A. Düzçeker, S. Galliani, C. Vogel, P. Speciale, M. Dusmanu, and M. Pollefeys, "DeepVideoMVS: Multi-view stereo on videowith recurrent spatio-temporal fusion," *arXiv preprint arXiv:2012.02177*, 2020.
- [36] X. Shi, Z. Chen, H. Wang, D.-Y. Yeung, W.-K. Wong, and W.-c. Woo, "Convolutional lstm network: A machine learning approach for precipitation nowcasting," *arXiv preprint arXiv:1506.04214*, 2015.
- [37] D.-A. Clevert, T. Unterthiner, and S. Hochreiter, "Fast and accurate deep network learning by exponential linear units (elus)," *arXiv preprint arXiv:1511.07289*, 2015.
- [38] J. L. Ba, J. R. Kiros, and G. E. Hinton, "Layer normalization," *arXiv preprint arXiv:1607.06450*, 2016.
- [39] H. Rezatofighi, N. Tsoi, J. Gwak, A. Sadeghian, I. Reid, and S. Savarese, "Generalized intersection over union: A metric and a loss for bounding box regression," in *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2019, pp. 658–666.
- [40] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 2980–2988.