

City-Scale Road Audit System using Deep Learning

Sudhir Yarram Girish Varma C.V. Jawahar

Abstract—Road networks in cities are massive and is a critical component of mobility. Fast response to defects, that can occur not only due to regular wear and tear but also because of extreme events like storms, is essential. Hence there is a need for an automated system that is quick, scalable and cost-effective for gathering information about defects. We propose a system for city-scale road audit, using some of the most recent developments in deep learning and semantic segmentation. For building and benchmarking the system, we curated a dataset which has annotations required for road defects. However, many of the labels required for road audit have high ambiguity which we overcome by proposing a label hierarchy. We also propose a multi-step deep learning model that segments the road, subdivides the road further into defects, tags the frame for each defect and finally localizes the defects on a map gathered using GPS. We analyze and evaluate the models on image tagging as well as segmentation at different levels of the label hierarchy.

I. INTRODUCTION

Cities across the globe are growing incredibly fast, both in area as well as population. Road network is a very important component of a city and any small disruption like traffic jams have a high cost in terms of safety, efficiency, and quality of life of citizens. Over time, even well-built roads degrade to form defects due to regular wear and tear, or even due to dynamic weather conditions like rain or storm. So, a process for frequent monitoring and identification of defects is required to resolve them. We call this process the “road audit”. However, the sheer scale of the problem rules out any manual intervention at the reporting stage. An automated system that is cost-effective, scalable and easy to implement is essential.

Processing images and inferring insights has been a well-known task in computer vision. The use of computer vision methods for road inspection was first proposed by [1]. Later, deep convolutional neural networks were used to build a classifier for classifying image patches as ‘crack’ or ‘non-crack’ in [2]. Since then, the problem of semantic segmentation has seen significant improvements in performance due to deep learning based methods like Fully Convolutional Network [3] and availability of large datasets for autonomous navigation [4], [5]. These models and data sets enabled training versatile image feature representations for road scenes. However, autonomous navigation datasets only have a generic road label that needs to be further finely

The authors are with Centre for Visual Information Technology, Kohli Center for Intelligent System, IIIT Hyderabad.

Author’s Email Addresses : yarramsudhirkumar.reddy@gmail.com, girish.varma@iiit.ac.in, jawahar@iiit.ac.in

For project page containing code and dataset see: <https://cvit.iiit.ac.in/research/projects/cvit-projects/city-scale-road-audit>

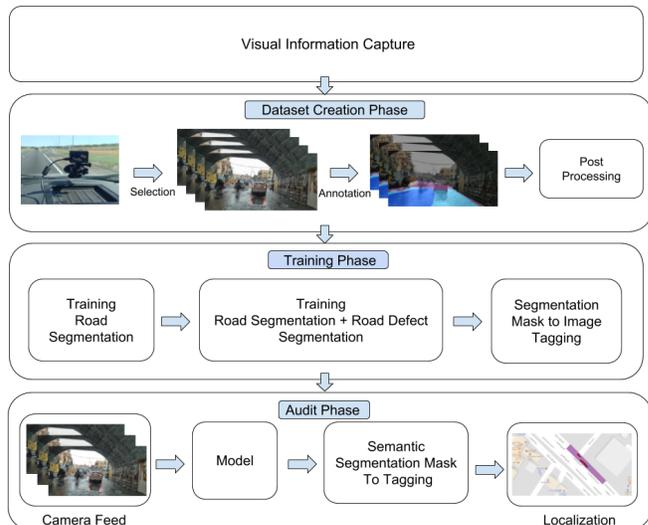


Fig. 1. The four stages of our system: i) Visual Information Capture ii) Dataset Creation iii) Model Training iv) Audit (or Testing) Phase.

classified into the various road defects for the purpose of road audit. Many of the road defects like potholes, water logging can be ambiguous due to subtle differences in the texture of the surface and hence require more carefully designed methods.

We build a cost-effective city-scale road audit system using some of the most recent developments in deep learning. Our system uses uncalibrated camera setup to capture visual information that can be mounted on vehicles that normally ply the roads compared to sophisticated setups required for autonomous navigation (see Section III-A). We present a road-audit dataset collected from highly unstructured driving environments having annotations specific to the road audit requirements (see Section IV). To minimize ambiguity among labels, we design a label hierarchy. We extract the relevant information about road quality from the dataset by modeling the problem as a cascaded semantic segmentation problem (see Section III). This is followed by tagging each frame for defects using the semantic segmentation maps (see Section III-D). We analyze and evaluate the models on image tagging as well as segmentation at different levels of the label hierarchy (see Section VI). Finally, we localize the defects and plot them on a map according to severity. We ensure that all processing steps can be implemented in real time and be installed with little effort in a normal vehicle. Hence using our system, a few vehicles can do a city-wide road audit greatly decreasing the man-hours required.

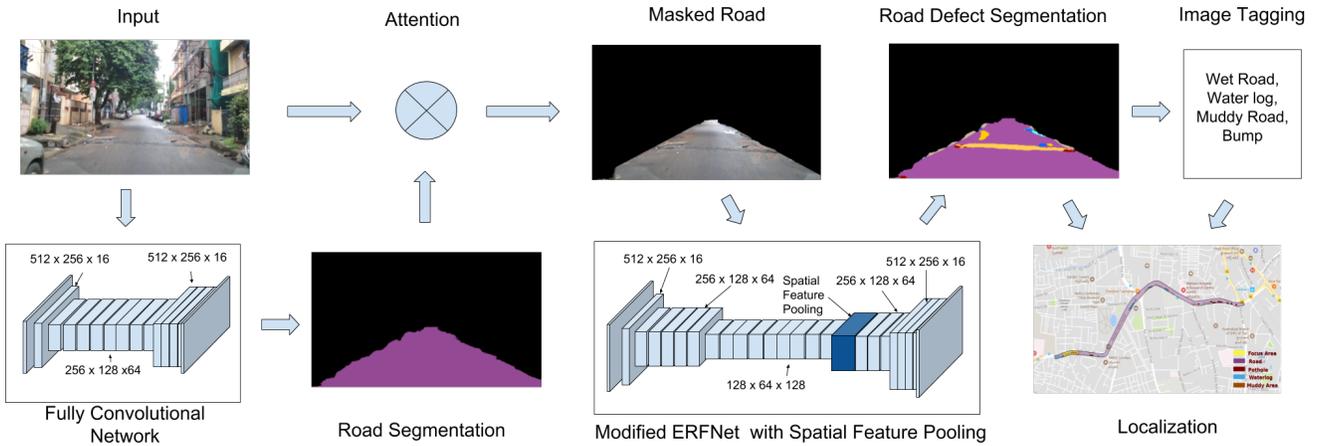


Fig. 2. A high-level overview of our proposed system. It is a multi-step procedure that uses more recent advances in real-time semantic segmentation. We segment the road, use it as an attention mechanism for obtaining road pixels, then use a powerful semantic segmentation model to segment multiple road defects. Finally, we tag the image frame for the defects and localize it in the map using GPS.

II. RELATED WORKS

Road audit is mostly conducted by manual inspectors. This could also be crowdsourced to volunteers. Some of the initial automated systems used GPS and accelerometer only, followed by computer vision techniques for road audit.

A. Non-Computer Vision methods.

A system for pothole detection was proposed using just vibration and GPS sensor in [6] and using smartphone sensors like accelerometer and GPS in [7]. These sensors are highly error-prone and fail to detect other road defects like water-logging, muddy roads, and even minor rough patches that cannot be easily inferred from them. While our system depends on more reliable visual information with robust deep learning techniques and also caters to detecting more defects than just a pothole.

B. Computer Vision methods.

The first paper to propose the use of computer vision techniques for road inspection was [1]. They used super-pixel based segmentation techniques to identify road distress mostly restricted to cracks, in highly organized road conditions such as in a developed countries. Image-based road distress detection was surveyed in [8]. Various crack segmentation methods are compared in [9]. The community interest in crack detection has been growing recently [10], [11], [12]. An integrated system for crack detection is proposed in [13]. A CNN based neural networks to detect cracks was proposed by [2], [14].

All these efforts were more focused on detection of cracks, potholes, and are more specialized for a specific region of roads, making it less adaptable to other, mostly unstructured regions. Most cities outside the developed world suffer from a much more diverse set of defects like water logging, rough roads, wet roads, muddy roads, etc. These defects are in some sense similar to cracks but they are more complex, ambiguous and the number of different defects makes it a

challenging problem. Also, most of the previous methods use handcrafted features and traditional classifier like SVM with some using deep learning techniques for limited tasks. In this paper, we try to address more generic setting of road audit in unstructured road conditions. We also leverage the advancements in real-time semantic segmentation and autonomous navigation datasets to build an end-to-end deep learning model.

III. APPROACH

We now describe our proposed system. Our system has four phases: i) Visual Information Capture ii) Dataset Creation iii) Model Training iv) Audit (or Testing) Phase as shown in Figure 1. Dataset creation phase will be described in Section IV. We describe the rest of the phases as well as the details of the deep learning model architecture in this section.

A. Phase I: Visual Information Capture

Our system uses videos/images captured using a camera mounted on vehicles for information gathering. We achieve this by using a single uncalibrated camera setup, that could even be a GPS enabled smartphone camera, making it cost-effective and scalable. The camera is mounted inside the vehicle so that there are lesser chances of exposure to climatic conditions, ambient dust and also to avoid accidental loss of the camera. Using this setup, the system can be implemented with little or no additional cost making it cost-effective and scalable.

B. Phase II: Model Training

We model the problem of localizing the road defects as a cascaded semantic segmentation problem, followed by image tagging and localization. All models and algorithms are chosen such that the system can be run in real-time at audit phase. This is required since the information gathering method described above generates a huge amount of data and it is not easy to store it and process it later. So our model

focuses on extracting the required information in real time. It also captures the fine-grained information necessary to get statistics of road defects that are further localized.

a) *Road Segmentation*: As our primary goal is to segment road defects, the context extracted from dynamic and ambient objects like other vehicles, the sky, buildings, pedestrians etc. provides very little additional information for the model to segment road defects. Moreover, these might mislead the model. So, as the first stage, we segment road pixels from the background. As shown in Section V, this proves to be effective.

b) *Fine-grained Segmentation of Roads*: Traditional semantic segmentation datasets like Cityscapes [4], KITTI [5] have the notion of objects, which implies that they are reasonably well defined in the space of shape and color. However, as part of our road audit dataset, we have to deal with defects which are agnostic to the notion of shape, in a way, which can only be semantically differentiated through texture. Incorporating these cues in the architecture as a second step, we further segment the road pixels into more fine-grained labels.

C. Real-time Model Architecture

In this section, we describe the different deep learning models used for solving this problem. We use a baseline network, as well as a custom designed network that incorporates the road segmentation and fine-grained defect segmentation into the network architecture. In the interest of creating a practical application, we have considered various models. Considering a right balance between efficiency and speed, we chose ERFNet [15] (69.8% with 41fps on Cityscapes) as our base-model.

With ERFNet as base-model, we experiment with two models: i.) fine-tuning a pre-trained cityscape model on our dataset ii.) train the model from scratch on our dataset. Note that both these models are initialized with the image features trained for Imagenet classification. ERFNet was designed for the application of autonomous navigation. Hence we also propose a modified architecture, which is more suited for the task of road defect segmentation.

Our proposed model is called the refined ERFNet, that contains a Cascade module and Spatial Feature Pooling module on top of ERFNet. These modules incorporate the road segmentation as well as the fine-grained defect segmentation into the model architecture as shown in Figure 2. As an overview, our network takes an image as an input and outputs a semantic mask of the road and road related defects. Our model works in multiple stages with Cascade Module and Spatial Feature Pooling Module designed specifically for the problem.

1) *Cascade Module*: Cascade module consists of two submodules - road segmentation module and road defect segmentation modules sequentially. The task of the road segmentation module is to segment out road pixels from the background (non-road) pixels. For this, we use shortened ERFNet, trained on cityscapes and fine tune it to segment road pixels. Then, the segmented mask generated is used

to pass input road pixels to the second module using the attention mechanism. These road pixels are run through road-defect segmentation module, which further segments out road defects. These two models are trained in an end-to-end procedure.

2) *Spatial Feature Pooling Module*: The Spatial Feature Pooling Module aggregates the features for different scales of the image segments and tries to capture texture level cues. This caters to different scale variation in the road defects.

D. Phase IV : Audit

In the audit phase, we deploy the real-time cascaded semantic segmentation model on a computer installed in the vehicle. It generates segmentation masks for the frames captured from the camera and does the following sequentially:

1) *Image Tagging*: After inferring the segmentation output from the input image, we decide on the appropriate labels for an image by thresholding the number of pixels segmented as that label. The threshold is decided by searching in the hyperparameter space on the validation data of the dataset.

2) *Localization*: The final stage of our system is to plot the road defects according to their severity on a map. We use the GPS information captured along with the image frame for this purpose. To indicate the severity, we use the number of pixels in the image classified as that particular label. However, doing this alone might not capture the severity, since a large defect could occupy only a small region due to viewing angles or even other vehicles blocking the view. Hence the severity score also needs to be averaged across nearby frames for getting a more accurate estimate.

IV. DATASET

In this section, we describe the dataset creation phase of our system. Our motto behind the methodology of constructing the dataset was to capture very complex real-world scenes in an even more generic way. We collected data from many drive sequences (distinctive captures) spanning close to 60 kilometers, out of which we have selected around 18 driving sequences. These drive sequences cover a variety of highway roads, flyovers, and street roads. It covers a multitude of scenarios, right from having differently sized defects to various combination of defects in a single image. The dataset also incorporates a variety of traffic scenarios with distinct levels of diversity in terms of vehicles covering from sparsely crowded to densely crowded. It also incorporates diverse road settings, with partial to complete occlusions of the various road defects. In order to cover seasonal defects like water logging and wet roads, we had driven just after the rainy days.

In the frame selection process, we have ensured that we have taken well-curated images in terms of avoiding any kind of glare due to reflection. In order to counter the motion blur induced by residual vibration of the vehicle and due to sudden movements of the vehicle, we have carefully selected images that had blur below a threshold.

We sample frames from each of the sequences. Our sampling was driven by the idea to cover a wide variety

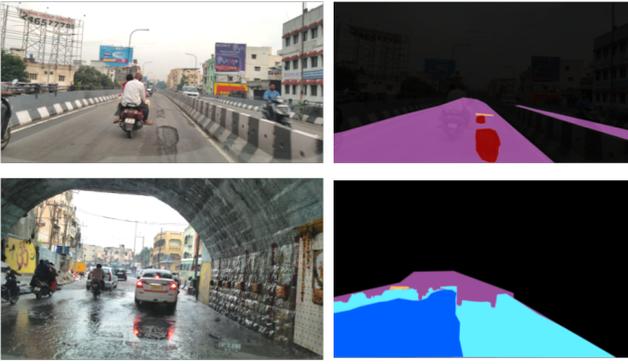


Fig. 3. The urban road is segmented for road defects. We can observe that road pixel annotation also includes pixels belonging to other objects like car, bike, etc., on the road which are later masked out as a post processing step.

of road defects. We annotate them coarsely. The dataset has images of size 1024×2048 px. This dataset segments various semantic defects a road can have.

1) *Labels*: After careful collection of data, we have analyzed the kinds of road defects and have come up with tar road, pothole, water log, muddy road, obstruction, wet road, shoulder, cement road, bump labels. These labels are grouped in a three-level hierarchy. At the top, we have road and road-defects. Road is branched into class level labels namely cement road, tar road and shoulder labels and the rest go into road-defects. Then, road defects are subdivided into four categories category 1, category 2, category 3, category 4. Category 1 is subdivided into pothole, water log and wet road class labels, category 2 contains muddy road label. category 3 contains rough road label while category 4 is subdivided into obstructions and bump.

2) *Annotation Method*: We employ pixel-level annotation. We do this for two reasons 1) Annotation by image tagging is very subjective (How much area of a road image occupied by pothole be tagged as the pothole?). However, employing a supervised pixel level annotation allows annotators to mark all pixels that belong to label (say, pothole), thereby decreasing the subjectivity from one annotator to another. 2) Pixel level annotation provides more supervision for the model to learn better. However, in the interest of resources, we have restrained ourselves to complete the task of annotation and basic quality control to about 15 minutes on average for a single frame. This is significantly less when compared to 45 minutes taken for annotation of typical urban road scene as reported by the cityscapes dataset [4] in their annotation protocol. Coming to our protocol, annotators were first asked to figure out the boundaries of road and shoulder. Then they annotate the road by ignoring all vehicles/objects/people present on the road. Later they segment the shoulder (or side-road). Then, they segment out road defects from already segmented road pixels circumventing the occlusions. The annotation was carried out in a coarse way; ensuring that all the essential pixels of a label are captured while loosening out the precision at label boundaries. Though this would hinder the performance of supervised algorithms, it was a necessary move, keeping in mind the sparsity of resources. Figure 3 provides a sample annotation.

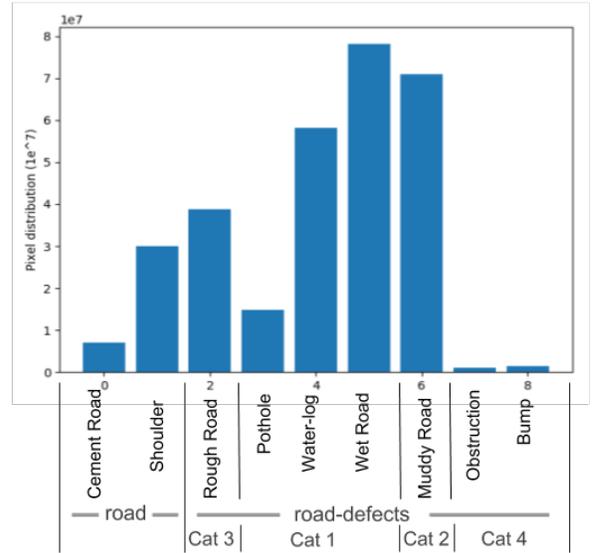


Fig. 4. Here 'Cat' means Category. Pixel distribution for various defects.

3) *Post Processing*: In order to mask out other vehicle/object pixels included along with the road segments, we run a pre-trained cityscapes model on these images which segments other object pixels separated from road pixels. This generated cityscapes label mask and annotator generated annotation mask are combined to come up with a segmentation mask for the road and its defects excluding all other object pixels. Though this was not completely error-free, from our qualitative analysis, it was good enough for the task at hand. We have used cityscapes trained ERFNet for generating cityscape label mask and this is appropriate as the road pixel IoU was 97.9 which is close to state of the art (98.6) evaluated on cityscapes dataset.

4) *Label Statistics*: The pixel distribution for each label in the train and in the test are ensured to be almost the same. Dataset labels pixel distribution is provided in Figure 4. The labels and the coarse annotation are sufficient in detecting the road defects in a satisfactory manner for the following two reasons. The labels that were incorporated cover all the major issues, which are generally reported in various surveys. The coarse annotation works for the task at hand as our primary focus is to detect the defects, which does not require pixel-level precision.

V. EXPERIMENTS

For all our experiments we have used the newly collected Indian Road dataset that was described in Section IV. We train three semantic segmentation networks: ERFNet from scratch, fine-tuning cityscapes pre-trained ERFNet and refined ERFNet. We consider the first 2 as our baseline and compare them against the refined ERFNet.

a) *Refined ERFNet*: We integrate the proposed Cascaded Spatial Feature Pooling module on the ERFNet architecture. We first train the initial module that segments out road pixels. We generate ground truth mask for road pixels as the union of ground truth of all the road defects. We train this module until the accuracies reach a satisfactory level. On

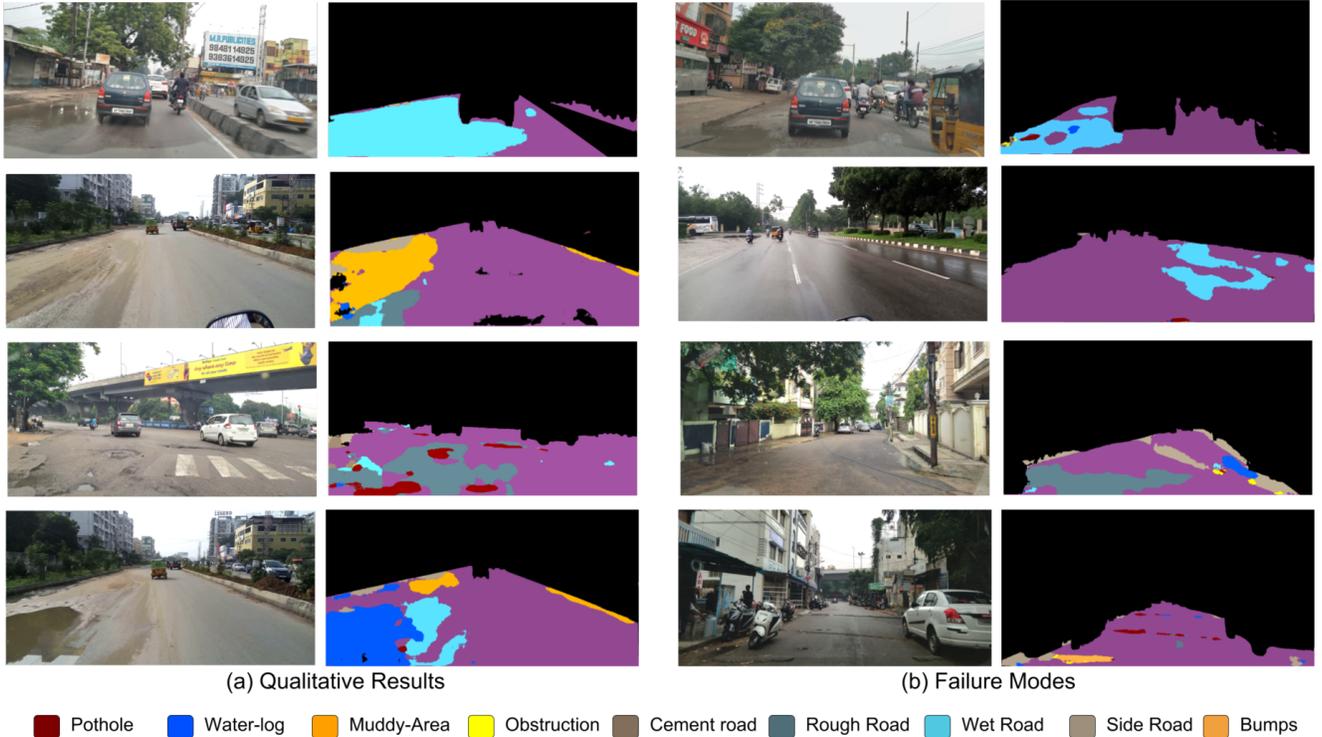


Fig. 5. (a) Qualitative Results for class level segmentation. These figures show the input image along with its segmentation mask as predicted by the refined ERFNet model. (b) Examples of Failures. Observations in the second column from top to bottom 1) Waterlog is segmented as wet road, 2) Only part of the wet road is segmented, 3) road is segmented as rough road 4) Bumps are not properly segmented.

top of this, we now attach a module whose task is to segment out road defect pixels. This module has an additional spatial feature pooling module. The generated output is compared with road defect ground truth to generate loss that is back-propagated.

We used the publicly available ERFnet code. It replaces the bottleneck module in the Resnet [16] with more efficient and better-performing 1D-non-bottleneck module. We modified this network by adding the cascade module along with the spatial feature pooling module. For training, we have an input image of size 1024×512 . We have used Adam optimizer with a learning rate of $5e-4$ for a batch size of 14.

1) *Metrics*: Since we are modeling the problem as a semantic segmentation problem, we use Intersection over Union (IoU) scores as accuracy metric. We evaluate the mean Intersection-over-Union. We also report the weighted mean Intersection over Union (wIoU) score. As pixel distribution among labels varies a lot, wIoU would give a good measure on how the algorithm is performing for all the pixels overall.

2) *Training complete setup end to end*: As described above, our model is trained end to end with a two-step training procedure. To gain more insights into the training procedure, we have independently trained the road level segmentation and defect level segmentation. This is contrasted with training in a two-step manner by first training road segmentation module to a satisfactory level and then attaching road-defect module and further train both together. We observed that the latter approach was better, resulting in a faster network convergence with similar performance.

TABLE I
MEAN IOU AND INFERENCE SPEED ON THE TEST SET OF INDIAN ROAD DATASET FOR CATEGORY LEVEL AND CLASS LEVEL FOR THREE SEGMENTATION NETWORKS (TABLE IA). (TABLE IB) IOU SCORES COMPUTED FOR DIFFERENT LEVELS IN DATASET HIERARCHY USING OUR REFINED ERFNET. HERE 'CAT4' MEANS CATEGORY 4 (TABLE IC) IOU SCORES FOR INDIVIDUAL CLASSES USING REFINED ERFNET.

Method (Category Level)	IoU (%)	wIoU (%)	fps (1024×512)
ERFNet (Fine-tuned)	41.8	59.8	41.7
ERFNet (Scratch)	42.1	59.8	41.7
Our refined ERFnet	45.7	60.7	26.2

Method (Class Level)	IoU (%)	wIoU (%)	fps (1024×512)
ERFNet (Fine-tuned)	24.1	55.2	41.7
ERFNet (Scratch)	24.4	57.2	41.7
Our refined ERFnet	28.1	59.0	26.2

(a)

Hierarchy(Our Method)	IoU(%)
Road-defects	65.1
Category level	45.7
Class Level(-cat4)	38.0
Class Level	28.1

(b)

Our Method(Class)	IoU (%)
Tar Road	72.5
Pot hole	23.5
Rough Road	34.5
Cement Road	13.8
Obstruction	10.7
Wet Road	32.3
Water log	28.8
Muddy Road	33.4
Bumps	03.3
Shoulder	28.1

(c)

TABLE II

OUR EXPERIMENT OF TAGGING EACH IMAGE AND THRESHOLDING THE GENERATED SEGMENTATION MASK HAS GIVEN AN F1 SCORE OF 50.3%

Our Method (Image Tagging)	Precision (%)	Recall (%)
Pot hole	59.3	83.3
Rough Road	67.7	61.7
Cement Road	6.2	25.0
Obstruction	6.6	100.0
Wet Road	81.3	53.3
Water log	56.2	54.5
Muddy Road	74.1	76.6
Bumps	28.5	50.0
Shoulder	16.0	25.0

VI. RESULTS

The segmentation results of refined ERFNet and baseline are listed in Table I. Refined ERFNet outperforms the original baselines by 3.6%, 3.5% for class level labels and category level labels respectively. The qualitative results are provided in Figure 5. As present in Table I, refined ERFNet gives a mean Intersection over union score of 65% for (road vs defects), 45.77% for category level (4 labels), 38.00% for class level (8 labels excluding category 4) and 28.13% for class level (10 labels). The significant drop in accuracy from category level to class can be attributed to fine-grained separations among labels making it hard for the model. To some extent, the visual complexity in differentiating these labels was unavoidable during annotation. However, the municipal authorities can decide on what level of label hierarchy would suit their task and work accordingly. Our model infers at a speed of 26.2 fps on the Titan X GPU for an image of size 1024×512 . Per class IoU scores are also provided in Table I.

a) Cascade Module and Spatial feature pooling Module: We have experimented with a hierarchical segmentation of segmenting road pixels followed by road defects. Experiments show a boost in IoU scores by 3.6%, 3.5% for class and category levels respectively. This validates that our model of first segmenting out road pixels (which predominantly uses object cues like shape, color) and then segmenting out road defects is well suited for this task. It can also be inferred that our spatial feature pooling module is better at capturing the essential information necessary for segmenting out the fine-grained pixel differences among labels.

b) Image Tagging: As our primary objective is to tag an image with road defects. We conducted an experiment on how would the model perform for an image tagging task. As ground truth, we asked the annotators to tag each image with all the class labels. Now, after generating a segmentation mask, we threshold the number of pixels belonging to each class and accordingly tag image with the corresponding label. With this, we achieve an F1 score of 50.3% as shown in Table II. This shows that the model is good enough to recognize appropriate labels for an image. This provides additional validation for the appropriateness of the model to be used for practical purpose.

c) Failure Modes: Figure 5 gives examples of the failure modes of our model. The primary reason for failure

is the essential similarity among labels. For example, in the top right example, we see that a water logging but is mislabeled as a wet road. However, precisely identifying such fine-grained differences are beyond human skills as many factors like illumination, lighting, surface reflectance play a crucial role. Other less common modes of failure are due to the color variations in the texture of road due to sediments, inconsistency in road material.

VII. CONCLUSION

We propose an automated system for road audit in large and unstructured cities. Our system is designed to be cost-effective, scalable and real time. Our data collection can be done using smartphones mounted inside normal vehicles. We curate a dataset with the required annotations for identifying fine-grained road defects, in unstructured road conditions. We propose a cascaded semantic segmentation model that can be trained using the dataset for segmenting the road defects. Furthermore, we tag the image with the defects using the segmentation masks and finally localize them on a map. We analyze and evaluate our models and show that they give reasonable performance for the task at hand. We expect that works along similar lines can help urban authorities identify areas vulnerable to damage by analyzing the changes roads underwent over a period of time. It could also be extended to infer reasons for traffic congestion, accident-prone areas etc.

REFERENCES

- [1] S. Varadharajan, S. Jose, K. Sharma, L. Wander, and C. Mertz, "Vision for road inspection," in *WACV*, 2014.
- [2] L. Zhang, F. Yang, Y. D. Zhang, and Y. J. Zhu, "Road crack detection using deep convolutional neural network," in *ICIP*, 2016.
- [3] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *CVPR*, 2015.
- [4] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The cityscapes dataset for semantic urban scene understanding," in *CVPR*, 2016.
- [5] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The kitti dataset," *IJRR*, 2013.
- [6] J. Eriksson, L. Girod, B. Hull, R. Newton, S. Madden, and H. Balakrishnan, "The pothole patrol: Using a mobile sensor network for road surface monitoring," in *MobiSys*, 2008.
- [7] R. Bhoraskar, N. Vankadhara, B. Raman, and P. Kulkarni, "Wolverine: Traffic and road condition estimation using smartphone sensors," in *COMSNETS*, 2012.
- [8] K. C. Wang and O. Smadi, "Automated imaging technologies for pavement distress surveys," *Transportation Research E-Circular*, 2011.
- [9] V. Kaul, Y. Tsai, and R. Mersereau, "Quantitative performance evaluation algorithms for pavement distress segmentation," *TRR Journal*, 2010.
- [10] M. Salman, S. Mathavan, K. Kamal, and M. Rahman, "Pavement crack detection using the gabor filter," in *ITSC*, 2013.
- [11] S. Mathavan, K. Kamal, and M. Rahman, "A review of three-dimensional imaging technologies for pavement distress detection and measurements," *ITS*, 2015.
- [12] R. Medina, J. L. Fernández, E. Z. Casanova, and J. G. García-Bermejo, "Enhanced automatic detection of road surface cracks by combining 2d/3d image processing techniques," in *ICIP*, 2014.
- [13] H. Oliveira and P. L. Correia, "Automatic road crack detection and characterization," *ITS*, 2013.
- [14] M. Quintana, J. Torres, and J. M. Menéndez, "A simplified computer vision system for road surface inspection and maintenance," *ITS*, 2016.
- [15] E. Romera, J. M. Alvarez, L. Bergasa, and R. Arroyo, "Erfnet: Efficient residual factorized convnet for real-time semantic segmentation," *ITS*, 2017.
- [16] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *CVPR*, 2016.