

# Mapping Road Safety Features from Streetview Imagery: A Deep Learning Approach

Arpan Man Sainju

asainju@crimson.ua.edu

Department of Computer Science

The University of Alabama

Tuscaloosa, Alabama

Zhe Jiang

zjiang@cs.ua.edu

Department of Computer Science

The University of Alabama

Tuscaloosa, Alabama

## ABSTRACT

Each year, around 6 million car accidents occur in the U.S. on average. Road safety features (e.g., concrete barriers, metal crash barriers, rumble strips) play an important role in preventing or mitigating vehicle crashes. Accurate maps of road safety features is an important component of safety management systems for federal or state transportation agencies, helping traffic engineers identify locations to invest on safety infrastructure. In current practice, mapping road safety features is largely done manually (e.g., observations on the road or visual interpretation of streetview imagery), which is both expensive and time consuming. In this paper, we propose a deep learning approach to automatically map road safety features from streetview imagery. Unlike existing convolutional neural networks (CNNs) that classify each image individually, we propose to further add Recurrent Neural Network (Long Short Term Memory) to capture geographic context of images (spatial autocorrelation effect along linear road network paths). Evaluations on real world streetview imagery show that our proposed model outperforms several baseline methods.

## KEYWORDS

Deep Learning, Streetview Imagery, Road Safety

### ACM Reference Format:

Arpan Man Sainju and Zhe Jiang. 2019. Mapping Road Safety Features from Streetview Imagery: A Deep Learning Approach. In *UrbComp '19: 8th SIGKDD International Workshop on Urban Computing, August 05, 2019, Anchorage, AK*. ACM, New York, NY, USA, 9 pages. <https://doi.org/10.1145/nnnnnnn.nnnnnnn>

## 1 INTRODUCTION

Every year, around 6 million car accidents occur in the U.S. on average [13]. Traffic safety has long been an important societal issue. In order to avoid or mitigate vehicle crashes, traffic engineers place roadside barriers to prevent out of control vehicles from diverting off the roads and hitting the roadside hazards. Such road safety features can also prevent vehicles from crossing into the path of other vehicles. During winter season, vehicles can become more difficult to control on icy and slippery road surface, particularly when the vehicle speed is high. Barriers on the roadside can act as a safety precaution in such cases. Other safety features such as rumble strips help alert inattentive drivers who are deviating from their lanes. Figure 1 shows three different common type of road safety features, rumble strip, concrete barrier, and metal crash barrier.

Federal, state and local governments spent several hundred billion dollars each year on transportation infrastructure development and maintenance [16]. Mapping safety features along road networks can play a crucial role in managing and maintaining road safety infrastructures. Traffic engineers can use the detailed safety feature map to identify locations where new safety infrastructure should be invested.

In current practice, mapping road safety features are mostly done manually by well-trained traffic engineers driving through road networks or visually interpreting streetview images. A **streetview image** is a geo-referenced image taken at a specific location on the ground. One common example is Google Streetview Imagery collected by vehicles equipped with GPS and cameras driving along streets on road networks. However, such a manual process is both expensive and time-consuming. Given the large amount of information to collect, the cost of these approaches quickly become prohibitive.

The focus of this paper is to develop a deep learning algorithm that can automatically map road safety features from streetview imagery. The results can be used by the transportation agencies in management and maintenance of road safety infrastructures, as well as planning the investment on new infrastructures. Specifically, we can utilize a small set of manually labeled imagery (whose road safety features are visually inspected) to learn a classification model. Then the model can be used to classify safety feature types on a large number of unlabeled imagery along the road network.

However, mapping the road safety features based on streetview imagery poses several unique challenges. First, streetview images are not independent and identically distributed along a road network. In contrast, the safety feature types of consecutive images along a same road network path often resemble each other (also called the spatial autocorrelation effect). Second, the spatial scales of road safety features may vary across different class categories. For example, concrete barrier is often very long (e.g., miles). In contrast, metal crash barriers are much shorter (e.g., hundred meters). Third, individual images may be imperfect due to some noise or obstacles. For example, a safety feature can be blocked by a large vehicle and thus become invisible in an image.

To address these challenges, we propose a deep learning model based on both convolutional and recurrent units. We use convolutional neural network (CNN) model to extract semantic features from individual images. We also use a recurrent neural network, Long Short-Term Memory (LSTM), to model spatial sequential structure on extracted features from consecutive images along a road network path (the spatial autocorrelation effect). The integration of CNN and LSTM enables our deep learning model to utilize not



**Figure 1: Three common classes of road safety features on Google Streetview Imagery. (a) Rumble strips (b) Concrete barrier (c) Metal crash barrier**

only the content of individual images but also the geographic context between images. Evaluations on real world streetview images collected from highways in Alabama show that our approach outperforms several baseline methods in classification performance.

In summary, the contributions of this paper are listed below:

- To the best of our knowledge, we are the first to explore a deep learning approach on Google Streetview imagery for road safety feature mapping.
- We propose to use an integrated deep learning model that combines CNN and LSTM. The integrated model can utilize not only the content of individual images but also the spatial sequential structure between images.
- We validate our approach on real world streetview imagery collected in Alabama.

The outline of the paper is as follows. Section 2 discusses some of the related works. Section 3 formally defines the problem. Section 4 introduces the approach. Section 5 summarizes the results of our experimental evaluation on a real world dataset. Section 6 concludes the paper with discussions on future works.

## 2 RELATED WORKS

In this section, we briefly review the relevant research on transportation safety and deep learning techniques for spatial and spatiotemporal data.

### 2.1 Transportation Safety

Related work in transportation safety often focuses on analyzing the protective effect of different road safety features (e.g., roadside barriers) [12, 18, 34]. For example, studies in [3, 7, 9] quantify the protective effect of barriers with regards to motorcyclist injury. Work in [4] analyzes the performance of roadside barriers related to vehicle size and type. [27] studies how to increase the effectiveness of the roadside barriers in safety protection. For example, studies found that concrete barriers can hold high-energy truck crash, but can also cause more fatalities. Some recent work focuses on developing energy absorbing barrier [21]. Beside the protective effect, other studies on roadside barriers focus on the impact on mitigating near-road air pollution [25]. The study on effect of solid

barriers on dispersion of roadway emissions in [8, 22] shows that roadside barriers is one of the most practical mitigation methods. There are also works that analyze spatial patterns from traffic accident event locations such as network hotspots and colocation patterns [17, 19, 20]. [11] proposes efficient algorithms to identify primary corridors from cyclists' GPS trajectories on urban road networks to study riding behaviors for safety issues. [15] shows techniques to detect coarse scale hotspots of road failure events through geo-tagged tweets from social media.

Recently, researchers have used Google Streetview imagery along the road network for traffic sign detection for roadway inventory management [1, 2, 26]. Other works use streetview imagery to estimate the demographic makeup of neighborhoods [6], to assess street-level greenness in an urban area [14]. To the best of our knowledge, there is little research on utilizing streetview imagery to automatically map road safety features.

### 2.2 Deep Learning for Spatio-Temporal Data

In recent years, deep learning techniques have shown great growth in the field of spatiotemporal data mining [10, 23]. One common approach is to integrate deep convolutional neural networks (CNN) with recurrent neural networks such as Long Short-Term Memory (LSTM). The CNN component can be used to model spatial dependency structure in one temporal snapshot, while the LSTM component can be used to model temporal dynamics between different snapshots. For example, [33] uses fully convolutional networks with LSTM to estimate vehicle count maps based on city cameras. [30] uses CNN-LSTM model together with multi-view learning to predict taxi demand. [28] uses a one-dimensional CNN to capture spatial features of traffic flow and two LSTM models to capture the short-term variability and periodicities of traffic flow. [29] addresses the traffic prediction problem with a new spatiotemporal model. It uses a flow gating mechanism to learn the dynamic similarity between locations, and uses a periodically shifted attention mechanism to handle long-term periodic temporal shifting. [31, 32] researches on better traffic accident prediction to improve transportation and public safety. In these existing works, LSTM model is often used to model temporal dynamics between multiple spatial snapshots. The difference from our work in this paper is that we

use LSTM to capture linear spatial sequential structure between consecutive images along a road network path.

### 3 PROBLEM DESCRIPTION

In this section, we discuss some basic concepts and describe our problem.

*Road network:* A road network is a network whose nodes are road intersections, and whose edges are road segments. At the same time, a road network is also a spatial network whose nodes are spatial points and whose edges are spatial line strings. In other words, a road network has both graph properties and geometric properties.

*Streetview imagery:* Streetview imagery is a sequence of geo-referenced images whose locations are embedded on road network edges (in the form of line strings). The imagery is collected through driving a vehicle equipped with GPS and camera, so that each image can be geo-referenced based on the GPS time stamp. In this paper, we used Google Streetview API to select imagery at a regular spatial interval of 20 meters.

*Road safety feature:* A road safety feature is defined as the measure or infrastructure placed on a road to improve safety. We consider three most common safety features: rumble strips, concrete barrier and metal crash barrier. Figure 1 shows examples of the three safety features from Google Streetview imagery.

- *Rumble Strips:* Rumble strips (Figure 1(a)) are milled grooves or rows of raised pavement markers placed perpendicular to the direction of travel, or a continuous sinusoidal pattern milled longitudinal to the direction of travel. It creates a vibration and rumbling sound transmitted through the wheels into the vehicle interior which can alert the drivers who have drifted from their lanes.
- *Concrete barrier:* Concrete barrier (Figure 1(b)) is a rigid barrier. It is easy to maintain. This type of barrier is often used on roads where traffic in opposing direction is flowing in close proximity due to lack of space.
- *Metal crash barrier:* Metal crash barrier (Figure 1(c)), also known as guardrails, is usually made from steel beams or rails. It ensures minimum damage to the vehicle and its occupants by absorbing the impact energy of the colliding vehicle. It can also act as a good visual guide during night time for the driver to maintain their lane position.

*Problem Definition:* Given a road network with geo-referenced streetview imagery sampled at an equal distance interval, as well as a small collection of labeled imagery sequences (each image has three binary class labels corresponding to the existence of rumble strips, concrete barrier, and metal crash barrier respectively), the road safety feature mapping problem aims to learn a classification model that can predict the labels for all unlabeled images on the road network. Since each image may contain multiple types of road safety features at the same time, our problem is a multi-label classification problem.

### 4 APPROACH

In this section, we introduce our proposed deep learning approach. Figure 2 illustrates the overall framework of our proposed model. The bottom component shows the data collection process. We sample a number of spatial points along road network edges

at an equal distance interval (e.g., 20 meters), and then use Google Streetview API to download geo-referenced imagery at those point locations. We fixed the distance interval of 20 meters because the average length of some road safety features such as metal crash barrier is only a few hundred meters. If we select a higher distance interval, there may not exist enough images for short extent barrier such as metal crash barrier. Although lower distance interval can provide fine-grained dataset, it increases the number of streetview images to be downloaded which incurs extra cost. The middle component of our proposed models is based on retrained CNN model to extract low dimensional features from individual images. The last component is LSTM layer. In contrast to existing works, our LSTM does not capture temporal dynamics between different spatial snapshots, but represents spatial sequential pattern between consecutive imagery along road network edges.

#### 4.1 Extract Image Feature with CNN

Convolutional Neural Network (CNN) was developed mainly for image classification. CNN introduces the concept of parameter sharing which allows the model to learn less number of parameters in comparison to regular neural network. Similar to regular neural networks, CNN also consists of a sequence of layers. We briefly describe each layer in CNN below.

*Input Layer:* Input Layer holds the raw pixel color values (RGB) of the images. Usually, the pixel values are normalized to stabilize the learning process and dramatically reduce the number of training epochs required to train deep learning models.

*Convolution Layer:* Convolutional layer transforms the input using convolution operation. A convolution operation is element-wise multiplication of a pixel and its neighborhood pixels color value (RGB) by a matrix. It is also known as convolution filter. Different filters are used to convolve around all the pixels in an image. Filters like horizontal and vertical edge detecting filter can extract the linear feature from the image. Other complicated filters such as sobel filters can extract non-linear edges. In CNNs, filters are not defined, they are learned during the training process. By stacking layers of convolutions on top of each other, we can get more abstract and in-depth information from a CNN.

*ReLU Layer:* ReLU stands for Rectified Linear Unit, which is a type of activation function commonly used in neural networks. Activation functions are applied to introduce non-linear properties to the network. The function returns 0 if it receives any negative input. However, for any positive value  $x$ , the function returns the same value back. So, it can be written as  $f(x) = \max(0, x)$ . ReLU activation function is computationally less expensive as there is no complicated math, which can reduce the model training time.

*Pooling layer:* The function of pooling layer is to reduce the spatial size of the input. It is also known as downsampling layer. Pooling layer can reduce the number of parameters and computation in the network. It applies a filter (usually of size  $2 \times 2$ ) to the input volume. Pooling filters can be based on different operations such as max, min or average. The most common one is max filter which extracts the max value from the filter region.

*Fully Connected Layer:* Fully connected (Dense) layer takes an input volume (output of activation function) and outputs a N-dimensional vector. Similar to regular neural networks, neurons in

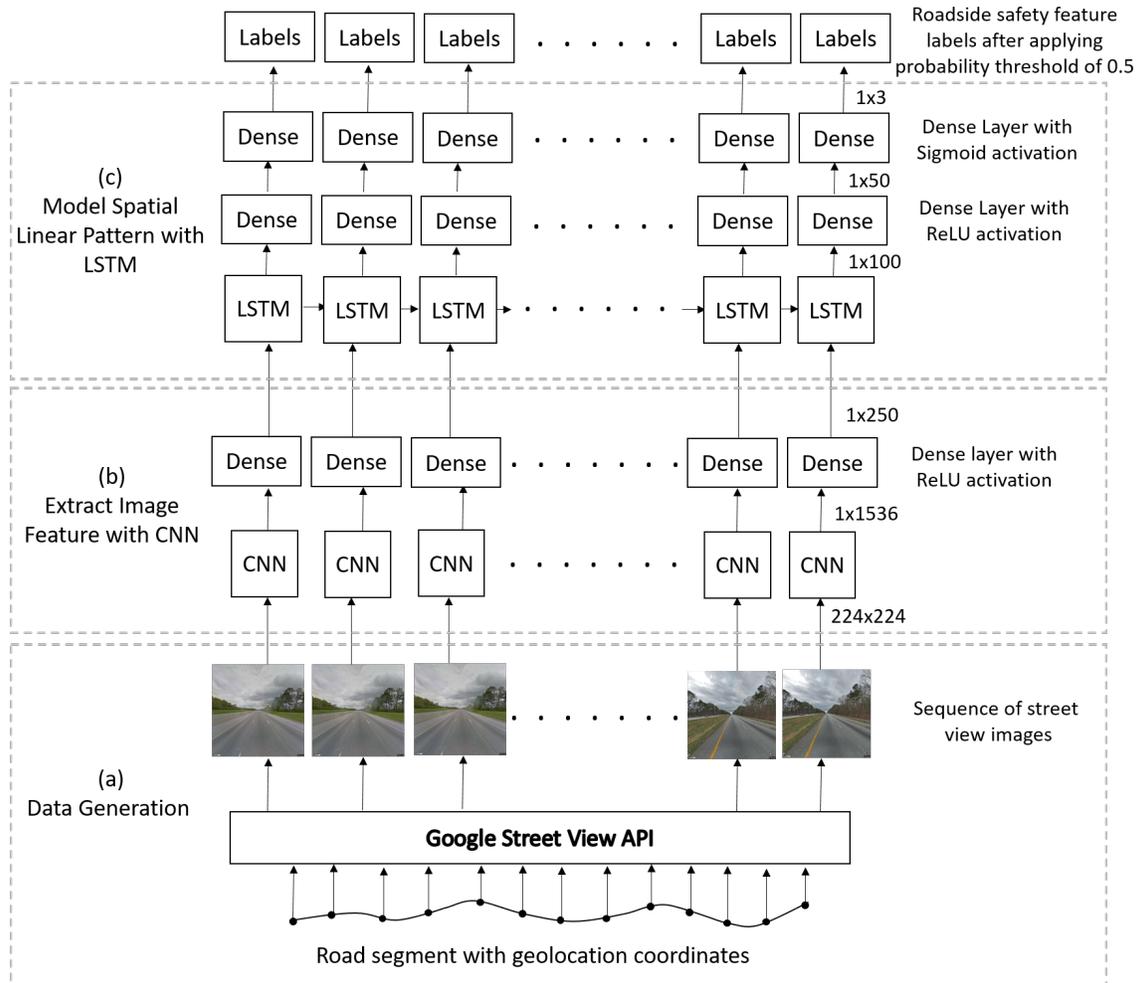


Figure 2: Overall framework of our deep learning approach

a fully connected layer have full connections to all activations in the previous layer.

For our proposed model, we use the current state-of-art Inception-ResnetV2 [24] model to extract features from the streetview images. We use the keras implementation of Inception-ResnetV2 pre-trained on ImageNet [5] dataset with 1000 classes. Inception-ResnetV2 combines the idea of residual connections to inception architecture. In residual connection, each layer feeds into the next layer and directly into the layers about few hops away. Residual connections are important for very deep architecture. When deeper networks starts converging, the accuracy can saturate at a point and eventually degrade. Residual connections are designed to overcome this degrading problem. As the Inception-v4 network is very deep with around 200 layers, combining Inception architecture with residual connections can be beneficial.

We removed the final dense layer with softmax activation function because the network was pretrained to classify 1000 classes. In our work, we only have 3 classes (rumble strips, concrete barrier, and metal crash barrier). Next, we add a dense layer with 250 nodes

after the last average pooling layer (with 1,536 nodes). We reduce the feature dimension because we are classifying our dataset into a lower number of classes than the pretrained model. Finally, we add a dense layer with 3 nodes with a sigmoid activation function so that each node provides a probability value for one class label.

As shown in Figure 2, we retrain the CNN model using our streetview dataset. The input to the CNN model is a set of 224x224 streetview images. After retraining, we extract the output of dense layer with 250 nodes to get a feature vector of 250 dimensions for each image in the sequence. We then create a set of feature sequences to feed into the LSTM model.

#### 4.2 Model Spatial Linear Pattern with LSTM

In order to model spatial linear (sequential) structure along a road network path, we used the LSTM model on a sequence of image features extracted by the CNN model. LSTM is a type of recurrent neural networks that uses gating functions to avoid the exploding and vanishing gradient issues. The gate function can help a model to memorize the state of previous units in a sequence. Such recurrent

structure is well-suited to capture the spatial autocorrelation effect across consecutive images. According to the first law of geography: "everything is related to everything else, but near things are more related than distant things." For example, concrete barriers are often very long spanning over several miles. Metal crash barriers, in contrast, have a shorter spatial scale within a few hundred meters.

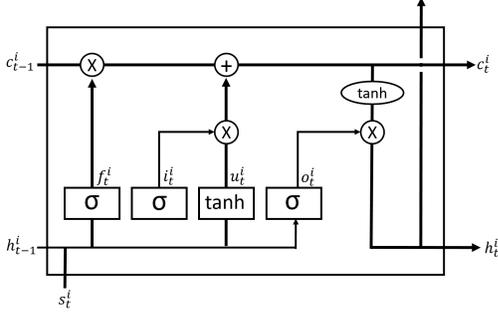


Figure 3: LSTM Unit

LSTM models a sequential structure by maintaining a sequence of memory cells ( $c_t$  with  $t$  as the spatial location index). In each spatial location  $t$ , LSTM takes an input feature  $s_t^i$ , hidden state  $h_{t-1}^i$  and cell state  $c_{t-1}^i$ . Figure 3 shows a LSTM unit with a cell state ( $c_t$ ) and three different gates: input gate, output gate and forget gate. The forget gate ( $f_t$ ) decides how much information from a previous cell unit is ignored before coming to the next cell. The input gate ( $i_t$ ) decides how much contribution an input feature vector makes to the current cell state. Finally, the output gate ( $o_t$ ) decides what the current LSTM unit is going to output (current cell state  $c_t$  and current hidden state  $h_t$ ) based on the cell state. The LSTM transaction equations are as follows,

$$\begin{aligned}
 f_t^i &= \sigma(W_f h_{t-1}^i + U_f s_t^i + b_f) \\
 i_t^i &= \sigma(W_i h_{t-1}^i + U_i s_t^i + b_i) \\
 o_t^i &= \sigma(W_o h_{t-1}^i + U_o s_t^i + b_o) \\
 u_t^i &= \tanh(W_u h_{t-1}^i + U_u s_t^i + b_u) \\
 c_t^i &= f_t * c_{t-1}^i + i_t^i * u_t^i \\
 h_t^i &= o_t * \tanh(c_t^i)
 \end{aligned} \tag{1}$$

where  $\sigma$  denotes the sigmoid activation function,  $\tanh$  is hyperbolic tangent function and  $*$  denotes element-wise product.  $W$  and  $U$  denote model parameters. As Figure 2(c) shows, our LSTM model consists of 4 hidden layers. The first layer is an LSTM layer with an output dimension of 100 units. The second layer is a 20% dropout layer. The third layer is a dense layer with 50 nodes. The last layer is a sigmoid transformation layer with 3 nodes, corresponding to the three independent class labels (rumble strips, concrete barrier, and metal crash barrier). This is different from the common softmax layer whose output node values sum into one because class labels are assumed to be exclusive to each other. We used the binary cross-entropy loss. To get final output labels for each image, we used a threshold of 0.5 on the sigmoid outputs.

## 5 EXPERIMENTAL EVALUATION

In this section, we compared our proposed method with baseline methods on two real world datasets in classification performance. Experiments were conducted on a Dell workstation with Intel(R) Xeon(R) CPU E5-2687w v4@3.00GHz, 64GB main memory, and a Nvidia Quadro K6000 GPU with 2880 cores and 12GB memory. We used Keras with Tensorflow as backend to run the deep learning models. Candidate classification methods included:

- **CNN only:** We used Inception-ResnetV2 CNN model on streetview images with three classes: rumble strips, concrete barriers and metal crash barriers. We added one more dense layer with 250 nodes and a *ReLU* activation function before the final sigmoid layer with 3 nodes.
- **CNN-DT:** We extracted output of second last layer (with 250 nodes) from our CNN only model (Inception-ResnetV2) as feature vectors and fed it into a Decision Tree (DT) model. We used the scikit-learn package in Python.
- **CNN-RF:** We extracted output of second last layer (with 250 nodes) from our CNN only model (Inception-ResnetV2) as feature vectors and fed it into a Random Forest (RF) model. We used the scikit-learn package in Python.
- **CNN-LSTM:** This is our proposed model to address the issue of multi-label classification using shared CNN-LSTM model for all three class labels together.

Unless specified otherwise, we used default parameters in open source tools in baseline methods.

### 5.1 Dataset Description

To evaluate the performance of the proposed model, we selected two different road segments to extract streetview imagery for training, validation, and testing. We selected a road segment in i-20 East for training and validation and a road segment in i-20 West for testing. We divided the road segments into an equally distanced set of geolocation coordinates. We set the distance interval of 20 meters. We then used Google Street View API to download the streetview images respective to each coordinate. We have three safety features classes: rumble strips (**RS**), concrete barriers (**CB**) and metal crash barriers (**MCB**). Table 1 shows the number of images and class distribution for training, validation and test dataset.

Table 1: Class Distribution

	Number of Images	Rumble Strips	Concrete Barriers	Metal Crash Barriers
Training Set	983	868	352	324
Validation Set	594	493	224	96
Test Set	950	857	354	279

To train our proposed CNN-LSTM model, we used the input vector length of 50 spatially continuous images for each sequence. We used a sliding window of 1 on training and validation data set to create training and validation sequences for CNN-LSTM model. We generated 883 training and 544 validation sequence.

## 5.2 Hyperparameter Settings

For our proposed model, we set the number of input vector in LSTM as 50, and the dimension of the hidden state of LSTM as 100. We set the learning rate of  $10^{-6}$  for both CNN and CNN-LSTM model. Both CNN and CNN-LSTM models were trained using Adam optimizer. The training batch size for CNN model was 32 and that for CNN-LSTM was 1.

## 5.3 Classification Performance

Figure 4 shows the CNN-LSTM model training performance. The training and validation loss are 0.06191 and 0.32520 respectively. We achieved training accuracy of around 0.98 and validation accuracy of 0.85. We set the probability threshold of 0.5. A safety feature probability value above the threshold indicates presence of the safety feature in the image. We compared the F-score of different candidate methods on precision, recall, and F-score. Results were summarized in Table 2. Most of the candidate method achieved over 0.85 average F-score. But, our proposed model outperformed all the baseline methods with an average F-score of 0.91. The average F-score for CNN with DT and RF is lower than CNN only. It may be because CNN-DT and CNN-RF takes the output of 2nd last layer (with 250 output dimension) and fits the models using it as the features. But the last layer in CNN only model, a dense layer with 3 nodes, have extra learnable parameters which can help CNN only model learn better.

Figures 5, 6 and 7 shows the prediction maps for three safety feature classes from CNN and CNN-LSTM models, together with the ground truth class map on the test path. We can observe that the CNN-LSTM model was able to reduce the isolated misclassified images from CNN. CNN only model may make some classification error on some images within a sequence due to not taking spatial dependency into consideration. Our proposed CNN-LSTM model can help to correct such isolated errors by incorporating spatial dependency in the learning process.

Figure 8, 9 and 10 shows samples from four consecutive streetview images for each safety feature class, where the CNN only model failed to correctly classify one or two images inbetween. For example, in Figure 8, the third image (c) was misclassified in the CNN model but correctly classified in the CNN-LSTM model due to incorporating the spatial sequential structure. Similarly results were shown in Figure 9 and Figure 10.

## 6 CONCLUSION

In this paper, we proposed CNN-LSTM based spatial classification model for mapping safety features along road networks. Our CNN-LSTM model can capture spatial linear structure between consecutive images along a road network path. Results on real world Google Streetview images collected in Alabama showed that our model outperforms several baseline methods.

In future work, we plan to conduct a case study on more streetview images over the entire road networks in Alabama. We will also investigate more general spatial network structure with graph-LSTM.

## ACKNOWLEDGMENTS

This work was supported by Alabama Transportation Institute.

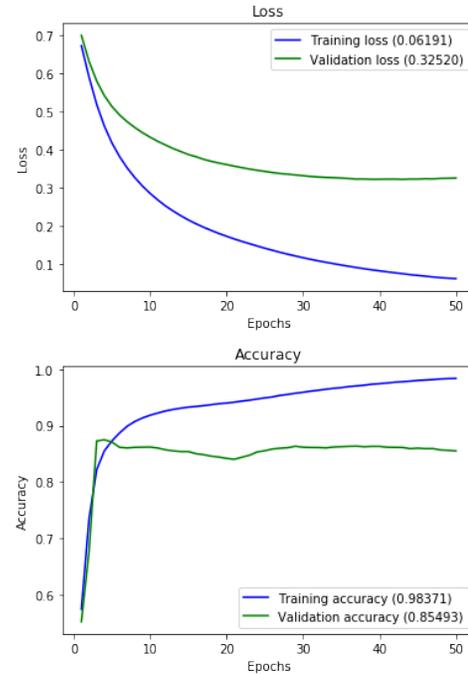


Figure 4: Training and Validation performance over 50 epochs

Table 2: Classification on Test Dataset

Classifiers	Class	Precision	Recall	F	Avg. F
CNN-DT	RS	0.95	0.83	0.89	0.85
	CB	0.81	0.76	0.78	
	MCB	0.79	0.82	0.80	
CNN-RF	RS	0.94	0.87	0.90	0.87
	CB	0.78	0.82	0.80	
	MCB	0.89	0.86	0.87	
CNN only	RS	0.94	0.95	0.95	0.89
	CB	0.74	0.85	0.79	
	MCB	0.91	0.78	0.84	
CNN-LSTM	RS	0.43	0.98	0.96	0.91
	CB	0.74	0.91	0.82	
	MCB	0.90	0.83	0.86	

## REFERENCES

- [1] Vahid Balali, Elizabeth Depwe, and Mani Golparvar-Fard. 2015. Multi-class traffic sign detection and classification using google street view images. In *Transportation Research Board 94th Annual Meeting, Transportation Research Board, Washington, DC*.
- [2] Vahid Balali, Armin Ashouri Rad, and Mani Golparvar-Fard. 2015. Detection, classification, and mapping of US traffic signs using google street view images for roadway inventory management. *Visualization in Engineering* 3, 1 (2015), 15.
- [3] Michael R Bambach, RJ Mitchell, and Raphael H Grzebieta. 2013. The protective effect of roadside barriers for motorcyclists. *Traffic injury prevention* 14, 7 (2013), 756–765.
- [4] James E Bryden and Jan S Fortuniewicz. 1986. Traffic barrier performance related to vehicle size and type. *Transportation Research Record* 1065 (1986), 69–78.
- [5] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. 2009. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*. Ieee, 248–255.

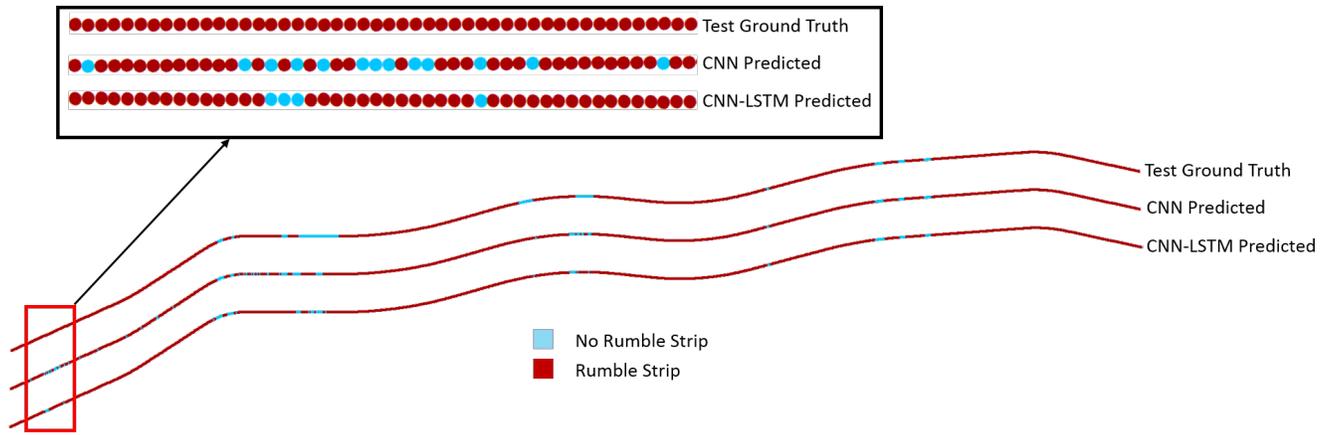


Figure 5: Prediction map on test road segment for rumble strips

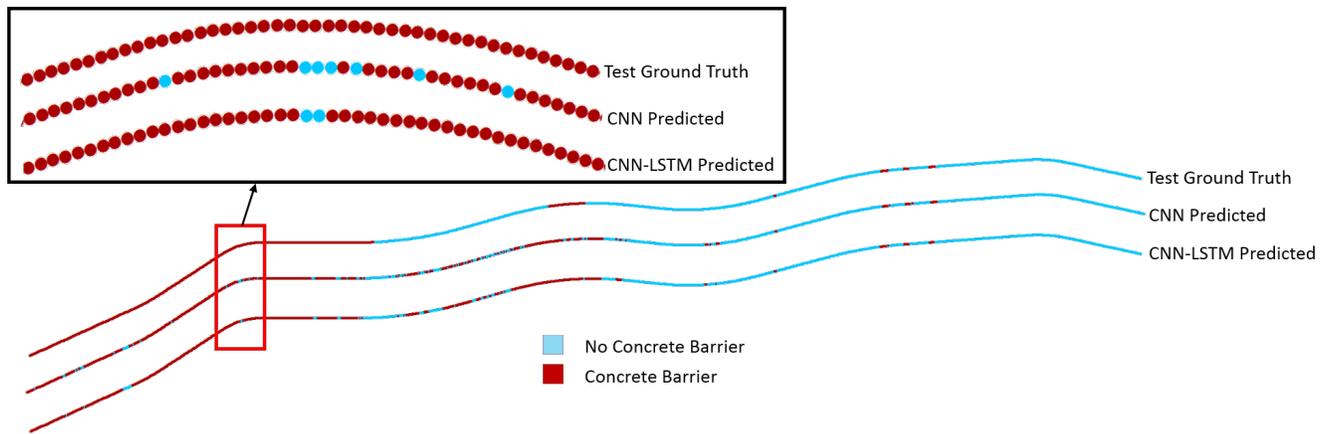


Figure 6: Prediction map on test road segment for concrete barriers

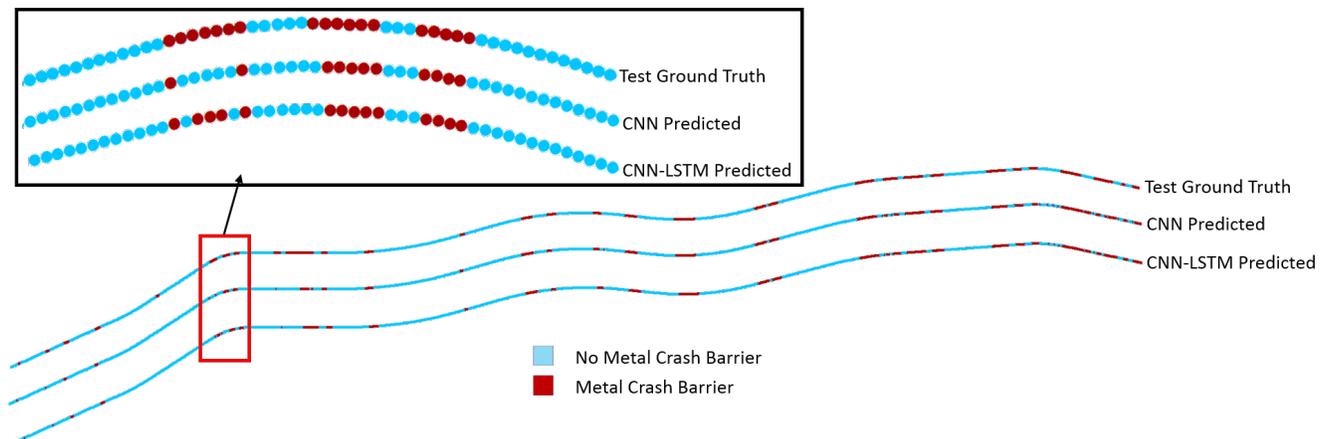


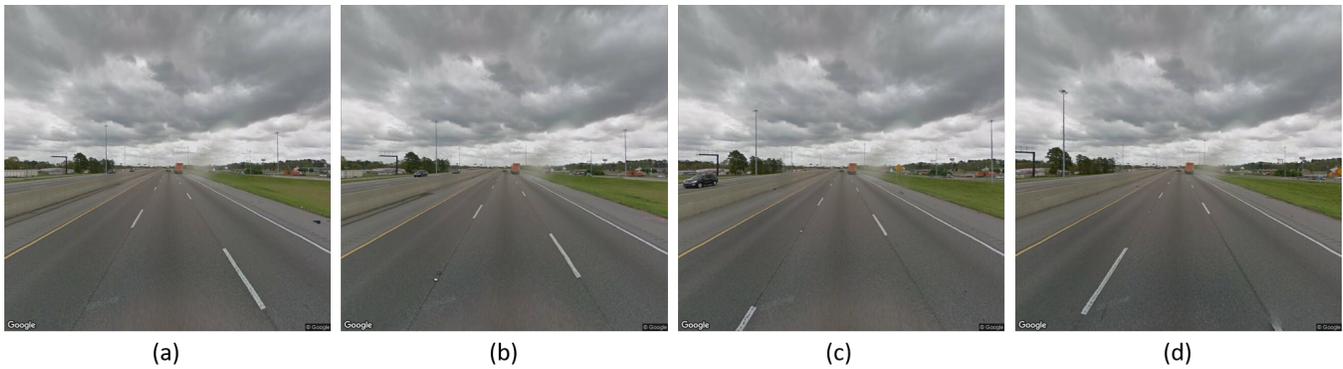
Figure 7: Prediction map on test road segment for metal crash barriers

[6] Timnit Gebru, Jonathan Krause, Yilun Wang, Duyun Chen, Jia Deng, Erez Lieberman Aiden, and Li Fei-Fei. 2017. Using deep learning and Google Street View to estimate the demographic makeup of neighborhoods across the United States.

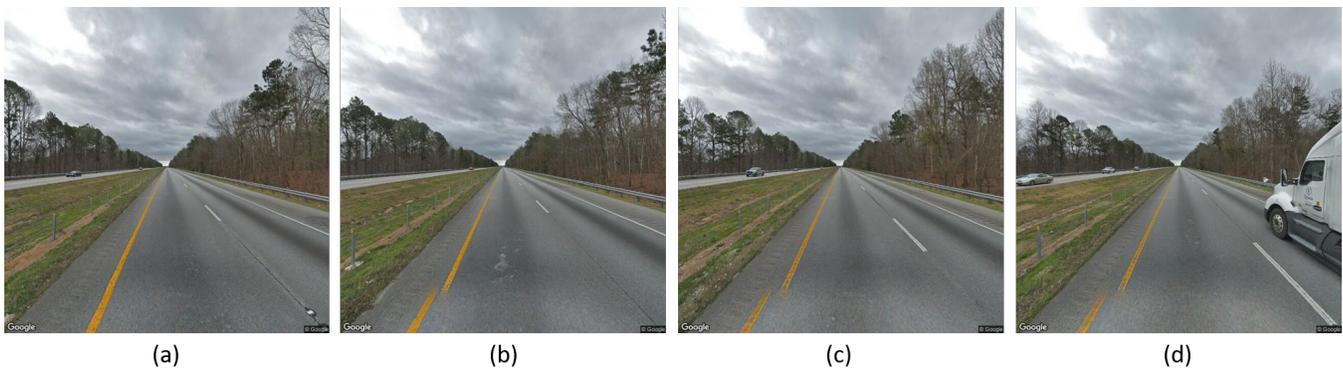
[7] Raphael Grzebieta, Mike Bambach, and Andrew McIntosh. 2013. Motorcyclist Impacts into Roadside Barriers: Is the European Crash Test Standard Comprehensive



**Figure 8: Four consecutive streetview images with rumble strip class in ground truth. Image (c) was misclassified as "not rumble strips" in CNN but were correctly classified in CNN-LSTM.**



**Figure 9: Four consecutive streetview images with concrete barrier class in ground truth. Images (b) and (c) were misclassified as "not concrete barrier" in CNN but were correctly classified in CNN-LSTM.**



**Figure 10: Four consecutive streetview images with metal crash barrier class in ground truth. Image (b) was misclassified as "not metal crash barrier" in CNN but were correctly classified in CNN-LSTM.**

Enough? *Transportation Research Record* 2377, 1 (2013), 84–91.  
 [8] Gayle SW Hagler, Wei Tang, Matthew J Freeman, David K Heist, Steven G Perry, and Alan F Vette. 2011. Model evaluation of roadside barrier impact on near-road air pollution. *Atmospheric Environment* 45, 15 (2011), 2522–2530.  
 [9] Hussein H Jama, Raphael H Grzebieta, Rena Friswell, and Andrew S McIntosh. 2011. Characteristics of fatal motorcycle crashes into roadside safety barriers in

Australia and New Zealand. *Accident Analysis & Prevention* 43, 3 (2011), 652–660.  
 [10] Zhe Jiang. 2018. A survey on spatial prediction methods. *IEEE Transactions on Knowledge and Data Engineering* (2018).  
 [11] Zhe Jiang, Michael Evans, Dev Oliver, and Shashi Shekhar. 2016. Identifying K Primary Corridors from urban bicycle GPS trajectories on a road network. *Information Systems* 57 (2016), 142–159.

- [12] Hawzheen Karim, Rolf Magnusson, and Mats Wiklund. 2012. Assessment of Injury Rates Associated with Road Barrier Collision. *Procedia-social and behavioral sciences* 48 (2012), 52–63.
- [13] Driver Knowledge. 2013. Accident Statistics. <https://www.driverknowledge.com/car-accident-statistics/>
- [14] Xiaojiang Li, Chuanrong Zhang, Weidong Li, Robert Ricard, Qingyan Meng, and Weixing Zhang. 2015. Assessing street-level urban greenery using Google Street View and a modified green view index. *Urban Forestry & Urban Greening* 14, 3 (2015), 675–685.
- [15] Aibek Musaev, Zhe Jiang, Steven Jones, Pezhman Sheinidashtegol, and Mirbek Dzhumaliev. 2018. Detection of Damage and Failure Events of Road Infrastructure Using Social Media. In *International Conference on Web Services*. Springer, 134–148.
- [16] Congressional Budget Office. 2018. Public Spending on Transportation and Water Infrastructure, 1956 to 2017. <https://www.cbo.gov/system/files/2018-10/54539-Infrastructure.pdf>
- [17] Benjamin Romano and Zhe Jiang. 2017. Visualizing traffic accident hotspots based on spatial-temporal network kernel density estimation. In *Proceedings of the 25th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*. ACM, 98.
- [18] Carlos Roque and João Lourenço Cardoso. 2013. Observations on the relationship between European standards for safety barrier impact severity and the degree of injury sustained. *IATSS research* 37, 1 (2013), 21–29.
- [19] Arpan Man Sainju, Danial Aghajarian, Zhe Jiang, and Sushil K Prasad. 2018. Parallel grid-based colocation mining algorithms on GPUs for big spatial event data. *IEEE Transactions on Big Data* (2018).
- [20] Arpan Man Sainju and Zhe Jiang. 2017. Grid-based colocation mining algorithms on gpu for big spatial event data: A summary of results. In *International Symposium on Spatial and Temporal Databases*. Springer, 263–280.
- [21] Jennifer D Schmidt, Ronald K Faller, Dean L Sicking, John D Reid, Karla A Lechtenberg, Robert W Bielenberg, Scott K Rosenbaugh, Jim C Holloway, et al. 2013. *Development of a new energy-absorbing roadside/median barrier system with restorable elastomer cartridges*. Technical Report. Nebraska. Dept. of Roads.
- [22] Nico Schulte, Michelle Snyder, Vlad Isakov, David Heist, and Akula Venkatram. 2014. Effects of solid barriers on dispersion of roadway emissions. *Atmospheric environment* 97 (2014), 286–295.
- [23] Shashi Shekhar, Zhe Jiang, Reem Ali, Emre Eftelioglu, Xun Tang, Venkata Gunturi, and Xun Zhou. 2015. Spatiotemporal data mining: a computational perspective. *ISPRS International Journal of Geo-Information* 4, 4 (2015), 2306–2338.
- [24] Christian Szegegy, Sergey Ioffe, Vincent Vanhoucke, and Alexander A Alemi. 2017. Inception-v4, inception-resnet and the impact of residual connections on learning. In *Thirty-First AAAI Conference on Artificial Intelligence*.
- [25] Zheming Tong, Richard W Baldauf, Vlad Isakov, Parikshit Deshmukh, and K Max Zhang. 2016. Roadside vegetation barrier designs to mitigate near-road air pollution impacts. *Science of the Total Environment* 541 (2016), 920–927.
- [26] Victor JD Tsai, Jyun-Han Chen, and Hsun-Sheng Huang. 2016. Traffic Sign Inventory from Google Street View Images. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences* 41 (2016), 243–246.
- [27] Carlos M Vieira, Henrique A Almeida, Irene S Ferreira, Joel O Vasco, Paulo J Bártolo, Rui B Ruben, and Sérgio P Santos. 2008. Development of an impact absorber for roadside barriers. In *Proceedings of the 7th LS-DYNA Forum*.
- [28] Yuankai Wu and Huachun Tan. 2016. Short-term traffic flow forecasting with spatial-temporal correlation in a hybrid deep learning framework. *arXiv preprint arXiv:1612.01022* (2016).
- [29] Huaxiu Yao, Xianfeng Tang, Hua Wei, Guanjie Zheng, Yanwei Yu, and Zhenhui Li. 2018. Modeling spatial-temporal dynamics for traffic prediction. *arXiv preprint arXiv:1803.01254* (2018).
- [30] Huaxiu Yao, Fei Wu, Jintao Ke, Xianfeng Tang, Yitian Jia, Siyu Lu, Pinghua Gong, Jieping Ye, and Zhenhui Li. 2018. Deep multi-view spatial-temporal network for taxi demand prediction. In *Thirty-Second AAAI Conference on Artificial Intelligence*.
- [31] Zhuoning Yuan, Xun Zhou, and Tianbao Yang. 2018. Hetero-ConvLSTM: A deep learning approach to traffic accident prediction on heterogeneous spatio-temporal data. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. ACM, 984–992.
- [32] Zhuoning Yuan, Xun Zhou, Tianbao Yang, James Tamerius, and Ricardo Mantilla. 2017. Predicting traffic accidents through heterogeneous urban data: A case study. In *Proceedings of the 6th International Workshop on Urban Computing (UrbComp 2017)*, Halifax, NS, Canada, Vol. 14.
- [33] Shanghang Zhang, Guanhang Wu, Joao P. Costeira, and Jose M. F. Moura. 2017. FCN-rLSTM: Deep Spatio-Temporal Neural Networks for Vehicle Counting in City Cameras. In *The IEEE International Conference on Computer Vision (ICCV)*.
- [34] Yaotian Zou, Andrew P Tarko, Erdong Chen, and Mario A Romero. 2014. Effectiveness of cable barriers, guardrails, and concrete barrier walls in reducing the risk of injury. *Accident Analysis & Prevention* 72 (2014), 55–65.