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

ARPAN MAN SAINJU and ZHE JIANG, University of Alabama

15

Each year, an average of around 6 million car accidents occur in the United States. 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 in 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 article, we propose a deep learning approach to automatically map road safety features from streetview imagery. Unlike existing convolutional neural networks that classify each image individually, we propose to further add a 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.

CCS Concepts: • **Information systems** → **Geographic information systems**; **Machine Learning**; • **Computing methodologies** → **Deep Learning**;

Additional Key Words and Phrases: Spatial classification, road network classification, spatial dependency, deep learning

ACM Reference format:

Arpan Man Sainju and Zhe Jiang. 2020. Mapping Road Safety Features from Streetview Imagery: A Deep Learning Approach. *ACM/IMS Trans. Data Sci.* 1, 3, Article 15 (September 2020), 20 pages.
<https://doi.org/10.1145/3362069>

1 INTRODUCTION

Every year, an average of around 6 million car accidents occur in the United States [22]. Traffic safety has long been an important societal issue. 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 the winter season, vehicles can become more difficult to control on icy and slippery road surfaces, particularly when the vehicle is traveling at a high speed. Barriers on the roadside can act as a safety precaution in such cases. Other safety features

This material is based upon work supported by the National Science Foundation IIS-1850546 and the Alabama Transportation Institute.

Authors' addresses: A. M. Sainju (corresponding author) and Z. Jiang, Department of Computer Science, The University of Alabama, Box 870290, Tuscaloosa, AL 35487; emails: asainju@crimson.ua.edu, zjiang@cs.ua.edu.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

© 2020 Copyright held by the owner/author(s). Publication rights licensed to ACM.

2577-3224/2020/09-ART15 \$15.00

<https://doi.org/10.1145/3362069>

Fig. 1. Three common classes of road safety features on Google Street View Imagery. (a) Rumble strips. (b) Concrete barrier. (c) Metal crash barrier.

such as rumble strips help alert inattentive drivers who are deviating from their lanes. Figure 1 shows three different common types of road safety features: rumble strips, concrete barrier, and metal crash barrier. Federal, state, and local governments spent several \$100 billion each year on transportation infrastructure development and maintenance [27]. 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 Street View 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 becomes prohibitive.

The focus of this article is to develop a deep learning algorithm that can automatically map road safety features from streetview imagery. The results can be used by 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 barriers are often very long (e.g., miles). In contrast, metal crash barriers are much shorter (e.g., 100 m). 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 a 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 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 approaches outperform several baseline methods in classification performance.

In summary, the contributions of this work are listed as follows:

- To the best of our knowledge, we are the first to explore a deep learning approach on Google Street View imagery for road safety feature mapping.
- We propose to use integrated deep learning models that combine CNN and LSTM. The integrated models can utilize not only the content of individual images but also the spatial sequential structure between images.
- We compare our approaches with several baseline methods on two real-world streetview imagery datasets collected in Alabama. We achieve 3% and 5% improvement in F-score over the best baseline method on two different test datasets.
- We perform a case study of mapping road safety features with 69,500 streetview imageries over all of the major highways in the entire state of Alabama.

The outline of the article is as follows. Section 2 discusses some of the related works. Section 3 formally defines the problem. Section 4 introduces the approaches. Section 5 summarizes the results of our experimental evaluation on two real-world datasets and discusses the case study of mapping road safety features over all of the major highways in Alabama. Section 6 concludes the article 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) [21, 29, 47]. For example, Bambach et al. [3], Grzebieta et al. [8], and Jama et al. [10] quantify the protective effect of barriers with regard to motorcyclist injury. Bryden and Fortuniewicz [4] analyze the performance of roadside barriers related to vehicle size and type. Vieira et al. [38] study how to increase the effectiveness of the roadside barriers in safety protection. For example, studies found that concrete barriers can hold a high-energy truck crash but can also cause more fatalities [48]. Some recent work focuses on developing an energy-absorbing barrier [32]. Besides the protective effect, other studies on roadside barriers focus on the impact on mitigating near-road air pollution [36]. The study on the effect of solid barriers on dispersion of roadway emissions in the work of Hagler et al. [9] and Schulte et al. [33] shows that roadside barriers are 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 [25, 28, 30, 31]. Jiang et al. [12] propose efficient algorithms to identify primary corridors from cyclists' GPS trajectories on urban road networks to study riding behaviors for safety issues. Musaev et al. [26] show techniques to detect coarse-scale hotspots of road failure events through geo-tagged tweets from social media.

Recently, researchers have used Google Street View imagery along the road network for traffic sign detection for roadway inventory management [1, 2, 37]. Other works use streetview imagery to estimate the demographic makeup of neighborhoods [6], to assess street-level greenness in

an urban area [23]. 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 Spatiotemporal Data

The work in this article generally falls into the area of spatial classification or prediction [11, 13–15, 17–20, 40] in spatiotemporal data mining [16, 34]. In recent years, deep learning techniques have shown great growth in this area. One common approach is to integrate deep CNNs with recurrent neural networks such as LSTM. The CNN component can be used to model spatial dependency structure in one temporal snapshot, whereas the LSTM component can be used to model temporal dynamics between different snapshots. For example, Zhang et al. [46] use fully convolutional networks with LSTM to estimate vehicle count maps based on city cameras. Yao et al. [42] use a CNN-LSTM model together with multi-view learning to predict taxi demand. Wu and Tan [39] use 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. In another work, Yao et al. [41] address 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. Yuan et al. [43, 44] research better traffic accident prediction to improve transportation and public safety. In these existing works, the LSTM model is often used to model temporal dynamics between multiple spatial snapshots. The difference from our work in this article is that we use LSTM to capture linear spatial sequential structure between consecutive images along a road network path. There is also other work based on spatiotemporal residual networks for traffic flow prediction [45].

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 a camera so that each image can be geo-referenced based on the GPS timestamp. In this work, we used Google Street View API to select imagery at a regular spatial interval of 20 m.

Road safety feature. A *road safety feature* is defined as the measure or infrastructure placed on a road to improve safety. We consider the three most common safety features: rumble strips, concrete barrier, and metal crash barrier. Figure 1 shows examples of the three safety features from Google Street View 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 that can alert 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 the opposing direction is flowing in close proximity due to lack of space.

Fig. 2. Overall framework of our deep learning approach.

- *Metal crash barrier*: Metal crash barrier (Figure 1(c)), also known as a guardrail, is usually made from steel beams or rails. Metal barrier helps absorb the energy from a crashed vehicle and reduce damages to the vehicle and occupants. It can also act as a good visual guide at night for the drivers 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 PROPOSED APPROACH

In this section, we introduce our proposed deep learning approaches. Figure 2 illustrates the overall framework of our proposed models. The bottom component shows the data collection process. We sampled several spatial points along road network edges at an equal distance interval (e.g., 20 m) and then used Google Street View API to download geo-referenced imagery at those point locations. We fixed the distance interval of 20 m 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 a short extent barrier such as metal crash barrier. Although a lower distance interval can provide a 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 a retrained CNN model to extract low-dimensional features from

individual images. The last component is the LSTM layer. In contrast to existing works, our LSTM does not capture temporal dynamics between different spatial snapshots but represents a spatial sequential pattern between consecutive imagery along road network edges.

4.1 Extracting Image Features with CNNs

The CNN was developed mainly for image classification. In addition, a CNN introduces the concept of parameter sharing, which allows the model to learn fewer parameters in comparison to a regular neural network. Similar to regular neural networks, a CNN also consists of a sequence of layers. We briefly describe each layer of a CNN in the following.

The 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. The convolutional layer transforms the input using convolution operation. A convolution operation is element-wise multiplication of a pixel and its neighborhood pixel color value (RGB) by a matrix. It is also known as a convolution filter. Different filters are used to convolve around all pixels in an image. Filters like the 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 but 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 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. Thus, it can be written as $f(x) = \max(0, x)$. The ReLU activation function is computationally less expensive because there is no complicated math, which can reduce the model training time. The function of the pooling layer is to reduce the spatial size of the input. It is also known as the downsampling layer. The pooling layer can reduce the number of parameters and computation in the network. It applies a filter (usually of size 2×2) to the input volume. Pooling filters can be based on different operations such as max, min, or average. The most common one is the max filter, which extracts the max value from the filter region. The fully connected (dense) layer takes an input volume (output of activation function) and outputs an N -dimensional vector. Similar to regular neural networks, neurons in a fully connected layer have full connections to all activation in the previous layer.

For our proposed models, we use the current state-of-the-art Inception-ResNetv2 [35] model to extract features from individual streetview images. We use the Keras implementation of Inception-ResNetv2 pre-trained on an ImageNet [5] dataset with 1,000 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 a few hops away. Residual connections are important for very deep architecture. When deeper networks start 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.

One issue is that the existing CNN models are not pre-trained on the road safety feature classes we defined. To address this issue, we retrain the Inception-ResNetv2 model. Specifically, we first remove the original final dense layer with the softmax function on 1,000 classes. Since we only have three classes (i.e., rumble strips, concrete barrier, and metal crash barrier), we replace the previous dense layer with a softmax function on 1,000 classes with a dense layer with a sigmoid function on three classes. The reason we use a sigmoid function instead of a softmax function is that our class labels are non-exclusive. In other words, each image may contain multiple road safety features (e.g., both rumble strips and metal crash barriers). We also add an additional

Fig. 3. LSTM unit.

dense (fully connected) layer with 250 nodes before the final sigmoid layer (and after the last average pooling layer with 1,536 nodes) to reduce the dimension of extracted image features (250 vs. 1536). This is important because we need to input the extracted image features from CNN to LSTM cells, and we are classifying images into a smaller number of classes than the original CNN model. As shown in Figure 2, we retrain the CNN model using our streetview dataset. The input to the CNN model is a set of 224×224 streetview images. After retraining, we extract the output of a 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 Modeling a Spatial Linear Pattern with LSTM

To model a 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 network that uses gating functions to avoid the exploding and vanishing gradient issues. The gate function can help a model memorize the state of previous units in a sequence. 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} , and cell state c_{t-1} . Figure 3 shows an 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 σ denotes the sigmoid activation function, $\tanh$ is hyperbolic tangent function, and the asterisk (*) denotes the element-wise product. W and U denote model parameters. As Figure 2(c) shows, our LSTM model consists of four 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.

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. LSTM has the strength of learning both short- and long-range spatial dependency. The scale of a spatial sequential structure can be controlled by the forget gates learned in the LSTM model. For example, for long rumble strips, the forget gates of individual cells would be largely open (receiving memory states from previous images far away due to longer spatial dependency). However, for shorter metal crash barriers that may cover only a few images, the forget gates in LSTM cells for corresponding images should be largely closed without receiving information from images far away.

4.3 Multi-Label Classification

Our problem is a multi-label classification problem because each image may contain multiple types of road safety features at the same time. To model multi-label classification, we propose two approaches. The first approach involves training a shared CNN-LSTM model for all three class labels together. The second approach involves training separate CNN-LSTM models for each class label independently.

In the first approach, the last layer in the model is a sigmoid transformation layer with three 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 of each other. We use binary cross-entropy loss. To get the final output labels for each image, we use a threshold of 0.5 on the sigmoid outputs.

In the second approach, we transform the multi-label problem into multiple single-label problems. We learn three independent models corresponding to each class label. For instance, the independent model for metal crash barriers will only classify the image based on the presence of a metal crash barrier, and, likewise, independent models for concrete barrier and rumble strips. In this design decision, the last layer is a sigmoid transformation layer with only one node corresponding to either one of the three independent class labels. We use binary cross-entropy loss. However, it would also be fine to use the softmax activation function with categorical cross-entropy loss.

5 EXPERIMENTAL EVALUATION

In this section, we compare our proposed method with baseline methods on two real-world datasets in classification performance. Experiments were conducted on a Dell workstation with Intel Xeon CPU E5-2687w v4@3.00 GHz with 64 GB of main memory and an Nvidia Quadro K6000 GPU with 2,880 cores and 12 GB of memory. We used Keras with TensorFlow as the backend to run the deep learning models. Candidate classification methods included the following:

- **CNN only:** We used an 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 three nodes.
- **CNN-DT:** We extracted the output of the 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 the output of the 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-sharedLSTM:** This is our proposed model to address the issue of multi-label classification using a shared CNN-LSTM model for all three class labels together.
- **CNN-separateLSTM:** This is our proposed model to address the issue of multi-label classification using separate single-label independent CNN-LSTM models corresponding to each class label.

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

Evaluation metrics. To evaluate the candidate classification methods, we used precision, recall, and F-score. We computed the precision, recall, and F-score for all tclass labels. Finally, we computed the weighted average F-score for all candidate classification methods. To calculate the weighted average, we used Equation (2):

$$Avg.F = F_{RS} \frac{RS_n}{RS_n + MCB_n + CB_n} + F_{MCB} \frac{MCB_n}{RS_n + MCB_n + CB_n} + F_{CB} \frac{CB_n}{RS_n + MCB_n + CB_n}, \quad (2)$$

where F_{RS} , F_{MCB} , and F_{CB} refer to the F-score of rumble strips, metal crash barriers, and concrete barriers class labels, respectively. Similarly, RS_n , MCB_n , and CB_n refer to the number of images containing class labels: rumble strips, metal crash barriers, and concrete barriers, respectively.

5.1 Dataset Description

To evaluate the performance of our proposed models, first we randomly selected 3,745 labeled isolated streetview images across the state of Alabama for pre-training. We used this dataset to pre-train the CNN in all baseline and proposed methods. Next, we selected different road segments within the I-20 highway in Alabama to extract spatially continuous streetview images for training, validation, and test datasets. The images in extracted datasets are different from the images in the pre-training dataset. For training and validation, we selected the road segment from 33°37'08.4"N 85°42'28.2"W to 33°35'06.5"N 86°03'40.6"W of I-20 East near Oxford, Alabama. We selected two test datasets. Our first test dataset was based on the road segment from 33°35'09.6"N 85°52'32.1"W to 33°36'56.4"N 85°41'25.2"W of I-20 West near Oxford, Alabama, which is closer to the training and validation datasets. Our second dataset was based on the road segment from 33°08'09.1"N 87°38'05.6"W to 33°11'13.2"N 87°19'59.3"W of I-20 West near Tuscaloosa, Alabama, which is far from the training and validation datasets. We refer to the first test dataset as Test Set *Oxford* and the second test dataset as Test Set *Tuscaloosa* for the rest of the article. We used the same training and validation dataset for both of the test datasets.

We divided the road segments into an equally distanced set of geo-location coordinates. We set a distance interval of 20 m. We then used Google Street View API to download the streetview images respective to each coordinate. We have three safety feature classes: rumble strips (RS), metal crash barriers (MCB), and concrete barriers (CB). Table 1 shows the number of images and class distribution for training, validation, and test datasets.

5.2 Hyperparameter Settings

For our proposed models, there are several design decisions to be made, including the input vector length in LSTM, dimension of the hidden state of LSTM, learning rate, dropout, optimization function, and training batch size. To train our proposed CNN-LSTM models, we used the input vector length of 50 spatially continuous images on each sequence. We used a sliding window of 1 on training and validation datasets to create training and validation sequences for our proposed CNN-LSTM models. We generated 883 training and 544 validation sequences. We set the input

Table 1. Class Distribution

	Number of Images	Rumble Strips	Metal Crash Barriers	Concrete Barriers
Pre-Training Set	3,745	1,882	1,149	1,632
Training Set	983	868	324	352
Validation Set	594	493	96	224
Test Set_ <i>Oxford</i>	950	857	279	354
Test Set_ <i>Tuscaloosa</i>	1,350	879	403	784

Fig. 4. Training and validation performance of CNN-sharedLSTM over 50 epochs for all three class labels.

vector length in LSTM as 50 and the dimension of the hidden state of LSTM as 100. The input vector length of 50 refers to streetview images covering 1,000 m (20-m separation between images). For the learning rate, we first started with a high learning rate of 10^{-3} and observed oscillating training and validation loss curves. We then gradually decreased the learning rate and obtained more stable curves with a learning rate of 10^{-6} for CNN and CNN-LSTM models. Next, we varied the dropout value from 0.2 to 0.6. We observed that with the higher value of dropout, the convergence of training and validation loss was slower and required a higher number of epochs. Thus, we tuned the dropout to the optimal value of 0.2. We trained CNN and CNN-LSTM models using the Adam optimizer. The training batch size for the CNN model was 32 and that for CNN-LSTM was 1.

5.3 Classification Performance

Figure 4 shows the training performance of the CNN-sharedLSTM model. Similarly, Figures 5, 6, and 7 show the training performance of the CNN-separateLSTM model for the rumble strips, metal crash barriers, and concrete barriers class labels, respectively. The training and validation loss for CNN-sharedLSTM are 0.06 and 0.32, respectively. In the case of CNN-separateLSTM, the training and validation loss for the rumble strips class label are 0.02 and 0.08, respectively. We can observe that the gap between training and validation loss for rumble strips in the CNN-separateLSTM model is lower than that of the CNN-sharedLSTM model. The training and validation loss in CNN-separateLSTM for the metal crash barriers class label are 0.05 and 0.28, respectively, which also shows a similar trend, because the CNN-separateLSTM model can better learn the spatial scale of rumble strips and metal crash barriers separately. Rumble strips have a very high spatial scale in comparison to the metal crash barriers. Furthermore, rumble strips and metal crash barriers

Fig. 5. Training and validation performance of CNN-separateLSTM for rumble strips over 50 epochs.

Fig. 6. Training and validation performance of CNN-separateLSTM for metal crash barriers over 50 epochs.

Fig. 7. Training and validation performance of CNN-separateLSTM for concrete barriers over 50 epochs.

are mostly easy to identify in the streetview images. Finally, the training and validation loss for the concrete barriers class label are 0.02 and 0.48, respectively. As shown in in Figure 7, the gap between the training and validation curve is high and indicates overfitting, which is likely because the concrete barriers are very diverse and hard to learn. We observe that the texture of concrete barriers in some regions are very similar to the texture of the roads.

We compare different candidate methods on precision, recall, and F-score. To obtain the predicted class label, we set the probability threshold of 0.5. A road safety feature probability value above the threshold indicates the presence of the road safety feature in the image. Results are summarized in Tables 2 and 3 for Test Set_*Oxford* and Test Set_*Tuscaloosa*, respectively.

Table 2. Classification on Test Set_*Oxford*

Classifiers	Class	Precision	Recall	F	Avg. F
CNN-DT	RS	0.95	0.83	0.89	0.85
	MCB	0.79	0.82	0.80	
	CB	0.81	0.76	0.78	
CNN-RF	RS	0.94	0.87	0.90	0.87
	MCB	0.89	0.86	0.87	
	CB	0.78	0.82	0.80	
CNN-only	RS	0.94	0.95	0.95	0.89
	MCB	0.91	0.78	0.84	
	CB	0.74	0.85	0.79	
CNN-sharedLSTM	RS	0.94	0.98	0.96	0.91
	MCB	0.90	0.83	0.86	
	CB	0.74	0.91	0.82	
CNN-separateLSTM	RS	0.95	0.96	0.96	0.92
	MCB	0.93	0.83	0.88	
	CB	0.77	0.93	0.84	

Table 3. Classification on Test Set_*Tuscaloosa*

Classifiers	Class	Precision	Recall	F	Avg. F
CNN-DT	RS	0.85	0.84	0.85	0.75
	MCB	0.50	0.86	0.63	
	CB	0.94	0.54	0.69	
CNN-RF	RS	0.88	0.87	0.88	0.77
	MCB	0.58	0.77	0.66	
	CB	0.93	0.58	0.71	
CNN-only	RS	0.87	0.97	0.92	0.78
	MCB	0.78	0.73	0.75	
	CB	0.89	0.49	0.63	
CNN-sharedLSTM	RS	0.85	0.98	0.91	0.79
	MCB	0.68	0.84	0.75	
	CB	0.93	0.53	0.68	
CNN-separateLSTM	RS	0.88	0.96	0.92	0.83
	MCB	0.74	0.80	0.77	
	CB	0.96	0.61	0.76	

For Test Set_*Oxford* in Table 2, the average F-score of CNN-DT, CNN-RF, and CNN-only models are 0.85, 0.87, and 0.89, respectively. We can observe that the average F-score for CNN with DT and RF is lower than CNN-only, because CNN-DT and CNN-RF take the output of the second last layer (with 250 output dimension) of the CNN-only model as the features and fits the models. But the last layer in the CNN-only model, a dense layer with three nodes, has extra learnable parameters that can help the CNN-only model train better. Next, we also observe that both of our proposed models (CNN-sharedLSTM and CNN-separateLSTM) perform better than the CNN-only model with an average F-score of 0.91 and 0.92, respectively. The CNN-only model may fail to correctly identify some in-between images in the test image sequence. Our proposed models can correct these errors

Fig. 8. Prediction maps on Test Set_Oxford for rumble strips (a), metal crash barriers (b), and concrete barriers (c), respectively.

by incorporating spatial dependency in the learning process using an LSTM network. In addition, we can observe that our CNN-separateLSTM performs better than CNN-sharedLSTM, because in the case of CNN-separateLSTM, the independent models for each class label can better learn the spatial scale of different class labels separately. For example, metal crash barriers have an average length of a few hundred meters, whereas other road safety features such as rumble strips may have an average length of a few kilometers. Furthermore, for CNN-separateLSTM, we can observe a very high F-score of 0.96 on the rumble strips class, which is consistent with the learning curve in Figure 5 with a small gap between the training and validation loss. In the case of metal crash barriers, we observe an F-score of 0.88, which is also consistent with the learning curve shown in

Fig. 9. Prediction maps on Test Set_*Tuscaloosa* for rumble strips (a), metal crash barriers (b), and concrete barriers (c), respectively.

Figure 6 with a medium gap between the training and validation loss. Finally, for concrete barriers, we observe the lowest F-score of 0.84, which also is consistent with the learning curve shown in Figure 7 with a high gap between the training and validation loss.

For Test Set_*Tuscaloosa* in Table 3, we can also observe similar trends as discussed earlier. However, the overall performance of all candidate methods is lower. The average F-score of CNN-DT, CNN-RF, CNN-only, CNN-sharedLSTM, and CNN-separateLSTM are 0.75, 0.77, 0.78, 0.79, and 0.83, respectively. This is likely because the Test Set_*Tuscaloosa* image location lies far from the training and validation region. Thus, the training and validation images might not be very similar to images in Test Set_*Tuscaloosa*. This can be easily improved by introducing some representative images closer to Test Set_*Tuscaloosa* in training and validation data.

Fig. 10. Classification map for the rumble strips class label using the CNN-separateLSTM model.

We also visualize the predicted class label on the map based on all predicted results summarized in Tables 2 and 3. Figure 8 shows the ground truth and prediction maps for three road safety feature classes based on CNN-only, CNN-sharedLSTM, and CNN-separateLSTM models on Test Set *Oxford*. Similarly, Figure 9 shows ground truth and prediction maps on Test Set *Tuscaloosa*. We can observe that different road safety features have different spatial scale. From Figures 8(a) and 9(a), we can observe that the ground truth spatial scale of rumble strips is usually very long. From Figures 8(b) and 9(b), we can observe that the ground truth spatial scale of metal crash barriers is usually very short. Finally, from Figures 8(c) and 9(c), we can observe that the ground truth spatial scale of concrete barriers can vary from short to long. Next, in all of the prediction maps, we can observe that CNN-only-based predictions contain some isolated errors. However, the CNN-sharedLSTM model was able to correct some of those isolated errors as highlighted in

Fig. 11. Classification map for the metal crash barriers class label using the CNN-separateLSTM model.

zoom-in sub-visualizations. Yet we observe the CNN-separateLSTM model to be more accurate, which is consistent with the summarized result in Tables 2 and 3. The CNN-sharedLSTM and CNN-separateLSTM models are able to generate a more accurate map than the CNN-only model due to the incorporation of a spatial sequential structure in the learning process.

5.4 Case Study

We downloaded 69,500 streetview images of all major highways in the entire state of Alabama. The major highways include I-10, I-20, I-59, I-65, I-85, and I-459 within Alabama. We then classified the road safety features in those images using our CNN-separateLSTM model. The map for each road safety feature based on the classification results is shown in Figures 10, 11, and 12. We randomly checked a few data points to verify the classification result and found that most of the images were accurately classified.

Fig. 12. Classification map for the concrete barriers class label using the CNN-separateLSTM model.

According to the predicted maps, 54,890 streetview images were classified to have rumble strips, which is around 1,200 km out of 1,390 km in major highways in Alabama. In addition, 12,841 streetview images were classified to have metal crash barriers, which implies around 257 km out of 1390 km. Finally, 26,997 streetview images were classified to have concrete barriers, which implies around 540 km out of 1390 km. We observed that the spatial scale of the predicted map for rumble strips is very long, which is consistent with the spatial scale observed in the ground truth for test datasets. Similarly, from the predicted map in Figure 11, we observed that the metal crash barriers are evenly distributed throughout the major highways in Alabama and have a short spatial scale. This is also consistent with the test datasets. Finally, from the predicted map in Figure 12, we observe that the spatial scale of concrete barriers can vary. We further observed that the long chain concrete barriers are usually located near city areas. Short concrete barriers are usually placed in

the bridges. As well, we noticed that the bridges with the concrete barriers usually do not contain rumble strips.

6 CONCLUSION

In this article, we proposed two different CNN-LSTM-based spatial classification models for mapping safety features along road networks. Our CNN-LSTM models can capture a spatial linear structure between consecutive images along a road network path. Results on real-world Google Street View images collected in Alabama showed that our models outperform several baseline methods. Furthermore, through experimental evaluation, we found out that the separate CNN-LSTM models for independent class labels performed better than the shared CNN-LSTM model.

In future work, we plan to explore extensions with more advanced recurrent neural network structures such as the bi-directional LSTM [7] and attention mechanism [24]. The bi-directional LSTM seems more suitable for our problem since the label of one image is correlated to nearby images in both directions along the road path. We also plan to investigate a more general spatial network structure with graph-LSTM.

REFERENCES

- [1] Vahid Balali, Elizabeth Depwe, and Mani Golparvar-Fard. 2015. Multi-class traffic sign detection and classification using Google Street View images. In *Proceedings of the Transportation Research Board 94th Annual Meeting*.
- [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, R. J. 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 *Proceedings of the 2009 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, Los Alamitos, CA, 248–255.
- [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. *Proceedings of the National Academy of Sciences* 114, 50 (2017), 13108–13113.
- [7] Alex Graves and Jürgen Schmidhuber. 2005. Framewise phoneme classification with bidirectional LSTM and other neural network architectures. *Neural Networks* 18, 5–6 (2005), 602–610.
- [8] Raphael Grzebieta, Mike Bambach, and Andrew McIntosh. 2013. Motorcyclist impacts into roadside barriers: Is the European crash test standard comprehensive enough? *Transportation Research Record* 2377, 1 (2013), 84–91.
- [9] Gayle S. W. 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.
- [10] 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.
- [11] Zhe Jiang. 2018. A survey on spatial prediction methods. *IEEE Transactions on Knowledge and Data Engineering* 31, 9 (2018), 1645–1664.
- [12] 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.
- [13] Zhe Jiang, Yan Li, Shashi Shekhar, Lian Rampi, and Joseph Knight. 2017. Spatial ensemble learning for heterogeneous geographic data with class ambiguity: A summary of results. In *Proceedings of the 25th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*. ACM, New York, NY, 23.
- [14] Zhe Jiang and Arpan Man Sainju. 2019. Hidden Markov contour tree: A spatial structured model for hydrological applications. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, New York, NY, 804–813.
- [15] Zhe Jiang, Arpan Man Sainju, Yan Li, Shashi Shekhar, and Joseph Knight. 2019. Spatial ensemble learning for heterogeneous geographic data with class ambiguity. *ACM Transactions on Intelligent Systems and Technology* 10, 4 (2019), 43.

- [16] Zhe Jiang and Shashi Shekhar. 2017. *Spatial Big Data Science*. Springer.
- [17] Zhe Jiang, Shashi Shekhar, Pradeep Mohan, Joseph Knight, and Jennifer Corcoran. 2012. Learning spatial decision tree for geographical classification: A summary of results. In *Proceedings of the 20th International Conference on Advances in Geographic Information Systems*. ACM, New York, NY, 390–393.
- [18] Zhe Jiang, Shashi Shekhar, Xun Zhou, Joseph Knight, and Jennifer Corcoran. 2013. Focal-test-based spatial decision tree learning: A summary of results. In *Proceedings of the 2013 IEEE 13th International Conference on Data Mining*. IEEE, Los Alamitos, CA, 320–329.
- [19] Zhe Jiang, Shashi Shekhar, Xun Zhou, Joseph Knight, and Jennifer Corcoran. 2014. Focal-test-based spatial decision tree learning. *IEEE Transactions on Knowledge and Data Engineering* 27, 6 (2014), 1547–1559.
- [20] Zhe Jiang, Miao Xie, and Arpan Man Sainju. 2019. Geographical hidden Markov tree. *IEEE Transactions on Knowledge and Data Engineering*. Early Access. July 23, 2019.
- [21] 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.
- [22] Driver Knowledge. 2013. Car Accident Statistics in the U.S. Retrieved July 16, 2020 from <https://www.driverknowledge.com/car-accident-statistics/>.
- [23] 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.
- [24] Yang Liu, Chengjie Sun, Lei Lin, and Xiaolong Wang. 2016. Learning natural language inference using bidirectional LSTM model and inner-attention. arXiv:1605.09090.
- [25] Pradeep Mohan, Shashi Shekhar, James A. Shine, James P. Rogers, Zhe Jiang, and Nicole Wayant. 2011. A neighborhood graph based approach to regional co-location pattern discovery: A summary of results. In *Proceedings of the 19th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*. ACM, New York, NY, 122–132.
- [26] 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 *Proceedings of the International Conference on Web Services*. 134–148.
- [27] Congressional Budget Office. 2018. Public Spending on Transportation and Water Infrastructure, 1956 to 2017. Retrieved July 16, 2020 from <https://www.cbo.gov/system/files/2018-10/54539-Infrastructure.pdf>.
- [28] 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, New York, NY, 98.
- [29] 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.
- [30] 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* 6, 1 (2018), 107–118.
- [31] Arpan Man Sainju and Zhe Jiang. 2017. Grid-based colocation mining algorithms on GPU for big spatial event data: A summary of results. In *Proceedings of the International Symposium on Spatial and Temporal Databases*. 263–280.
- [32] Jennifer D. Schmidt, Ronald K. Faller, Dean L. Sicking, John D. Reid, Karla A. Lechtenberg, Robert W. Bielenberg, Scott K. Rosenbaugh, et al. 2013. *Development of a New Energy-Absorbing Roadside/Median Barrier System with Restorable Elastomer Cartridges*. Technical Report. Nebraska Department of Roads.
- [33] 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.
- [34] 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.
- [35] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alexander A. Alemi. 2017. Inception-v4, Inception-ResNet and the impact of residual connections on learning. In *Proceedings of the 31st AAAI Conference on Artificial Intelligence*.
- [36] 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.
- [37] Victor J. D. Tsai, Jyun-Han Chen, and Hsun-Sheng Huang. 2016. Traffic sign inventory from Google Street View images. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences* 41 (2016), 243–246.
- [38] 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*.
- [39] Yuankai Wu and Huachun Tan. 2016. Short-term traffic flow forecasting with spatial-temporal correlation in a hybrid deep learning framework. arXiv:1612.01022.

- [40] Miao Xie, Zhe Jiang, and Arpan Man Sainju. 2018. Geographical hidden Markov tree for flood extent mapping. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, New York, NY, 2545–2554.
- [41] Huaxiu Yao, Xianfeng Tang, Hua Wei, Guanjie Zheng, Yanwei Yu, and Zhenhui Li. 2018. Modeling spatial-temporal dynamics for traffic prediction. arXiv:1803.01254.
- [42] 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 *Proceedings of the 32nd AAAI Conference on Artificial Intelligence*.
- [43] 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 and Data Mining*. ACM, New York, NY, 984–992.
- [44] 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'17)*, Vol. 14.
- [45] Junbo Zhang, Yu Zheng, Dekang Qi, Ruiyuan Li, Xiuwen Yi, and Tianrui Li. 2018. Predicting citywide crowd flows using deep spatio-temporal residual networks. *Artificial Intelligence* 259 (2018), 147–166.
- [46] 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 *Proceedings of the IEEE International Conference on Computer Vision (ICCV'17)*.
- [47] 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.
- [48] A. Mehrara Molan, M. Rezapour, and K. Ksaibati. 2019. Investigating the relationship between the type of traffic barrier and crash severity in trucks and light vehicles. *J. Traffic. Transp. Eng.* (in press).

Received June 2019; revised August 2019; accepted September 2019