TRAFFIC CONTROL RECOGNITION WITH AN ATTENTION MECHANISM USING SPEED-PROFILE AND SATELLITE IMAGERY DATA

Hao Cheng[1,*], Haoran Lei[2], Stefania Zourlidou[1], Monika Sester[1]

[1] Institut für Kartographie und Geoinformatik, Leibniz Universität Hannover, Germany - {cheng, zourlidou, sester}@ikg.uni-hannover.de
[2] Institut für Kommunikationstechnik, Leibniz Universität Hannover, Germany - haoran.lei@stud.uni-hannover.de

KEY WORDS: Traffic Regulation, Deep Learning, Generative Model, Attention Mechanism, Classification.

ABSTRACT:

Traffic regulators at intersections act as an essential factor that influences traffic flow and, subsequently, the route choices of commuters. A digital map that provides up-to-date traffic control information is beneficial not only for facilitating the commuters' trips, but also for energy-saving and environmental protection. In this paper, instead of using expensive surveying methods, we propose an automatic way based on a Conditional Variational Autoencoder (CVAE) to recognize traffic regulators, i. e., arm rules at intersections, by leveraging the GPS data collected from vehicles and the satellite imagery retrieved from digital maps, i. e., Google Maps. We apply a Long Short-Term Memory to extract the motion dynamics over a GPS sequence traversed through the intersection. Simultaneously, we build a Convolutional Neural Network (CNN) to extract the grid-based local imagery information associated with each step of the GPS positions. Moreover, a self-attention mechanism is adopted to extract the spatial and temporal features over both the GPS and grid sequences. The extracted temporal and spatial features are then combined for detecting the traffic arm rules. To analyze the performance of our method, we tested it on a GPS dataset collected by driving vehicles in Hannover, a medium-sized German city. Compared to a Random Forest model and an Encoder-Decoder model, our proposed model achieved better results with both accuracy and F1-score of 0.90 for the three-class (arm rules of *uncontrolled*, *traffic light*, and *priority sign*) task. We also carried out ablation studies to further investigate the effectiveness of the GPS input branch, the image input branch, and the self-attention mechanism in our model.

1. INTRODUCTION

As the road networks become increasingly complex, commuters need up-to-date road traffic conditions. A digital map that shows real-time traffic conditions and automatically selects the best route is a handy tool to save commuters' commuting time, consequently reducing energy consumption and pollution of the environment. Road traffic conditions and routing can be affected by many factors, and one crucial factor is the intersection regulator. However, the regulator information is subject to change due to, e.g., extreme weather, construction, or traffic control. Considering that using surveying and mapping personnel to inspect and record the latest information of intersection regulators and keep the digital maps updated is highly time-consuming and cost-expensive, the problem of automatic intersection regulator detection needs to be addressed urgently.

Images and vehicle motion data are often used for intersection regulator detection. A popular way to achieve automatic traffic regulator detection is to classify traffic signs based on image features collected by vehicles (Houben et al., 2013). For example, Artificial Neural Networks are trained to extract image features and then the extracted image features are applied for detecting road traffic signs (John et al., 2014; Tian et al., 2019). However, image processing on a large amount of data can consume many resources, such as storage, bandwidth, and energy. Also, there are privacy concerns in processing images, such as the risk of misuse of license plates and personal information. In addition, traffic rules at intersections are highly related to the trajectories and motion of vehicles (Cheng et al., 2020).

Figure 1. Example of an intersection with four arms from Google satellite image.

With the rapid development of Internet of Things (IoT) technology, car floating data (Protschky et al., 2015) including vehicle speed, direction, and location is getting easier and easier to collect. For example, GPS tracks with motion dynamics, instead of images, are leveraged for traffic sign recognition (Zourlidou and Sester, 2019a). Even though vehicle motion data has rich temporal and location information, it lacks the background environment information of the intersection of interest. As exemplified in Fig. 1, the satellite image provides rich spatial features such as the intersection geometry, lane division, and markers. The motion dynamics and intersection contextual information can be combined for traffic regulator detection.

* Corresponding author

In this paper, in order to predict intersection regulators, we propose a deep generative model that takes multimodal inputs, i. e., intersection contextual information from satellite imagery and vehicle motion dynamics. We leverage GPS signals to learn the motion dynamics of the vehicles driving at intersections, namely, the distance and speed information. In addition, we use the GPS positions to extract grid-based local images to learn spatial features aligned with the motion dynamics. These two types of information are combined to train a deep learning framework based on a Conditional Variational Auto-Encoder (CVAE) (Kingma et al., 2014) for intersection regulator prediction. The contributions of this paper are summarized as follows:

- We provide a novel way to combine both GPS sequential data and satellite imagery for detecting arm rules at intersections automatically.
- The GPS and imagery data is leveraged to train a generative model based on a CVAE model. This generative model can detect traffic rules at different arms of intersections.
- We apply a self-attention mechanism (Vaswani et al., 2017) to further learn both the dynamics over the speed sequences and the spatial features over the grid sequences.
- Compared to a Random Forest model and a deep learning Encoder-Decoder deterministic model, the proposed generative CVAE-based model achieve enhanced performance tested on a real-world GPS dataset.

2. RELATED WORK

In this section, we provide a literature review on the works that are most relevant to our work, which is focused on using vehicle motion information for traffic control at intersections.

In general, machine learning approaches, especially Random Forest (Ho, 1995), are commonly used for traffic control recognition based on vehicle motion dynamics. For example, Hu et al. (2015) used the duration of the last stop, minimum traversal speed, number of deceleration, number of stops, and distance from the last stop at an intersection as statistical features (i. e., min, max, mean, and variance) to identify uncontrolled intersection, stop sign, and traffic signal. They tested several supervised and unsupervised methods with different feature settings. They found that the Random Forest classifier with the enabled active and self-learning adapters achieved accuracy above 0.90 trained by only 20% of the data available. Similarly, the work by Méneroux et al. (2018) uses the spatial distribution of vehicle stopping events for traffic signal detection and localization. A Random Forest classifier achieved a detection rate of 85% along with a localization accuracy of about 5 meters for the corresponding traffic regulator. An improved solution to the same two-class classification problem is that of (Méneroux et al., 2020), where traffic signals are detected using speed profiles. A Random Forest classifier using a feature extraction technique achieved 0.95 accuracy. The feature extraction applies a functional analysis of the speed measurement series combined with a wavelet transform. Another Random Forest-based approach is that of Golze et al. (2020), which achieved 0.88 and 0.82 accuracy (averaged performance on cross-validation folders) using non-turning trajectories only and both turning and non-turning trajectories, respectively, in predicting three arm rule classes, i. e., uncontrolled, traffic light, and priority sign. Their Random Forest classifier uses physical features similar to (Hu et al., 2015; Saremi and Abdelzaher, 2015) and the percentage of the trajectories with at least one stop event. The classifier was then fed with the statistical values (minimum, maximum, mean, variance) of the physical attributes, thereby describing each statistical vector's motion behavior along an intersection arm. In this paper, our model is also designed for the same three-class classification task using both turning and non-turning trajectories.

In recent years, deep learning approaches have been widely used for traffic control recognition. The earliest work in this area is that of (Pribe and Rogers, 1999), which describes the training of a Neural Network to learn driving behavior for two types of traffic controls – traffic signal and stop sign. From the GPS trajectories, they extracted several features related to stop events, such as the number of stops, the total duration of stops, and the last three stops before crossing the intersection. In addition, they measured the percentage of the intersections with at least one-stop event for each intersection. Their method achieved an accuracy of 100%, but was only tested on a very small dataset. Recently, Munoz-Organero et al. (2018) analyzed time series of speed and acceleration recordings using a deep learning approach for traffic signal, roundabout, and road crossing identification. Overall, they achieved high accuracy and recognition for the regulator types, i. e., a combined recall of 0.89 and a combined accuracy of 0.88 for the classification task. However, the result for the class of traffic signal shows a significant limitation compared to the other regulator types. Alternatively, Cheng et al. (2020) applied a sequence-to-sequence recognition method by training a CVAE-based classifier on vehicle speed profiles directly extracted from GPS trajectories. The CVAE-based model achieved a prediction accuracy of 0.90. The most recent study to our knowledge is that of (Liao et al., 2021), which proposes a framework for detecting and assessing traffic signals using a Long Short-Term Memory (LSTM) (Hochreiter and Schmidhuber, 1997) based network, called DLSTM. It achieves an AUC value below the ROC curve of 0.95. Along with detecting traffic signals, an estimation of the potential area of influence is provided. This is achieved by exploiting the corresponding GPS trajectories and the contextual characteristics of the intersections, such as intersection type, road type, and traffic flow information.

Furthermore, a more systematic literature review was conducted by Zourlidou and Sester (2019b) on the methods of detection and identification of traffic controllers based on GPS data. The main finding is that GPS data has predictive capabilities, e. g., above 80% in all the reviewed research papers. However, none of these methods reported in the papers directly combined both GPS and imagery data for traffic control recognition. In these papers, in general, the following traffic regulator classes were identified: traffic signals, stop sign, priority sign, yield sign, uncontrolled intersection, roundabout, and turn restriction. Nevertheless, each of the above studies investigated within a subset of those traffic rules. As each method was applied to a single dataset with a specific subset of the regulators, the classification performance, however, cannot be easily generalized; Some studies were considered in the context of a simple two-class classification problem, e. g., traffic signal and non-traffic signal. This makes beachmarking across different models very difficult. In order to guarantee a fair comparison, in this paper we only compare our models with the Random Forest-based model in (Golze et al., 2020) that was tested on the same dataset for the same classification task. We also implement a deep learning Encoder-Decoder approach with the same setting for the evaluation purpose.

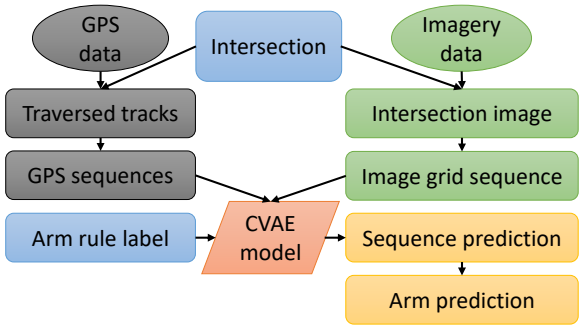

Figure 2. The workflow for arm rule detection using GPS and imagery data.

3. METHODOLOGY

This section explains the methodology of arm rule detection and the proposed model in detail.

3.1 Problem Formulation

Given the central point with the UTM coordinates (x_k, y_k) of intersection k that has m arms, the traffic rules for the arms are defined as $Y_k = \{Y_{k,1}, ..., Y_{k,m}\}$. The GPS tracks traversed the intersection are defined as $\mathbf{X}_k = \{\boldsymbol{X}_i\}_{i=1}^{N}$, where $\boldsymbol{X}_i$ is a GPS track that contains the ordered speed profile $\{d_x^t, d_y^t, v_x^t, v_y^t\}_{t=1}^{T}$ at each time step. d_x^t and d_y^t denote the local distances to the intersection center and v_x^t and v_y^t denote the speeds for both the x and y directions at time step t. The satellite image of the given intersection is defined as I_k. The task of arm rule detection is to build a model $f(Y|\mathbf{X}, I)$ that maximizes the probabilistic prediction of arm rules $\Pi_k^K P(Y_k|\mathbf{X}_k, I_k)$ over all the intersections, where K denotes the total number of intersections.

Figure 2 depicts the workflow for the arm rule detection task. First, we use the central point (x_k, y_k) of the given intersection k to extract all the traversed tracks $\mathbf{X}_k$, and then we compute the vehicle speed profile $\{d_x, d_y, v_x, v_y\}_{t=1}^{T}$ over the recorded total steps T in each track and put it into a timely ordered sequence. Note that we do not use the actual GPS coordinate sequence because, for the detection task, this can cause a domain gap between different intersections. Furthermore, we also use the intersection location to retrieve its satellite image I_k from Google Maps using the developers' API[1]. Considering the large global difference between the images from one intersection to another, we only focus on the local image context associated with the traversed GPS tracks, e.g., intersection geometry, road markers, and lane divisions. To be more specific, a small grid centralized by the current GPS position is cropped at each step from the satellite image over the whole intersection, and the cropped grids are aligned with the speed sequences. Fig. 3 shows an example of the grid sequence aligned with a vehicle's speed sequence.

3.2 Proposed Model

To solve the task defined above, we propose a CVAE-based model. The model that performs probabilistic prediction is denoted as

$$f(Y|\boldsymbol{X}, I) = \arg\max_{\boldsymbol{Y}} p(\boldsymbol{Y}|\boldsymbol{X}, I, \mathbf{z}), \quad (1)$$

[1] https://developers.google.com/maps

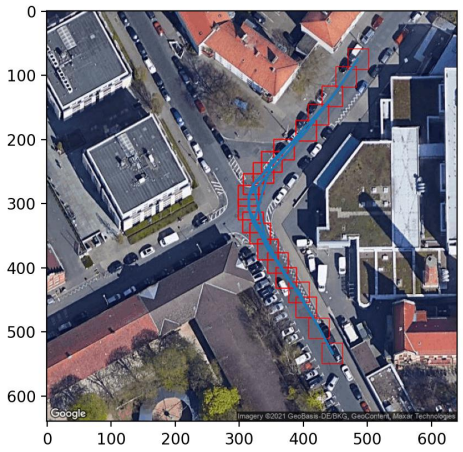

Figure 3. An example of the grid sequence aligned with a vehicle's speed sequence. As denoted by the red bounding boxes, each step is a 32 × 32–pixel grid with the current step's GPS position located in the center.

where $\mathbf{z}$ are the Gaussian latent variables. This generative model is trained to learn a joint distribution into a latent space conditioned on the observations of the intersection images I and vehicle motion dynamics $\boldsymbol{X}$, as well as the inserted ground truth arm rule labels Y. At inference time, a latent variable z_i can be sampled from the learned latent space to combine with the intersection image I_i and a vehicle motion sequence $\boldsymbol{X}_i$ as the input for predicting the intersection rule $\hat{Y}_i$ of each arm.

The variational lower bound (Sohn et al., 2015) of the model is computed as follows:

$$\begin{aligned}\log p_\theta(Y|\boldsymbol{X}, I) \geq & - D_{KL}(q_\phi(\mathbf{z}|\boldsymbol{X}, I, Y)||p_\theta(\mathbf{z})) \\ & + \mathbb{E}_{q_\phi(\mathbf{z}|\boldsymbol{X}, I, Y)}[\log p_\theta(Y|\boldsymbol{X}, I, \mathbf{z})].\end{aligned} \quad (2)$$

The equation above defines two steps in the detection task: training and inference. At training step, $q_\phi(\mathbf{z}|\boldsymbol{X}, I, Y)$ acts as an encoder that uses the training data samples to learn the latent space with the random variables $\mathbf{z}$. Simultaneously, the decoder, denoted as $\log p_\theta(Y|\boldsymbol{X}, I, \mathbf{z})$, decodes the arm rule labels conditioned on the speed and image sequences, as well as the latent variables. At inference step, The latent variables are sampled from the prior $p_\theta(\mathbf{z})$. Hence, the trained decoder can generate the prediction $\hat{Y}_i$ conditioned on speed and image sequences, as well as the sampled latent variables.

In Eq. (2), two losses are used to penalize the prediction errors. The first term on the right-hand side of the equation quantifies the dissimilarity between the approximated posterior $q_\phi(\,)$ and the prior $p_\theta(\mathbf{z})$. Note that here $p_\theta(\mathbf{z})$ is simplified since it can be made statistically independent from the input $\boldsymbol{X}$. This loss is calculated analytically assuming that both distributions are Gaussian (Kingma and Welling, 2014). Also, this term can be seen as a regularizer to make this model more robust against overfitting. The second term of the right-hand side of the equation denotes the expectation of the prediction given the speed information, intersection image, and the learned latent variables. In this paper, we use a categorical cross-entropy to calculate the loss between the ground truth Y_i and the prediction $\hat{Y}_i$. Compared to other deep learning frameworks, this generative model can better learn the stochastic behaviors of various vehicle motions and is relatively easy to train using a limited number of observations for traffic control recognition (Cheng et al., 2020).

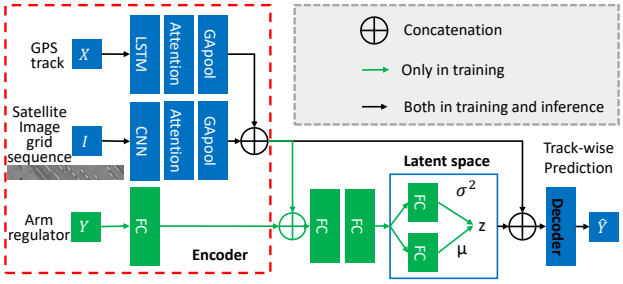

Figure 4. The structure of the CVAE-based model. The red box marks the encoding process. The black arrows indicate the steps done in both the training and the inference processes, while the green arrows indicate the steps done only in the training process.

Figure 4 shows the CVAE-based model implemented with Neural Network layers. The model is divided into *Encoder*, *Latent Space*, and *Decoder*. As shown in the red box in Fig. 4, speed sequences derived from GPS tracks are fed to an LSTM layer for extracting temporal features. To further learn the salient interconnections of temporal features along the time axis of the speed sequences, the output of the LSTM is then fed to a self-attention layer (Vaswani et al., 2017). In the end, a global average pooling layer (GApool) is used to extract the overall GPS features over the vehicle motion sequences. Simultaneously, a light-weight Convolutional Neural Network (CNN) extracts spatial features from the aligned grid sequences one step after another. The feature maps of the CNN are concatenated along the time axis. After the CNN, similarly, a global average pooling layer is used to extract the global spatial features over the grid sequences. The ground truth regulator labels are inserted only during training to train the CVAE model, which is embedded by an FC layer. After the encoder, the extracted features from GPS tracks, satellite image grid sequences, and ground truth labels are combined for learning the latent space via several FC layers. In the end, the decoder implemented by two FC layers predicts arm rules based on the conditional input (GPS and gird sequences) and the latent variables of the latent space. It should be noted that in the training process the latent space is constrained as close as possible to a prior distribution $p_\theta(\boldsymbol{z})$, normally, a Gaussian distribution $p_\theta(\boldsymbol{z}) = \mathcal{N}(\boldsymbol{z}; 0, I)$. At inference time, the decoder generates predictions only based on the conditional input with the latent variables sampled from this prior.

4. DATASET AND PREPROCESSING

In this paper, we leverage the GPS dataset collected by driving vehicles in a medium-sized German city Hannover (Zourlidou and Sester, 2019a). In total, there are 1204 GPS trajectories. These trajectories traversed 1064 intersections that are regulated by 3538 intersection arm rules, which are mainly *traffic-lights*, *priority-signs*, and *uncontrolled rules*.

The GPS data was preprocessed before applying it to the experiments. First, the GPS data was clustered based on the central point of each intersection. To avoid long GPS tracks covering the adjacent intersection(s) of the intersection of interest, they were cut into segments with a maximum distance to the intersection central point in both East (x) and North (y) directions. Second, because the raw GPS data is noisy due to, e. g., signal blockage and drifting, we removed GPS segments that have intervals larger than a predefined threshold between two consecutive time steps to prevent sudden changes in positions. Moreover, similar to Golze et al. (2020), we discarded the intersections that do not have enough number of GPS segments for all the experiments. In the end, the remaining GPS segments were used to compute the speed sequences. We applied a sliding window to further divide the sequences into fixed-sized subsequences to cope with sequence length variation based on GPS signal samples. We did not apply interpolation between GPS signal samples to avoid potential erroneous offset when mapping the GPS position to the image grid. The detailed threshold values for the preprocessing are given in Sec. 5.1 of the experimental settings. Fig. 5 shows the examples of the GPS sequences after preprocessing for intersections regulated by traffic light 5(a), uncontrolled 5(b), and priority sign 5(c). Note that, as opposite to (Golze et al., 2020), we do not differentiate non-turning and turning sequences in our dataset.

We used the developers' API provided by Google Maps to retrieve high-resolution satellite images for each intersection. Empirically, we found that the zoom level 19 provides the best trade-off between a single intersection coverage and image resolution. Considering the differences of scene context information, e. g., buildings, vegetation, and shadows, across intersections, we only focus on the local area aligned with the GPS positions. These local areas are more homogeneous, such as road surface, lane markers, and divisions. Hence, based on the GPS positions in each sequence, we cropped the intersection image into a grid centralized by the current position at each step. Over a complete sequence, we aligned the speed sequence with the corresponding image grid sequence. In order to further reduce the dissimilarity across intersections, the RGB satellite images were converted into grayscale. Examples of the grid sequence aligned with a vehicle's speed sequence can be seen in Fig. 3.

The resulting data is partitioned into training, validation, and test sets for the experiments. Different from (Cheng et al., 2020) that partitioned the dataset randomly over all intersections, we partitioned the dataset according to different arms, ensuring that the image data in training does not leak the intersection contextual information for the test set. Considering the unbalanced data samples in each arm rule class, we sampled the data using a consistent ratio across arm rule classes. The partitioning ratio is set to 56:14:30 for training, validation, and test. Namely, 30% of the unseen arms separated from the total data is used for the test, and 80% and 20% of the remaining data are used for training and validation. Table 1 lists the statistics for each set.

Sets	# Inters.	# Arms	# Seq	# UN	# TF	# PS
Train	205	362	35581	9673	16375	9533
Val.	75	91	7358	2197	3017	2144
Test	141	192	16351	5885	6089	4377
Total	421	645	59290	17755	25481	16054

Table 1. Data for training, validation, and test. UN stands for uncontrolled, TF for traffic light, and PS for priority sign.

5. EXPERIMENT

5.1 Experimental Settings

In the following, we list the basic settings for all the experiments. The thresholds for data preprocessing and hyperparameters of the models are set empirically, and they are kept consistent across all the experiments.

At the data preprocessing step, the satellite images are of a size of 640x640 pixels. The resolution is 0.18 meters, meaning that

(a) Traffic light (b) Uncontrolled (c) Priority sign

Figure 5. Intersections controlled by different arm rules with GPS tracks.

each image covers a squared area of width and height of 116.67 meters. The grid size around each position is 32x32, covering a squared area of width and height of 5.83 meters. According to the average width of the German roads, from 3 to 4 meters[2], this grid size slightly covers a broader area than a lane width so that when the vehicle does not stand in the middle of the lane, the grid is still more likely to cover the whole lane width and includes both lane borders. The maximum distance to the intersection central point is set accordingly to the image coverage, i. e., half of the image width/height in meters. As identical to Golze et al. (2020), the minimum number of GPS sequences in each intersection is 16. We found that the mean value and standard deviation of the time interval between two consecutive GPS signals are 3.1s and 49.8s, respectively. Based on these statistics, sequences with a time-interval larger than 60 seconds were removed.

The hyper-parameters for the Neural Networks are set as below. The sliding window size to divide the speed sequences is set to 8. The CNN has two 2D-convolutional layers with a kernel-size 2 followed by a batch normalization (Ioffe and Szegedy, 2015) and the Rectified Linear Unit activation. The first layer outputs 4 channels and the second layer reduces to 2 channels. The self-attention layer has an embedding dimension of 128 with 8 heads. The dimension of the latent variable is set to 16. All the models were trained using the Adam optimizer (Kingma and Ba, 2014) with the default beta settings and a learning rate of 0.005. Early stopping (Prechelt, 1998) was applied to monitor the training process. The batch size is set to 256, and the number of maximum epochs is 300. In total, our proposed model has circa 363 K trainable parameters. All the models were implemented in Python (Van Rossum and Drake Jr, 1995) using the Keras framework (Chollet et al., 2015) with the TensorFlow backend (Abadi et al., 2016).

In the inference time of the CVAE-based models, we sampled 100 times of the latent random variables from a Gaussian prior. We used the average classification results as the final results.

5.2 Models and Evaluation Metrics for Comparison

The arm rule detection results are compared with that of a Random Forest and Encoder-Decoder model. We also removed the GPS branch, the image branch, and the self-attention layer to analyze their effectiveness for arm rule detection.

[2] http://www.german-autobahn.eu

- RF/GPS features: this is a Random Forest classifier proposed by Golze et al. (2020). It uses the physical and statistical features, such as percentage, distance, and duration of vehicle standstill phases, mean and maximum vehicle speed. We compare the classification accuracy of our proposed model with the Random Forest classifier reported for the complete dataset that contains both turning and non-turning trajectories.
- En-De/GPS+image+att: this is an Encoder-Decoder classifier with the self-attention layers for arm rule detection using the same inputs as the proposed CVAE model. It also uses the same layers to extract spatial and temporal features from the GPS and grid image sequences. The main difference is that this model does not apply the arm label information to learn a Gaussian latent space. Hence, this is a deterministic model compared to the proposed CAVE-based model.
- C/GPS: this is the ablation of the proposed CVAE-based model that only uses the GPS sequences as input. The branch of the image input is removed.
- C/image: this is the ablation of the proposed CVAE-based model that only uses the image grid sequences as input. The branch of the GPS input is removed.
- C/GPS+image: this is the ablation of the proposed CVAE-based model that uses both the GPS and image inputs, while the self-attention layers on both branches are removed.
- C/GPS+image+att: this is the complete proposed model with the self-attention layers and uses both GPS and image data.

The performances of the above models are measured by precision, recall, F1-Score, and accuracy. Considering the unbalanced numbers of samples across different classes, we report the weighted average as a reference for the overall performance.

5.3 Experimental Results

Table 2 lists the classification results by the proposed model C/GPS+image+att on the sequence level. Overall, the model achieved around 0.70 for precision, recall, and F1-Score. As shown in Table 3, when we use a majority vote to summarize the classification results from the sequence level to the arm level, the performance increases but the pattern across classes maintains. Namely, the proposed model achieved 0.90 for precision, recall, and F1-Score for the arm rule detection.

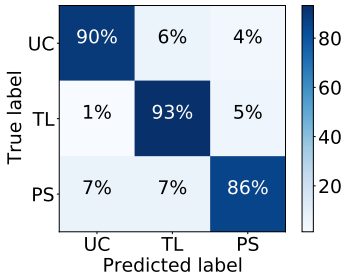

Figure 6. The confusion matrix for arm rule detection.

However, the detection performance differs across the classes. The precision for priority sign is much lower than that for uncontrolled and traffic light at the sequence level. Furthermore, from the confusion matrix shown in Fig.6, 7% of the priority sign has been wrongly classified as traffic light and another 7% of the priority sign has been wrongly classified as uncontrolled. These false detection rates show that the model is not optimal in distinguishing priority sign from the other two regulators.

Regulator	Precision	Recall	F1-Score	Support
Uncontrolled	0.80	0.73	0.77	5885
Traffic light	0.69	0.75	0.72	6089
Priority sign	0.60	0.60	0.60	4377
Weighted avg.	0.71	0.70	0.71	16351

Table 2. Results of the arm rule detection on sequence level.

Regulator	Precision	Recall	F1-Score	Support
Uncontrolled	0.90	0.90	0.90	49
Traffic light	0.90	0.95	0.92	73
Priority sign	0.91	0.86	0.88	70
Weighted avg.	0.90	0.90	0.90	192

Table 3. Results of the arm rule detection.

Table 4 and 5 show the weighted average results of all the models on sequence level and arm level, respectively. Our model achieved superior performance on both levels compared to the Random Forest and Encoder-Decoder models measured by all evaluation metrics. Note that the Random Forest model uses the statistics of the GPS speed profile features. Hence, it is merely measured on the arm level, and only the accuracy value was reported in (Golze et al., 2020). The comparison indicates that our proposed model is more effective in extracting temporal and spatial information from the GPS and satellite imagery data for arm rule detection. Moreover, after removing the self-attention layer, our model performs slightly worse (e.g., accuracy 0.85) but comparable to the Encoder-Decoder model (e.g., accuracy 0.85). Our model only using the GPS data (e.g., accuracy 0.84) performs slightly worse than the Encoder-Decoder model but still better than the Random Forest model (e.g., accuracy 0.82), which is consistent with the result reported by a similar CVAE-based model but using different data partitioning (Cheng et al., 2020). Interestingly, the performance of the CVAE-based model only using the image grid sequences drops significantly (e.g., accuracy 0.68). These observations demonstrate the efficacy of both the attention mechanism and the combination of the image and GPS branches for arm rule detection.

6. DISCUSSION

Despite the enhanced performance as shown above, there are several limitations of our proposed model. First, the model can-

Models	Accuracy	Precision	Recall	F1-score
C/GPS	0.61	0.62	0.61	0.61
C/image	0.61	0.62	0.61	0.60
C/GPS+image	0.68	0.68	0.68	0.67
En-De/GPS+image	0.69	0.69	0.69	0.69
C/GPS+image+att	**0.70**	**0.71**	**0.70**	**0.71**

Table 4. Sequence detection results of different models.

Models	Accuracy	Precision	Recall	F1-score
RF/GPS features	0.82			
C/GPS	0.84	0.85	0.84	0.84
C/image	0.68	0.78	0.68	0.65
C/GPS+image	0.85	0.85	0.85	0.85
En-De/GPS+image	0.85	0.86	0.85	0.85
C/GPS+image+att	**0.90**	**0.90**	**0.90**	**0.90**

Table 5. Arm rule detection results of different models.

not correctly differentiate priority sign from traffic light and uncontrolled. In Germany, similar to traffic light, drivers need to follow priority sign for yielding or proceeding strictly. It has a strong regulation effect similar to traffic light on the traffic at intersections. This may lead to the limited performance of the proposed model in distinguishing these two arm rules. At uncontrolled intersections, drivers are guided by right-of-way. Sometimes, drivers need to negotiate concurrently at the intersection. This increases the difficulty for the model to distinguish priority sign and uncontrolled regulators. Second, we only tested our model using the Hannover dataset. This can restrict the model's generalization in other cities, such as cities in Asia with much higher traffic density and cities in the United States with broader and more lanes. Last but not least, the accuracy of our model needs to be further improved. Compared to many computer vision-based traffic sign detection (Houben et al., 2013; John et al., 2014; Tian et al., 2019), our model still has a large room to be improved. More importantly, arm rules act as a safety-critical factor for traffic at intersections. Thus, the error-tolerant rate, such as false negative and false positive, has to be reduced as much as possible.

7. CONCLUSION AND FUTURE WORK

This paper proposed an automatic approach for arm rule detection at intersections. The model takes multimodal inputs, i. e., intersection contextual information from satellite imagery data and vehicle motion dynamics, for the detection task. The motion dynamics of the vehicles' driving speed are learned from GPS signals. Also, the GPS positions are used to extract grid-based local images for learning spatial features aligned with the motion dynamics. These two types of information are combined to train a deep learning framework based on a Conditional Variational Auto-Encoder (CVAE) for intersection regulator prediction. The ablation studies showed that combining the GPS and satellite imagery data is more beneficial than only depending on the GPS data for arm rule detection. This multimodal spatial and temporal information is effectively extracted by the LSTM, CNN , and self-attention layers. Moreover, the CVAE-based generative model is more robust than a Random Forest model and a deterministic Encoder-Decoder model.

In future work, rather than assuming that an intersection is independent from other intersections in its vicinity, we will explore more sophisticated models, such as graph neural networks (Wu et al., 2020), to learn the spatial connectivity between intersec-

tions, e. g., treating intersections as nodes and the lane segments between intersections as directed edges. In addition, we will seek different GPS datasets collected in other cities and countries to analyze the model's generalizability.

ACKNOWLEDGMENTS

This work is supported by the German Research Foundation (DFG) via the project GRK2159 i.c.sens.

References

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., Dean, J., Devin, M. et al., 2016. Tensorflow: Large-scale machine learning on heterogeneous distributed systems.

Cheng, H., Zourlidou, S., Sester, M., 2020. Traffic Control Recognition with Speed-Profiles: A Deep Learning Approach. *ISPRS International Journal of Geo-Information*, 9(11), 652.

Chollet, F. et al., 2015. Keras.

Golze, J., Zourlidou, S., Sester, M., 2020. Traffic Regulator Detection Using GPS Trajectories. *KN-Journal of Cartography and Geographic Information*, 70(3), 95–105.

Ho, T. K., 1995. Random decision forests. *Proceedings of 3rd international conference on document analysis and recognition*, 1, IEEE, 278–282.

Hochreiter, S., Schmidhuber, J., 1997. Long short-term memory. *Neural computation*, 9(8), 1735–1780.

Houben, S., Stallkamp, J., Salmen, J., Schlipsing, M., Igel, C., 2013. Detection of traffic signs in real-world images: The German Traffic Sign Detection Benchmark. *International Joint Conference on Neural Networks*.

Hu, S., Su, L., Liu, H., Wang, H., Abdelzaher, T. F., 2015. SmartRoad: Smartphone-Based Crowd Sensing for Traffic Regulator Detection and Identification. *ACM Trans. Sen. Netw.*, 11(4), 55:1–55:27.

Ioffe, S., Szegedy, C., 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *International conference on machine learning*, PMLR, 448–456.

John, V., Yoneda, K., Qi, B., Liu, Z., Mita, S., 2014. Traffic light recognition in varying illumination using deep learning and saliency map. *17th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, IEEE, 2286–2291.

Kingma, D. P., Ba, J., 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Kingma, D. P., Mohamed, S., Jimenez Rezende, D., Welling, M., 2014. Semi-supervised learning with deep generative models. *Advances in neural information processing systems*, 27.

Kingma, D. P., Welling, M., 2014. Auto-encoding variational bayes. *ICLR*.

Liao, Z., Xiao, H., Liu, S., Liu, Y., Yi, A., 2021. Impact Assessing of Traffic Lights via GPS Vehicle Trajectories. *ISPRS International Journal of Geo-Information*, 10(11).

Méneroux, Y., Guilcher, A., Saint Pierre, G., Hamed, M., Mustiere, S., Orfila, O., 2020. Traffic signal detection from in-vehicle GPS speed profiles using functional data analysis and machine learning. *International Journal of Data Science and Analytics*, 10, 101–119.

Méneroux, Y., Kanasugi, H., Saint Pierre, G., Guilcher, A. L., Mustière, S., Shibasaki, R., Kato, Y., 2018. Detection and Localization of Traffic Signals with Gps Floating Car Data and Random Forest. 114, Schloss DagstuhltextendashLeibniz-Zentrum fuer Informatik, Melbourne, Australia, 11:1–11:15.

Munoz-Organero, M., Ruiz-Blaquez, R., Sánchez-Fernández, L., 2018. Automatic detection of traffic lights, street crossings and urban roundabouts combining outlier detection and deep learning classification techniques based on GPS traces while driving. *Computers, Environment and Urban Systems*, 68, 1-8.

Prechelt, L., 1998. Early stopping-but when? *Neural Networks: Tricks of the trade*, Springer, 55–69.

Pribe, C. A., Rogers, S. O., 1999. Learning To Associate Observed Driver Behavior with Traffic Controls. *Transportation Research Record: Journal of the Transportation Research Board*, 1679(1), 95-100.

Protschky, V., Ruhhammer, C., Feit, S., 2015. Learning traffic light parameters with floating car data. *2015 IEEE 18th International Conference on Intelligent Transportation Systems*, IEEE, 2438–2443.

Saremi, F., Abdelzaher, T. F., 2015. Combining Map-Based Inference and Crowd-Sensing for Detecting Traffic Regulators. *2015 IEEE 12th International Conference on Mobile Ad Hoc and Sensor Systems*, 145-153.

Sohn, K., Lee, H., Yan, X., 2015. Learning structured output representation using deep conditional generative models. *In Proceedings of Advances in Neural Information Processing Systems (NIPS)*, 3483–3491.

Tian, Y., Gelernter, J., Wang, X., Li, J., Yu, Y., 2019. Traffic sign detection using a multi-scale recurrent attention network. *IEEE transactions on intelligent transportation systems*, 20(12), 4466–4475.

Van Rossum, G., Drake Jr, F. L., 1995. *Python reference manual*. Centrum voor Wiskunde en Informatica Amsterdam.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., Polosukhin, I., 2017. Attention is all you need. *NeuIPS*, 5998–6008.

Wu, Z., Pan, S., Chen, F., Long, G., Zhang, C., Philip, S. Y., 2020. A comprehensive survey on graph neural networks. *IEEE transactions on neural networks and learning systems*, 32(1), 4–24.

Zourlidou, S., Sester, M., 2019a. Traffic regulator detection and identification from crowdsourced data—a systematic literature review. *ISPRS International Journal of Geo-Information*, 8(11), 491.

Zourlidou, S., Sester, M., 2019b. Traffic Regulator Detection and Identification from Crowdsourced Data—A Systematic Literature Review. *ISPRS International Journal of Geo-Information*, 8(11).