

Synthetic Traffic Signs Dataset for Traffic Sign Detection & Recognition In Distributed Smart Systems

Ilias Siniosoglou, Panagiotis Sarigiannidis
Department of Electrical and Computer Engineering
University of Western Macedonia
Kozani, Greece
{isiniosoglou, psarigiannidis}@uowm.gr

Yannis Spyridis, Anish Khadka, Georgios Efstathopoulos
0 Infinity Limited
London, United Kingdom
{yannis, anish, george}@0infinity.net

Thomas Lagkas
Department of Computer Science
International Hellenic University
Kavala Campus, Greece
tlagkas@cs.ihu.gr

Abstract—Traffic sign recognition (TSR) is a key aspect involved in the development of robust automated transportation systems. It inherently involves the task of traffic sign detection (TSD), which can be challenging due to traffic signs often being subject to deterioration or occlusion, caused by various environmental factors, or through actions of vandalism. Even though, notable advancements have been achieved in the areas of TSR and TSD, few studies have provided robust algorithms, able to be generalized in real-world applications. This mostly stems from the lack of an extensive traffic sign dataset, standardized for benchmarking purposes. In light of the aforementioned, this paper presents a novel traffic sign dataset, which consists of the Carla Traffic Sign Detection (CTSD), and the Carla Traffic Sign Recognition Dataset (CATERED), targeting the detection and recognition processes respectively. Using the proposed dataset for training and evaluation, a deep Auto-Encoder algorithm is presented, demonstrating high accuracy in detecting and recognizing the distorted traffic signs. Finally, the system is further extended to a federated learning environment, exemplifying its applicability in modern decentralized and interconnected architectures.

Index Terms—Traffic Sign Recognition, Traffic Sign Detection, Autoencoder, Federated Learning, Image synthesis, Image classification, Anomaly Detection

I. INTRODUCTION

In the modern age and with the technological advancements in the fields of smart interconnection and smart systems, such as, smart cities, smart-grids and other cyber-physical systems, the fields encapsulating optical detection and further classification of the key information they envelop are constituted as critical. Systems like self driving cars, localization services and guidance platforms have been blooming in the light of the advancement of artificial intelligence (AI) and the accelerated hardware that support it, for tasks like scene recognition and environmental analysis that help create a digital

understanding of the respective surrounding environment. In particular, infrastructures like traffic sign recognition (TSR) systems that undertake the real time detection and analysis of traffic signs in public transport networks have started to play a major role in automating transportation and localization infrastructures, the lack of which can have catastrophic consequences, like in the scenario of self-driving cars.

Traffic sign recognition systems are essential in many real-world applications such as autonomous driving, traffic surveillance, driver safety and assistance, road network maintenance, and analysis of traffic scenes. Normally, a TSR system concerns two related subjects which are traffic sign detection (TSD) and traffic sign recognition. The former focuses on the localization of the targets in the pictures while the latter performs a fine-grained classification to identify the type of targets detected.

These applications though, suffer from some basic drawbacks. Firstly, the lack of established benchmark traffic sign detection and recognition datasets [1] that redound in the need for the generation of such custom data, which is an arduous and rather time consuming process, that will not necessarily procure the correct information to train TSD and TSR algorithms, respectively. Another drawback is that, commonly, landmarks like traffic signs tend to never go unsullied, either by the destructive forces of nature or the destructive forces of vandalism. This constitutes the need for datasets that incorporate not only clear images of the correlated traffic signs, but also distorted, smudged or otherwise obstructed images of those signs in order to train robust and generalized detection and recognition algorithms. Moreover, modern systems are slowly but steadily migrating to decentralized architectures, keeping the security and privacy of the handled data as a key factor of their subsequent implementations. Such systems

utilize the power of decentralized technologies, like Federated Learning (FL), which is a field that focuses on remotely training Machine Learning (ML) and Deep Learning (DL) models on the edge, in a completely data-agnostic way. Due to its decentralized span, the data of an FL environment can be either statistically distributed or not. This, though, constitutes the need for a versatile and flexible, but also robust, dataset and accompanying algorithm to accommodate the decentralized paradigm, that is steadily becoming the new Deep Learning status quo. Finally, though a lot of work has been done in the field to oblige to the aforementioned drawbacks, a benchmark robust and easily adaptable algorithm has yet to be defined and used in a general manner.

In response to the demand for a dataset that fulfils the described preconditions, this work presents a traffic sign recognition dataset to be used for training Machine Learning and Deep Learning algorithms in successfully detecting and interpreting roadside guidance traffic signs, both in a central and decentralized environment. The contributions of this work can be summarized as follows:

- *Produces a novel traffic sign recognition dataset to be used in training TSR ML and DL algorithms that includes multiple classes and image deterioration as anomalies to ensure model generalization*
- *Develops a DL model specifically for TSR that is evaluated on vast testing data*
- *Presents a robust system and a methodology to train and evaluate the aforementioned model*
- *Extends the methodology to a Federated Learning environment proving its utility to modern decentralized interconnected systems*

The rest of the paper is organized as follows. Section II introduces some basic information about the techniques used in this work as well as a review of similar work done in the TSD and TSR fields. Section III presents the methodology, dataset and the general workings of the presented contributions, while section IV validates the results of this paper. Finally, section V concludes this work.

II. BACKGROUND

A. Traffic Sign Detection and Recognition Data

Over the last decade, several traffic sign databases have been released, targeting the tasks of traffic sign detection and classification. One of the most distinctive publicly available datasets, is The German Traffic Sign Recognition Benchmark (GTSRB) [2], which consists of multi-class images, captured in various locations in Germany. Likewise, the DFG Traffic Sign Data Set [1], includes 7000 annotated images from Slovenia province roads, intended primarily for Traffic Sign Recognition, and incorporating additional augmented datasets. In addition, the Mapping and Assessing the State of Traffic Infrastructure (MASTIF) [3], [4] dataset includes up to 88 augmented sign categories, acquired from road maintenance in Croatia, intended for traffic sign recognition as well.

Comprehensive datasets for both detection and recognition of traffic signs are included inside the BelgiumTS dataset [5].

The detection dataset provides a total of 25634 images, with 5905 and 3101 annotated images used for training and testing respectively. The included classification dataset consists of 4591 images used for training and 2534 images used for testing purposes. Containing more than 20000 images, including both highway and city road traffic signs, the Swedish Traffic Signs (STS) [6] dataset has a total of 3488 annotated traffic signs, and can be used for both tracking and detection techniques.

One of the most extensive benchmarking datasets, containing thousands of annotated signs, is the Tsinghua-Tencent 100K [7]. The included images are of very high resolution and target realistic scenarios by covering different environmental conditions, such as low luminance, or occlusion circumstances. Additional datasets of interest include the LISA [8] dataset, containing 6610 annotated traffic signs and the Stereopolis Database [9] which involves signs from complex urban sites. Finally, the European Traffic Sign Dataset (ETSD) [10], contains annotated signs from six different countries, consisting of more than 80000 images that are categorized into 164 classes. Table I outlines some of the most important publicly available datasets.

B. Traffic Sign Detection and Recognition Methods

The areas of traffic sign detection and recognition have received significant attention by the research community in recent years, in light of autonomous driving systems that have risen, due to the proliferation of AI techniques that enable their robust realization. In lack of extensive traffic sign datasets established as a standard for benchmarking related approaches, review studies mainly focus on identifying challenges associated with TSD and TSR, offering relative analyses that aim to draw conclusions for prospective research [11], [12].

Common approaches for recognition of traffic signs include the use of support vector machines (SVMs) [13], which are linear classifiers based on supervised ML models. SVMs can make use of histogram of oriented gradient (HOG) features for training [14], demonstrating enhanced resilience in environmental lighting fluctuations, which often affect the process of TSR. Due to the generalization capabilities inherent in Deep Learning architectures, other methods examine the use of convolutional neural networks (CNNs) [15], [16] that offer

Dataset	No. of classes	No. of signs
GTSRB	43	>50000
DFG	200	13239
MASTIF	88	5184
BTSD	62	4627
BTSC	62	7125
STS	7	3488
TT 100K	45	30000
LISA	49	6610
Stereopolis	10	251
ETSD	164	82476

TABLE I: Publicly available traffic sign recognition and detection datasets.

enhanced feature extractions, and are able to produce a high recognition accuracy. These networks however, often suffer from high computational costs, which led to new approaches that examine a combined solution, where a CNN is used for feature extraction and a classifier such as an Extreme Learning Machine (ELM) [17] is used for the classification.

C. Auto-Encoders

Concerning the implementation of the proposed algorithms, this work relies heavily on the concept of Auto-Encoder and their powerful utility in augmentative tasks. The core design behind the Deep Auto-Encoder [18] architecture is the assimilation of given data of space X into a compressed manifold F by a sub-portion of the model working as an encoder and consequently the scaling of that manifold F to the predicted value P of those given data by a second sub-portion acting as a decoder, where $P \sim X$. Equation 1 defines the concept of the Auto-Encoder's mapping of data space X to manifold F and its decompression to space P .

$$\begin{aligned} r, p : \underset{r, p}{\operatorname{argmin}} \|X - (p \circ r)X\|^2 \\ r : X \rightarrow F, p : F \rightarrow P \end{aligned} \quad (1)$$

D. Federated Learning

Federated Learning is defined as a stochastic distributed learning and privacy-preserving process that orchestrates the distribution, organization, training and fusion of Deep Learning models across a distributed corpus of edge devices or remote workers over distributed networks [19] [20]. FL works by training locally DL models on edge devices with the local on-device collected data, diminishing the need to communicate that data to a centralized system, thus keeping the data private and the nodes secure. The weights of the distributed locally trained models are then retrieved by a central system and are aggregated into one global model containing the merged knowledge of the corpus of edge devices, using specified fusion algorithms such as the commonly utilised Federated Averaging [21]. Consequently, the models are send back to the remote nodes and replace the local models with the fused model. This way, the new models can perform better with knowledge that does not necessarily exist on the respective device, but on another, without the need for data sharing.

Specifically, the central server distributes a global model w_{Global}^0 along with training instructions to a Federated population $P_f \in [1, N]$ where $N \in \mathbb{N}^*$, each holding local dataset $D_{i \in N}$ and local models w_i^l . The distributed models are subsequently trained on the local data D_i and then the weights w_{Global}^i are send back to the central system to be aggregated through a process like Federated Averaging (2), or similar, in order to produce an updated global model w_{Global}^k [22].

$$w_G^k = \frac{1}{\sum_{i \in N} D_i} \sum_{i=1}^N D_i w_i^k \quad (2)$$

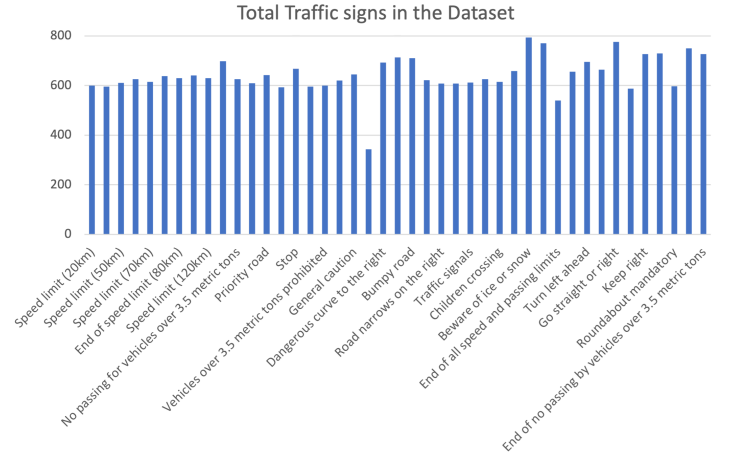


Fig. 1: Histogram of the total samples in the proposed dataset.

Where w_G^k is the global model at the k_{th} training iteration and w_i^k denotes the Federated population i_{th} model at that iteration.

III. METHODOLOGY

This section overviews the summary of the developed methodology, as well as the techniques used in this work. The methodology is divided into four parts, a) the produced dataset, b) the anomaly detection and mitigation strategies, c) the training of the developed centralized algorithms and d) the federated model training.

A. Produced Datasets

As mentioned previously, the core aspect of this work focuses on producing a new TSD and a TSR dataset in order to help train and validate ML and DL systems, for traffic roadside landmark recognition. For the detection dataset, the Carla Simulator [2] was used to generate a large dataset for traffic sign detection purposes. The histogram of the total samples in the proposed dataset is shown in Figure 1. The Carla Traffic Sign Detection (CTSD) dataset contains:

- 55,323 images of 1920 x 1080 image pixel resolution
- Captured in various day time settings
- 43 different classes of traffic signs
- Highly accurate annotation performed automatically
- Various weather and lighting conditions (e.g., cloudy, sunny, day, night, etc)

The produced traffic sign recognition dataset, i.e., the Carla Traffic Sign Recognition Dataset (CATERED), incorporates cropped traffic signs, generated also through the Carla simulator. The dataset consists of:

- 94478 images with varying resolution of traffic signs
- 43 different traffic signs
- Highly accurate annotation, performed automatically

Figure 2 provides a sample preview of the 43 traffic sign classes. For the purposes of detecting distorted traffic sign images, alterations have been augmented onto the images which henceforth will be referred to as attacks. The augmentations are documented bellow.



Fig. 2: Sample preview of the 43 traffic sign classes.

B. Anomaly Detection and Mitigation Strategies

The attack detection model for synthetic traffic signs, is an optimized version of the model developed for the real image traffic attack detection model. The optimised model has been updated according to the needs of the new dataset and the requirements of anti-hacking devices. The new architecture is simpler and layers such as Convolution Transpose, Leaky-ReLU etc. are not considered in this version. Instead of convolution Transpose, UP sampling and convolution strides are utilized, while Leaky-ReLU was replaced by the classic ReLU. These modifications achieve a good balance between performance and complexity, especially for the embedded anti-hacking devices.

The model consists of a total of 8,487,891 parameters. Figure 3 and Table II show the architecture of the attack detection model for synthetic traffic signs, and the corresponding parameters respectively. The attack detection is carried out in the same manner as the analyses for the real traffic signs.

C. Centralized Model Training

The centralized model is an Auto-Encoder DL model, developed in order to accumulate traffic sign images and reproduce them in the best possible manner. For the model construction, the Adam optimizer [23] was used as a gradient optimizer with a standard decay over the training. Additionally, during the training phase, various data augmentation techniques were also applied to the data. The augmentation process aims to increase the robustness of the attack detection and recognition models, by producing new varieties on the input data. A list of all the augmentation methods applied to the training data is shown in Table IV, while the parameters of the produced DL model are presented in Table III. The Auto-Encoder model is trained over a specified number of iterations and is optimized with the Mean Squared Error loss function, shown in equation 3:

$$MSE = \frac{1}{n} \sum_{i=1}^n (y_i - \tilde{y}_i)^2, \quad (3)$$

where n represents the number of predictions, while Y and \tilde{Y} are the samples and predicted values vector, respectively.

Layer Type	Output	Strides	Activation	Filter
Input	48x48	1	ReLU	3
Conv2D- 1	48x48	1	ReLU	64
Conv2D- 2	24x24	2	ReLU	128
Conv2D- 3	24x24	1	ReLU	256
Conv2D- 4	12x12	2	ReLU	128
Conv2D- 5	12x12	1	ReLU	512
Conv2D- 6	6x6	2	ReLU	128
Dropout- 1	-	-	-	-
Flatten	4608	-	-	-
Dense - 1	48	-	ReLU	-
Dense - 2	55584	-	ReLU	-
Reshape	12x12	-	ReLU	386
Conv2D- 7	12x12	1	ReLU	512
Up sampling -1	24x24	-	-	-
Conv2D- 8	24x24	1	ReLU	256
Dropout- 2	-	-	-	-
Conv2D- 9	24x24	1	ReLU	256
Up sampling -2	48x48	-	-	-
Conv2D- 10	48x48	1	ReLU	64
Dropout- 3	-	-	-	-
Conv2D- 11	48x48	1	Tanh	3

TABLE II: The proposed architecture of anomaly detection used for real traffic signs.

Layer Type	Output Shape	Parameters
Input Layer	48 x 48 x 3	0
Model	48 x 48 x 3	8,487,891

TABLE III: Spatial characteristics of the produced DL model

Moreover, since an Auto-Encoder presents an Augmentative architecture, meaning that it outputs an interpretable value with the characteristics of the input, in this work an image, a way to adapt it to the problem at hand has to be defined. In order for the Auto-Encoder to output a value pointing at the probability of the input sample to belong to an attack or not a distance function was defined. In particular, the distance function takes as input both the input given to the Auto-Encoder and the output sample and finds the distance between them resulting to an error ε . ε over a value θ are found to belong to an attack sample whilst the rest are benign. The value θ is found through an experimental process. Equation 4 denotes the distance function utilized to find the aforementioned error,

$$\varepsilon = \|s_r - s_p\| \quad (4)$$

where ε denotes the calculated error, s_r the real input sample and s_p the Auto-Encoder's output, resulting from the knowledge that the model has accumulated.

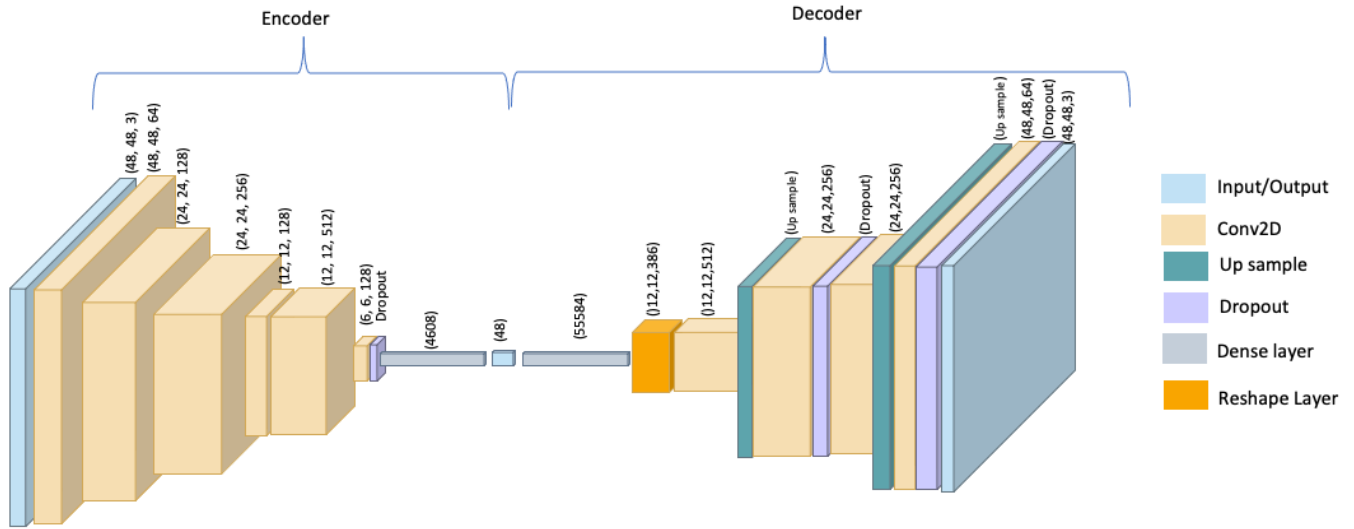


Fig. 3: The architecture of the attack detection model on traffic signs for the synthetic data.

Type	Parameters	Description
Rotation rang	0 - 20 degree	The training image i.e., GPS map is randomly rotated between 0 - 20 degrees.
Zoom range	0 - 40	Zooming is applied on the training images by a random scaling factor in the range of 0 - 40 percent.
Width and height shift range	0 - 10 percent	The images are shifted left, right, up or down in random combinations and magnitudes in the range of 0 - 10 percent.
Shear	0 - 10 percent	The images are stretched randomly in the range of 0 - 10 percent.
Fill	Nearest mode	Missing pixel values are filled with nearby pixel values.
Horizontal and Vertical Flip	True	The images are flipped in a horizontal and vertical direction randomly. Only used for the GPS map

TABLE IV: All the augmentation methods that were applied on the training data for the anomaly detection model.

D. Federated Model Training

The federated learning training methodology extends the centralized training method and applies it in a distributed manner. Specifically, multiple Auto-Encoder models are disbursed at a number of remote edge devices in a distributed network. A central FL server serves to orchestrate the federated process for this implementation, as a could service. Figure 5 depicts the architecture followed to realize the FL in this work. When beginning the federated process, the remote workers register to the FL server to signify their willingness for federation. Next, the FL server sends the training information to the nodes and they start to locally train their Auto-Encoder models on their data. On the end of every FL round the weights of the local Auto-Encoders are sent to the FL server where they are fused, and subsequently sent back and synced to the old preexisting model. This process is reiterated until the model is optimized.

IV. EVALUATION

This section provides the justification of the proposed dataset and algorithms and establishes the experimental results

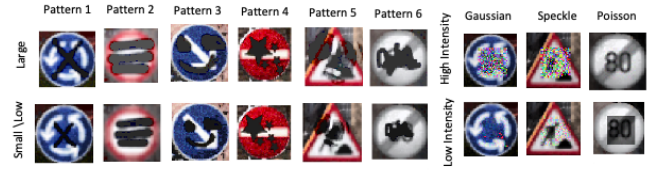


Fig. 4: Attacked traffic sign images, for different area size and noise intensity

that prove this work's contribution. Its main purpose is to provide the experimental impact of the produced dataset of this work on two benchmark applications, i.e., a standalone model and a distributed Deep Learning platform. The experiments are oriented in differentiating the attack augmented traffic sign images from the real ones. The two deployments with their respective experiments are described below.

A. Centralized Model Evaluation

The model was trained and evaluated on an NVIDIA GTX 2080Ti GPU, using the Keras-Tensorflow framework. For traffic sign attack detection, extensive experiments were carried out to validate the performance of the developed pipeline. The traffic synthetic signs, were altered with a pattern attack, consisting of six different types of patterns and three different types of noise attacks. The attacks have further variants, such as that pattern attacks can cover a large and small area, whereas noise attacks can be of low or high intensity. The synthetic traffic sign dataset contained 12,900 images, with 300 images selected for each of the 43 classes. Figure 4 displays the attacked traffic sign images, for different area size and noise intensity.

Table V presents the results of the Auto-Encoder against the different image alterations or attacks. As can be seen, the accuracy of detection in most of the attacks cross the 0.9 threshold, with the exception of the Poisson alteration attack as well as the Poisson Noise - High attack.

B. Federated Deployment Evaluation

Since the federated setting needs multiple nodes, a virtual testbed was devised. The experiments ran on an Ubuntu 20.04 workstation consisting of an Intel Core i7, 64Gb of memory and an NVIDIA RTX 3080 GPU/10Gb GPU memory. The setting, which also can be seen in Figure 5, incorporated four virtualized remote worker nodes enclosed in Docker containers. The Docker containers were then connected to an FL server running on the same machine.

The data used for the FL experiments were segmented samples of the proposed dataset of this work. To be able to evaluate correctly the performance of the Federated Auto-Encoders on the proposed dataset, the intact data could not be used. The data were segmented into four different configurations to fairly produce trusted results. The segmentation categories are a) Independent and Identically Distributed (IID), b) Independent and Identically Distributed (IID - No Repeat), c) non-Independent and Identically Distributed (Non-IID) and d) non-Independent and Identically Distributed (Non-IID- No Repeat). In particular, the IID category describes data that are balanced, fairly distribute and in whole statistically homogeneous, in extent IID - No Repeat are IID data without reoccurring samples, i.e. each sample exists only once in the whole FL environment dataset. On the other hand, Non-IID and Non-IID- No Repeat reveal non identically distributed data with heterogeneous statistical placements each, respectively.

The FL experiments ran for 3 federated rounds, each consisting of 3 local training rounds and 100 steps each round, using a batch size of 128 samples. After each round, the distributed models were fused by the FL server and synced to the remote devices.

As can be extrapolated by the results seen in Table VI, the FL environment combined with the proposed data help the Auto-Encoder model converge rather fast and perform well on the test data.

V. CONCLUSION

With the advancement of smart infrastructures and automatic recognition/response systems, the need for detailed data,

Attack Type	Precision	Recall	F1-score	Model Accuracy
Pattern 1 - Larger	0.989	0.9961	0.9925	0.9919
Pattern 1 - Small	0.9424	0.9021	0.9218	0.945
Pattern 2 - Larger	0.9885	0.9196	0.9528	0.9537
Pattern 2 - Small	0.9655	0.8776	0.9195	0.9327
Pattern 3 - Larger	0.9891	0.9998	0.9944	0.9938
Pattern 3 - Small	0.989	0.9934	0.9912	0.9906
Pattern 4 - Larger	0.9658	0.932	0.9486	0.9599
Pattern 4 - Small	0.989	0.9979	0.9934	0.9928
Pattern 5 - Larger	0.9891	0.9998	0.9944	0.9938
Pattern 5 - Small	0.9658	0.932	0.9486	0.9599
Pattern 6 - Larger	0.989	0.9775	0.9832	0.9826
Pattern 6 - Small	0.9658	0.8893	0.9259	0.9385
Gaussian Noise - Low	0.9796	0.3928	0.5608	0.6903
Gaussian Noise - High	0.9891	1	0.9945	0.9939
Poisson - Low	0.9058	0.31	0.4626	0.6492
Poisson Noise - High	0.3464	0.0207	0.0392	0.5043
Speckle Noise - Low	0.9888	0.9645	0.9765	0.9762
Speckle Noise - High	0.9874	0.8961	0.9395	0.9419

TABLE V: Centralized anomaly detection results.

Experiment	Precision	Recall	F1-Score	Accuracy
IID	0.9459	0.9136	0.9289	0.9298
IID - No Repeat	0.9655	0.9228	0.9397	0.9403
Non-IID	0.991	0.899	0.9419	0.9404
Non-IID - No Repeat	0.9572	0.934	0.9428	0.9419

TABLE VI: FL Anomaly Detection Results

but also algorithms to take advantage of that data, has been an increasing need in the latest years. Specifically, in the field of auto-navigation, location and surveillance systems, and with the introduction of Deep Learning and its distributed extension Federated Learning, the need for benchmark traffic-oriented information is ever-increasing. Undertaking the task of producing data to tackle this demand, in this paper we present a new Traffic Sign Data Collections and Recognition Dataset, i.e., the CTSD and CATERED datasets for TSD and TSR respectively. These data are augmented with targeted deformations in order to simulate phenomena like naturally distorted signs, smudged images or vandalism. To detect these deformations a DL model following the Auto-Encoder architecture is proposed and tested in two separate scenarios, i) a centralized detection system and ii) a decentralized (Federated) system, aiming to simulate the dataset's impact on modern smart infrastructures. The produced results show that the proposed data and techniques achieve a high percentage of success in the detection of the aforementioned deformations in both said settings, presenting over 90% in almost all cases and in different data configurations, proving their utility in general these systems.

ACKNOWLEDGMENT

This project has received funding from the European Union's Horizon 2020 programme under grant agreement No. 833611 (CARMEL).

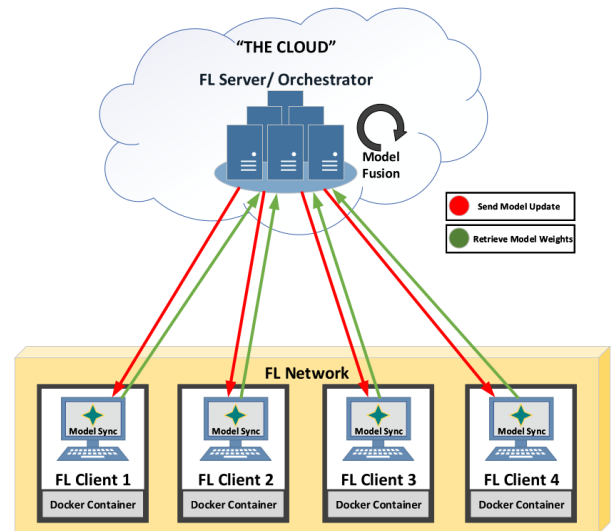


Fig. 5: Federated Testbed Architecture.

REFERENCES

- [1] D. Tabernik and D. Skočaj, "Deep Learning for Large-Scale Traffic-Sign Detection and Recognition," *IEEE Transactions on Intelligent Transportation Systems*, 2019.
- [2] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, "CARLA: An open urban driving simulator." [Online]. Available: <http://arxiv.org/abs/1711.03938>
- [3] S. Šegvić, K. Brkić, Z. Kalafatić, V. Stanisavljević, M. Ševrović, D. Budimir, and I. Dadić, "A computer vision assisted geoinformation inventory for traffic infrastructure," in *13th International IEEE Conference on Intelligent Transportation Systems*, 2010, pp. 66–73.
- [4] Z. Huang, Y. Yu, J. Gu, and H. Liu, "An efficient method for traffic sign recognition based on extreme learning machine," *IEEE Transactions on Cybernetics*, vol. 47, no. 4, pp. 920–933, 2017.
- [5] M. Mathias, R. Timofte, R. Benenson, and L. Van Gool, "Traffic sign recognition — how far are we from the solution?" in *The 2013 International Joint Conference on Neural Networks (IJCNN)*, 2013, pp. 1–8.
- [6] F. Larsson and M. Felsberg, "Using fourier descriptors and spatial models for traffic sign recognition," in *Image Analysis*, A. Heyden and F. Kahl, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 238–249.
- [7] Z. Zhu, D. Liang, S. Zhang, X. Huang, B. Li, and S. Hu, "Traffic-sign detection and classification in the wild," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 2110–2118.
- [8] A. Møgelmoose, D. Liu, and M. M. Trivedi, "Detection of u.s. traffic signs," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 6, pp. 3116–3125, 2015.
- [9] R. Belaroussi, P. Foucher, J.-P. Tarel, B. Soheilian, P. Charbonnier, and N. Paparoditis, "Road sign detection in images: A case study," in *2010 20th International Conference on Pattern Recognition*, 2010, pp. 484–488.
- [10] C. Gámez Serna and Y. Ruichek, "Classification of traffic signs: The european dataset," *IEEE Access*, vol. 6, pp. 78 136–78 148, 2018.
- [11] C. Liu, S. Li, F. Chang, and Y. Wang, "Machine vision based traffic sign detection methods: Review, analyses and perspectives," *IEEE Access*, vol. 7, pp. 86 578–86 596, 2019.
- [12] S. Wali, M. Hannan1, A. Hussain1, and S. Samad1, "Comparative survey on traffic sign detection and recognition: a review," *Przegląd Elektrotechniczny*, vol. 1, pp. 40–44, 12 2015.
- [13] C. Yao, F. Wu, H.-j. Chen, X.-l. Hao, and Y. Shen, "Traffic sign recognition using hog-svm and grid search," in *2014 12th International Conference on Signal Processing (ICSP)*, 2014, pp. 962–965.
- [14] J. Greenhalgh and M. Mirmehdi, "Real-time detection and recognition of road traffic signs," *IEEE Transactions on Intelligent Transportation Systems*, vol. 13, no. 4, pp. 1498–1506, 2012.
- [15] P. Sermanet and Y. LeCun, "Traffic sign recognition with multi-scale convolutional networks," in *The 2011 International Joint Conference on Neural Networks*, 2011, pp. 2809–2813.
- [16] J. Jin, K. Fu, and C. Zhang, "Traffic sign recognition with hinge loss trained convolutional neural networks," *IEEE Transactions on Intelligent Transportation Systems*, vol. 15, no. 5, pp. 1991–2000, 2014.
- [17] Y. Zeng, X. Xu, Y. Fang, and K. Zhao, "Traffic sign recognition using deep convolutional networks and extreme learning machine," in *Intelligence Science and Big Data Engineering. Image and Video Data Engineering*, X. He, X. Gao, Y. Zhang, Z.-H. Zhou, Z.-Y. Liu, B. Fu, F. Hu, and Z. Zhang, Eds. Cham: Springer International Publishing, 2015, pp. 272–280.
- [18] C. Zhou and R. Paffenroth, "Anomaly detection with robust deep autoencoders," 08 2017, pp. 665–674.
- [19] K. Bonawitz, H. Eichner, W. Grieskamp, D. Huba, A. Ingerman, V. Ivanov, C. Kiddon, J. Konečný, S. Mazzocchi, H. McMahan, T. Overveldt, D. Petrou, D. Ramage, and J. Roselander, "Towards federated learning at scale: System design," 02 2019.
- [20] S. Misra, S. Chatterjee, and M. S. Obaidat, "On theoretical modeling of sensor cloud: A paradigm shift from wireless sensor network," *IEEE Systems Journal*, vol. 11, no. 2, pp. 1084–1093, 2017.
- [21] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-Efficient Learning of Deep Networks from Decentralized Data," ser. Proceedings of Machine Learning Research, 2017, pp. 1273–1282.
- [22] W. Y. B. Lim, N. C. Luong, D. T. Hoang, Y. Jiao, Y. C. Liang, Q. Yang, D. Niyato, and C. Miao, "Federated learning in mobile edge networks: A comprehensive survey," *IEEE Communications Surveys Tutorials*, vol. 22, no. 3, pp. 2031–2063, 2020.
- [23] D. P. Kingma and J. Ba, "Adam: a method for stochastic optimization. CoRR."