EVENT: Real-time Video Feed Anomaly Detection for Enhanced Security in Autonomous Vehicles

Georgios Aivatoglou, Nikolaos Oikonomou, Georgios Spanos,
Kristina Livitckaia, Konstantinos Votis, Dimitrios Tzovaras
Information Technologies Institute (ITI)
Center for Research & Technology Hellas (CERTH)
Thessaloniki, Greece
{*aivatoglou, nikosoik,gspanos,klivit,kvotis,Dimitrios.Tzovaras*}*@iti.gr*

***Abstract*—Autonomous Vehicles have long leveraged Artificial Intelligence to be capable of self-driving without the need for a human supervisor. To achieve self-driving autonomy, various sensors are installed onboard the vehicle in order to be able to perceive information from its surroundings. However, since autonomous vehicles' capabilities rely heavily on sensor readings, various challenges arise in terms of security and privacy. Thus, it is of the essence to design methodologies able to detect anomalies caused by malicious threat actors or sensor malfunctions. This paper proposes an anomaly detection algorithm for autonomous vehicle camera sensors. By utilizing Recurrent Neural Networks in combination with Convolution operations, it is possible to obtain a sequence of images and reconstruct the next frame in real-time. By leveraging image similarity techniques such as Mean Squared Error and Structural Similarity Index, it is possible to compare the ground truth with the predicted image and draw conclusions about whether an anomaly is present. The experiments in real datasets captured from autonomous vehicles within the European-funded nIoVe project highlighted that the proposed framework is able to detect anomalies and malfunctions with high accuracy, clearly indicating the necessity of such algorithms to enhance the security of autonomous vehicles.**

***Index Terms*—Autonomous Vehicles, Artificial Intelligence, Cyber Security, Deep Neural Networks, Image Reconstruction, Recurrent Neural Networks, Convolution**

1. Introduction

Autonomous vehicles constitute a breakthrough in the research field of urban transportation, aiming to make transport safer, greener, and more efficient [1]. Additionally, the Internet of Things (IoT) paradigm expanded, covering the entire area of Cooperated Connected and Autonomous Vehicles (CCAV) [2]. As a result, the term Internet of Vehicles (IoV) has emerged to encompass the CCAV area.

Due to the burgeoning growth of IoV technologies and the increasing volume of communications, it is indisputable that many related security and safety threats also rise proportionally, introducing new cyber security vulnerabilities and concerns towards a considerable number of AV and IoV systems [3]. Consequently, ensuring the safety and security of road traffic participants and stakeholders becomes indispensable.

It is evident that the security and safety of AVs start from the detection of anomalies in sensors, either caused by malfunction or intentional. This finding reasoned the purpose of our research study presented in this paper, aiming to enhance the cyber security of AVs by suggesting a novel defense mechanism to detect real-time camera anomalies. The proposed methodology combines Recurrent Neural Networks (RNN) and convolution operations to construct and predict the camera's next frame and compare it with the actual frame using Mean Squared Error (MSE) and Structural Similarity Index (SSI) to detect possible anomalies. Our solution is validated for its performance and robustness in real automated vehicle datasets and simulated datasets within the scope of the nIoVe project[1].

The rest of the paper is organized as follows. The following section defines the related work in the research field, followed by the details for the dataset. Next, the methodology is described and the results are presented. Finally the last section discuss the conclusions and the future work.

2. Related Work

As mentioned earlier, there is an imperative need to evolve as dynamically as possible AV cyber security solutions and technologies against sensor attacks. For this reason, many researchers suggest novel approaches and countermeasures. A representative example is the research study of Sun and Cao [4] introduced the first study that exploited the general vulnerability of LiDAR-based perception and the corresponding detection method called CARLO. In this study, the authors initially validated the success rate of black-box spoofing attacks, which was particularly high (around 80%). Finally, the authors tested their countermeasure CARLO, which uses occlusion patterns as invariant physical features to detect anomalies, reducing the success rate of the spoofing attacks from 80% to 5.5%.

1. https://www.niove.eu

DOI: 10.1109/MED59994.2023.10185766

Continuing with defense mechanisms against AV camera attacks, which is also related to the scope of the study presented in this paper, Lagraa et al. [5] proposed a real-time attack detection system in robot cameras of a self-driving car to enhance its security. Their defense approach is based on image comparisons and unsupervised statistical anomaly detection. The authors evaluated their system against ten attack scenarios using a publicly available dataset, resulting in the high performance of proposed anomaly detection systems. Moreover, the authors in [6] finalized the paper above by presenting and analyzing the deep learning aspect (autoencoders) of their methodology to reconstruct the image. Finally, within the scope of the CARAMEL project, similar to the nIoVe project, Kyrkoy et al. [7] suggested a defense mechanism against camera sensor attacks. The researchers studied the impact of attacks on the camera sensor in a simulated environment created by the CARLA simulator and proposed certain mitigation actions to strengthen the cyber security of AVs against such attacks.

The proposed methodology is in line with the the related work concerning unsupervised methodology consideration, combining the prowess of deep learning methodologies with similarity metrics. The main contribution and differentiation of this study from other related work is an evaluation of such methods in a real-time application (demonstrated within the nIoVe project), validated thoroughly in real and simulated environments also considering timely detection.

3. Dataset

ONCE (One millioN sCenEs) dataset [8] was utilized for the training process. The data was collected in multiple cities in China over a three-month period, providing various road, weather, and lighting conditions. Specifically, the ONCE dataset contains 63.8% sunny scenes, rainy scenes account for 6.11%, and cloudy scenes are 30.09% of the entire scenes, strengthening the diversity of encountered conditions for Machine Learning (ML) tasks.

A multi-sensor platform was employed to capture data, including seven cameras in a 360-degree setup covering all angles of the vehicle. The original camera data was recorded with 10 frames per second (FPS) frequency at a 1920×1020 frame resolution. The camera data was later downsampled to 2 FPS, and then the distortions were removed to enhance the quality of the images resulting in a total of 32.483 compressed JPEG images per camera.

4. Methodology

In this section the proposed methodology to detect anomalies is described. The main consideration of the suggested approach is to combine deep learning techniques to construct the estimated next camera frame and compare it with the actual received frame. Hence, in case of a camera cyberattack, having as target the injection of fake frames or its blindness, the proposed methodology expects to be able to detect it.

4.1. Data Preprocessing

The selected methodology takes an input of 5 sequential images; therefore, chunks of the training data were produced using a sliding window. Specifically, the training set was created with fragments of 5 sequential images, shifting the set by one frame to create artificially additional input training fragments. On top of that, the target chunks are derived by shifting each input chuck by one additional frame, ensuring that the model learns how to make a next-frame reconstruction.

Further, preprocessing techniques were applied to the images. Each image was first resized to a 64 by 64 pixel image, retaining the original aspect-ratio, and subsequently, each pixel value was divided by 255 to normalize pixel values to the $0 - 1$ range.

4.2. Experimental Setup

The proposed algorithm was trained on the ONCE dataset utilizing frames from the back-facing camera. Moreover, experimentation was conducted with various values for the batch size, learning rate, and filters of the ConvLSTM layers. For the hyper-parameter tuning, the optimal batch size was found to be 10, while the learning rate was 0.001. Additionally, 64 filters were selected for the ConvLSTM layers having the best results for the specific use case. Adadelta [9], a well-established technique, was used for weight optimization through back-propagation; and binary cross entropy [10] was experimentally selected as the loss function. Additionally, a Dropout layer [11] was implemented after each ConvLSTM, with a dropout probability of 50% to tackle any over-fitting issues.

For the training process, the ONCE dataset was split into training, validation, and test sets. Specifically, the training set has a shape of $1994 \times 5 \times 64 \times 64$, and the validation set and test set $1328 \times 5 \times 64 \times 64$ and $3167 \times 5 \times 64 \times 64$, respectively. Such an approach was followed to select the best model during the training process and then validate the best model on the test set. To further support the model, the learning rate was reduced by a factor of 10 once the learning process started stagnating. As an additional regularization technique, an early stopping strategy was adopted when the validation loss stopped decreasing for more than 5 epochs.

The experiments were conducted on a computer with 2 $\times$ Tesla K40 (12GB memory), 256 GB of RAM, and an Intel Xeon E5-2630 processor.

4.3. Architecture

The proposed architecture is based on Convolutional LSTMs (ConvLSTM) [12], which are Recurrent Neural Networks (RNN) built for spatio-temporal learning. ConvLSTMs utilize convolutional operations in input-to-state and state-to-state transitions to capture better spatio-temporal correlations in data. The proposed approach was found experimentally and consists of 3 ConvLSTM ($2D$) layers and a $3D$ Convolution layer for a total of 742465 trainable parameters.

By keeping the filter movement (stride) to 1 and adding zeros evenly to the top/bottom and left/right (padding) of the input, we were able to keep the input image resolution to the original size since the goal of the algorithm is the next frame prediction. Moreover, we utilized 64 filters for all the $2D$ ConvLSTM layers and a kernel size of 3×3. Considering the

non-linearity nature of the problem, the ReLu [13] activation function was applied for all the ConvLSTM layers offering the model capabilities of learning more complex patterns.

Finally, the output layer consists of a 3D Convolutional layer with a filter size of 1, a kernel size of $3\times3\times3$, and the sigmoid activation function. To tackle over-fitting issues and help the model converge faster, a batch normalization [14] and a dropout layer with a dropout probability of 50% was applied after each convolution.

As previously mentioned, an early stopping strategy was utilized by monitoring the validation loss. The algorithm's learning rate was reduced by 10% whenever the validation loss stopped decreasing for a total period of 10 sequential epochs. Finally, the L2 regularization [15] was applied as a weight decay to overcome over-fitting issues by penalizing large weights. The architecture design follows the standard strategy used in image processing problems. The complete processing pipeline based on the proposed methodology is shown in Figure 1.

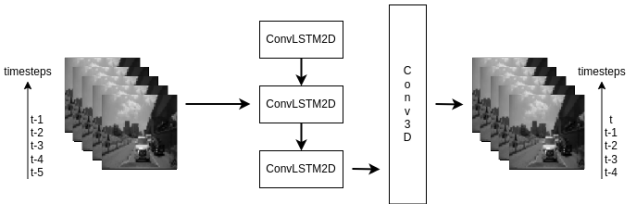

Figure 1. Architecture based on the proposed methodology.

4.4. Similarity Metrics

For the evaluation of the proposed methodology, 2 different metrics were used - the Mean Squared Error (MSE) and the Structural Similarity Index [16] (SSIM). In contrast with MSE, which can be viewed as a basic approach to comparing images, SSIM is more indicative of image perceived similarity since it takes into account the texture [17].

Specifically, MSE takes the square of differences between every pixel in the two images, sums them up, and divides them by the number of total pixels. Evidently, the two images have to be of the same shape. Therefore, the result is an approximate comparison between the two images.

On the other hand, SSIM, as denoted by Wang et al. [16], extracts three different features from an image: luminance, contrast, and structure. The two input images are compared based on these 3 facets.

An inference example based on the proposed methodology is depicted in Figure 2. As illustrated, in the first (upper) row, five sequential images are used as input to the model, and the second (lower) row shows the ground truth (GT), the prediction, and the injected image. Figure 2 shows two similarity metrics calculated with the corresponding scores. On the one hand, the similarity metrics show that the MSE score is less when comparing the ground truth with the prediction than when comparing the prediction with the injection. On the other hand, the SSIM score is greater for the first comparison indicating the higher similarity between the ground truth and the prediction.

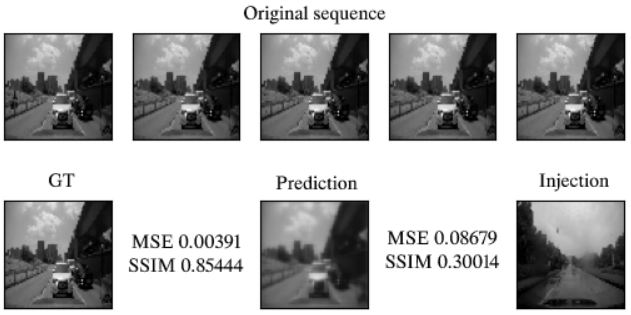

Figure 2. Inference example based on the proposed methodology.

5. Results and Discussion

Three different datasets with diverse characteristics were used to evaluate the algorithm in anomaly detection to better understand the model's generalization capabilities. Specifically, the datasets were captured from various cameras fronting different directions, and the images had different resolutions and frames-per-second rates. The same preprocessing techniques were used for all the test sets before inputting following the proposed architecture.

The first test set was derived directly from the ONCE [8] dataset containing 15.835 images from the onboard back-facing camera of the vehicle. The second test set was captured using AV at the Centre for Research and Technology - Hellas (CERTH) premises through the vehicle's front-facing camera, containing 2.000 images. Finally, the third test set was captured using a tram vehicle at Transports Publics Genevois (TPG) premises, containing 295 images from a camera placed between two wagons and pointing downwards to the tracks (to avoid any personal data collection and processing).

For the algorithm evaluation, the F1-macro score was calculated [18]. Since False Negatives and False Positives are crucial considering the scenario of an altered camera stream, the F1-score metric was considered essential. Moreover, due to the imbalanced nature of the real-world problems and the non-equal distribution between the classes, the F1-score is a more appropriate evaluation metric in contrast to the accuracy [19]. Thus, since the correct predictions of the samples belonging to the abnormal classes were of crucial meaning, accuracy was not the optimal metric for the validation tests. Specifically, the F1-score is formulated as the combination of the Precision and Recall using their harmonic mean. Precision equals the number of True Positives (TP) divided by the number of False and True Positives (FP, TP). On the other hand, Recall equals the number of True Positives (TP) divided by the number of True Positives (TP) and False Negatives (FN).

As discussed, the developed model can predict the next frame by taking the previous five frames as input. Hence, by reconstructing the next frame, the algorithm can compare and thus determine whether there is an anomaly. Since we used MSE and SSIM metrics, it was essential to set a threshold for comparing images. A threshold calibration technique was developed using 10% of each test set to avoid a fixed threshold for different test sets. Considering this, it

was possible to test different test sets that utilized various cameras and even different frame rates.

The second crucial threshold for the proposed methodology in the algorithm evaluation is the ”how-past” threshold, which determines how old the injected images will be in the video stream to be considered attacks. Although different comparisons between different values were performed, an optimal value of 150 frames back was selected. Thus, all the results discussed in the following subsections are considered with a how-past value of 150 frames.

5.1. ONCE Test Data

The first test set used to evaluate the developed model was part of the ONCE dataset having a frame-rate of 2 FPS. Since the interest of this research is to validate the model based on the proposed methodology for abnormal camera streams, a mechanism that can inject frames into the normal camera stream was developed. Specifically, the injected frames were black, white, random RGB color, and past frames and were randomly positioned into the normal camera stream of the vehicle. Thus, it was feasible to test the model on anomaly detection tasks by injecting abnormal frames. The aforementioned reasoning for the attacks was inspired by the results of [6, 5].

Specifically, for the past frames class, a critical decision related to the ”how-past” parameter was the images that had to be injected into the normal camera stream. The ”how-past” was one of the most crucial parameters of the methodology and was closely related to the algorithm’s accuracy. Hence, various values ranging from 5 to 150 were tested, and the results can be found in Figure 3.

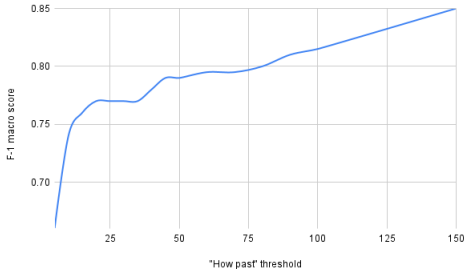

Figure 3. ONCE test set: ”how-past” vs F1-macro score.

As clearly depicted in Figure 3, by increasing the ”how-past” parameter, the F1-macro accuracy of the algorithm increases as well. As expected, class ”past frames” accuracy was closely related to the ”how-past” parameter. Evidently, the closely distanced frames will have considerable similarities and, therefore, low MSE and high SSIM values. Considering that the ONCE dataset includes frames with a rate of 2 FPS, anything below 25 frames back corresponds to 12.5 seconds back in time. Hence, the differences between the frames are ambiguous, especially considering the moments when the vehicle is completely stopped due to traffic lights or pedestrian crossings.

As discussed, to have consistency between the different test sets for the model evaluation, a fixed ”how-past” threshold was selected. That threshold was set to 150 frames back since that value was an optimal solution leading to accurate classification results for all the sets.

TABLE 1. ONCE TEST SET: RESULTS.

Class	Detected	Total	Accuracy
Normal	1279	1350	94.74%
White	17	17	100%
Black	20	20	100%
Random	19	19	100%
Past	16	19	84.21%

Table 1 shows the results of the algorithm on the ONCE test set. Since the algorithm works as a binary classifier of whether an incident was an anomaly, everything out of the Normal class was considered an anomaly and was further labeled with a specific injection type manually. Hence, the overall accuracy of the algorithm for that particular test set was 95.8%, while the F1-macro score was 85%. It should be noted that since the algorithm detects possible anomalies without any further information about the specific class of a possible attack, we randomly selected the class ”P” as the class for the False Positives predictions of the ”N” class.

The results showed that the algorithm could find all the injected anomalies of the White, Black, and Random classes accurately, while for the Past class, it misclassified 3 samples. Further, 71 False Positives were identified as anomalies. That can be evidently interpreted since we calibrated the frame comparison metrics only with the first 10% of the corresponding test set. Thus, the characteristics of the first frames (e.g., roads, traffic lights, tunnels, landscape, etc.) may not represent the rest of the trip.

5.2. CERTH Test Data

To better test the generalization capabilities of the developed model based on the proposed methodology, we tested it on a completely different dataset. Specifically, we utilized CERTH AV with a front-facing camera pointing toward the road. Moreover, the new test set was about 6 FPS, three times higher than the ONCE test set. The AV employed a front-facing camera, and the recorded data included a round trip at CERTH premises. Considering different camera settings and the utterly different FPS rate, we experimented with testing the proposed model on new and unseen data.

Similarly to the procedure with the ONCE test set, we had to inject abnormal frames in the camera stream. Thus, we utilized the exact mechanism, injecting Black, White, Random RGB color, and Past frames. The images were randomly positioned in the camera stream to simulate better a possible attack on the vehicle’s onboard camera. Figure 4 indicates the F1-score in contrast to the ”how-past” parameter. It is explicit that the algorithm score is highly correlated with the parameter ”how-past”, which denotes the position of the injected images with respect to time.

The results of the algorithm execution on the CERTH test set are described in Table 2. In general, the algorithm could identify the anomalies of all the abnormal classes denoting its proper generalization.

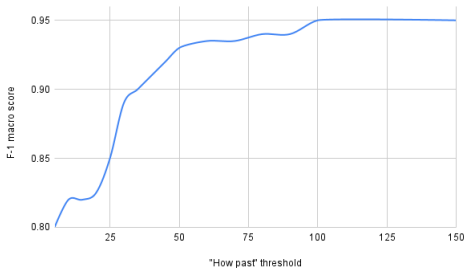

Figure 4. CERTH test set: ”how past” vs F1-macro score.

TABLE 2. CERTH TEST SET: RESULTS.

Class	Detected	Total	Accuracy
Normal	1579	1800	87.72%
White	15	15	100%
Black	17	17	100%
Random	13	13	100%
Past	16	16	100%

5.3. TPG Test Data

The final test was conducted with a tram test set with a frame rate of 10 FPS provided by TPG. The TPG dataset was utterly different from the previous two test sets concerning the camera’s positioning. Specifically, the camera was onboarded between two trains of the tram and was pointing down.

Figure 5 shows the F1-score of the ”how past” parameter. In contrast to the previous test sets, by injecting frames that were as close as 25 frames back in time, the model achieved nearly 94% F1-macro score. The results of the last test set are detailed in Table 3.

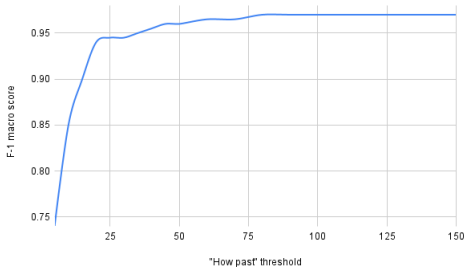

Figure 5. Tram test-set: ”How past” vs F1-macro score.

TABLE 3. TPG TEST SET: RESULTS.

Class	Detected	Total	Accuracy
Normal	141	265	53.21%
White	21	21	100%
Black	19	19	100%
Random	19	19	100%
Past	13	13	100%

The developed model was evaluated on three different and diverged test sets to assess the model’s generalization capabilities. Overall, the model could reconstruct the next frame of a sequence with high accuracy regardless of the test set characteristics. The overall results are described in Table 4.

TABLE 4. OVERALL RESULTS ON THE THREE TEST SETS.

Test set	FPS	Angle/Position	F1 score
ONCE	2	backward	85%
CERTH	6	frontward	81%
TPG	10	downward	77%

5.4. Real-Time Anomaly Detection Response Time

This section discusses the inference response time of the model. Table 5 displays the maximum, minimum, mean, and standard deviation values for each of the three test sets. The depicted inference time encapsulates both the next frame prediction and the error calculation processes for the anomaly detection task. Thus, the entire inference process is timed, from the new frame’s arrival to the final decision of whether there is an anomaly.

TABLE 5. INFERENCE TIME PER TEST SET IN SECONDS.

Test set	Max	Min	Mean	SD
ONCE	0.04712	0.04278	0.04325	0.00024
CERTH	0.04784	0.04283	0.04331	0.00021
TPG	0.04385	0.04246	0.04279	0.00017

Emphasis has been given to the inference response time of the model due to the target device executing the inference process being an onboard computer with limited hardware capacity. Several optimizations were required to ensure accurate and on-time predictions. Namely, the trained model was converted to TensorFlow Lite using the default TensorFlow optimization set and selecting the target spec with FLOAT16 support for weight quantization to minimize CPU utilization and memory consumption during the execution of the inference process. The converted TensorFlow Lite model resulted in a significant reduction of memory consumption without sacrificing detection accuracy.

6. Conclusions and Future Work

Our research efforts focused on developing an anomaly detection framework for autonomous vehicles. The data used for training in this study was derived from the ONCE dataset, containing images from a back-facing onboard camera on the vehicle. Furthermore, the model was tested on three different test sets and the results highlighted its performance. More specifically, by employing various test sets with different characteristics, such as the position or angle of the camera and the FPS rate, we were able to evaluate the generalization capabilities of the model in diverse real-world settings.

Future work concerns deeper analysis and implementation, which could further benefit the proposed methodology

including the exploitation of more sensors. The exploitation of more camera positions and angles is another branch of future work continuation. In this study, we experimented with the onboard back camera of the autonomous vehicle for the training process. Although we achieved a very good model generalization, further experiments must be included to test the architecture's benefits and generalization capabilities when trained with rear or side cameras. For this reason, the work performed and the corresponding results extracted under the nIoVe project could be the input and the basis for the ULTIMO project in order to enhance the cybersecurity of autonomous vehicles against camera attacks.

Acknowledgments

We would like to acknowledge the support provided by Transports Publics Genevois (TPG) and especially Quentin Matthewson for gathering and supply of tram data. Their contribution greatly improved the quality of the manuscript giving us the opportunity to test further the generalization capabilities of the developed model.

Funding

This work is partially funded by the European Union's Horizon 2020 Research and Innovation Programme through the nIoVe project[2] (A Novel Adaptive Cybersecurity Framework for the Internet-of-Vehicles) under Grant Agreement No. 833742 and the ULTIMO project[3] (Advancing Sustainable User-centric Mobility with Automated Vehicles) under Grant Agreement No. 101077587. This paper reflects only the authors' view; the European Union is not liable for any use that may be made of the information contained therein.

References

[1] Zhenyu Zhou et al. "When vehicular fog computing meets autonomous driving: Computational resource management and task offloading". In: *IEEE Network* 34.6 (2020), pp. 70–76.

[2] Jianhua He et al. "Cooperative connected autonomous vehicles (CAV): research, applications and challenges". In: *2019 IEEE 27th International Conference on Network Protocols (ICNP)*. IEEE. 2019, pp. 1–6.

[3] Priyank Sharma, Meet Patel, and Apoorva Prasad. "A systematic literature review on Internet of Vehicles Security". In: *arXiv preprint arXiv:2212.08754* (2022).

[4] Jiachen Sun et al. "Towards robust {LiDAR-based} perception in autonomous driving: General black-box adversarial sensor attack and countermeasures". In: *29th USENIX Security Symposium (USENIX Security 20)*. 2020, pp. 877–894.

[5] Sofiane Lagraa et al. "Real-time attack detection on robot cameras: A self-driving car application". In: *2019 Third IEEE International Conference on Robotic Computing (IRC)*. IEEE. 2019, pp. 102–109.

[6] Faouzi Amrouche et al. "Intrusion detection on robot cameras using spatio-temporal autoencoders: A self-driving car application". In: *2020 IEEE 91st Vehicular Technology Conference (VTC2020-Spring)*. IEEE. 2020, pp. 1–5.

[7] Christos Kyrkou et al. "Towards artificial-intelligence-based cybersecurity for robustifying automated driving systems against camera sensor attacks". In: *2020 IEEE Computer Society Annual Symposium on VLSI (ISVLSI)*. IEEE. 2020, pp. 476–481.

[8] Jiageng Mao et al. "One Million Scenes for Autonomous Driving: ONCE Dataset". In: 2021.

[9] Matthew D Zeiler. "Adadelta: an adaptive learning rate method". In: *arXiv preprint arXiv:1212.5701* (2012).

[10] Usha Ruby and Vamsidhar Yendapalli. "Binary cross entropy with deep learning technique for image classification". In: *Int. J. Adv. Trends Comput. Sci. Eng* 9.10 (2020).

[11] Nitish Srivastava et al. "Dropout: a simple way to prevent neural networks from overfitting". In: *The journal of machine learning research* 15.1 (2014), pp. 1929–1958.

[12] Xingjian Shi et al. "Convolutional LSTM network: A machine learning approach for precipitation nowcasting". In: *Advances in neural information processing systems* 28 (2015).

[13] Abien Fred Agarap. "Deep learning using rectified linear units (relu)". In: *arXiv preprint arXiv:1803.08375* (2018).

[14] Sergey Ioffe and Christian Szegedy. "Batch normalization: Accelerating deep network training by reducing internal covariate shift". In: *International conference on machine learning*. PMLR. 2015, pp. 448–456.

[15] Andrew Y Ng. "Feature selection, L 1 vs. L 2 regularization, and rotational invariance". In: *Proceedings of the twenty-first international conference on Machine learning*. 2004, p. 78.

[16] Zhou Wang et al. "Image quality assessment: from error visibility to structural similarity". In: *IEEE transactions on image processing* 13.4 (2004), pp. 600–612.

[17] Umme Sara, Morium Akter, and Mohammad Shorif Uddin. "Image quality assessment through FSIM, SSIM, MSE and PSNR—a comparative study". In: *Journal of Computer and Communications* 7.3 (2019), pp. 8–18.

[18] Margherita Grandini, Enrico Bagli, and Giorgio Visani. "Metrics for multi-class classification: an overview". In: *arXiv preprint arXiv:2008.05756* (2020).

[19] Marina Sokolova, Nathalie Japkowicz, and Stan Szpakowicz. "Beyond accuracy, F-score and ROC: a family of discriminant measures for performance evaluation". In: *Australasian joint conference on artificial intelligence*. Springer. 2006, pp. 1015–1021.

2. https://www.niove.eu/
3. https://ultimo-he.eu/