2022 IEEE Intelligent Vehicles Symposium (IV)
June 5-9, 2022. Aachen, Germany

2022 IEEE Intelligent Vehicles Symposium (IV) | 978-1-6654-8821-1/22/$31.00 ©2022 IEEE | DOI: 10.1109/IV51971.2022.9827143

Traffic Sign Classifiers Under Physical World Realistic Sticker Occlusions: A Cross Analysis Study

Yasin Bayzidi[1,2], Alen Smajic[1], Fabian Hüger[1], Ruby Moritz[1],
Serin Varghese[1], Peter Schlicht[1] and Alois Knoll[2]

Abstract— **Recent adversarial attacks with real world applications are capable of deceiving deep neural networks (DNN), which often appear as printed stickers applied to objects in physical world. Though achieving high success rate in lab tests and limited field tests, such attacks have not been tested on multiple DNN architectures with a standard setup to unveil the common robustness and weakness points of both the DNNs and the attacks. Furthermore, realistic looking stickers applied by normal people as acts of vandalism are not studied to discover their potential risks as well the risk of optimizing the location of such realistic stickers to achieve the maximum performance drop. In this paper, (a) we study the case of realistic looking sticker application effects on traffic sign detectors performance; (b) we use traffic sign image classification as our use case and train and attack 11 of the modern architectures for our analysis; (c) by considering different factors like brightness, blurriness and contrast of the train images in our sticker application procedure, we show that simple image processing techniques can help realistic looking stickers fit into their background to mimic real world tests; (d) by performing structured synthetic and real-world evaluations, we study the difference of various traffic sign classes in terms of their crucial distinctive features among the tested DNNs.**

I. Introduction

In the recent years, various traffic sign recognition systems are introduced, which are developed using DNNs [2], [3], [4], [5], [6], [7]. However, the proposed DNN based methods have been vulnerable against changes in the input data which could occur due to aging, vandalism in different ways such as destruction, stickers, paintings and graffiti or adversarial attacks [8], [9], [10], [11], which often result in bad detection even by human drivers. Consequently, ensuring the reliability of such detectors is requisite before their deployment, as it could lead to catastrophic hazards otherwise.

Based on that, there is a need to evaluate such DNNs under challenging conditions such as adversarial or non-adversarial augmentations that could potentially lead to their misbehaviour. It is essential for such an evaluation to provide a controlled environment, where one can conduct the severity of such augmentations. Furthermore, such an analysis could extract possible common sensitive spots in the traffic signs that, when occluded, could affect the detection accuracy of different DNNs similarly. This can help to conclude if it is possible for the attackers to optimise their attacks on common DNNs and apply them on traffic signs, hoping for misleading the deployed DNNs in the vehicles.

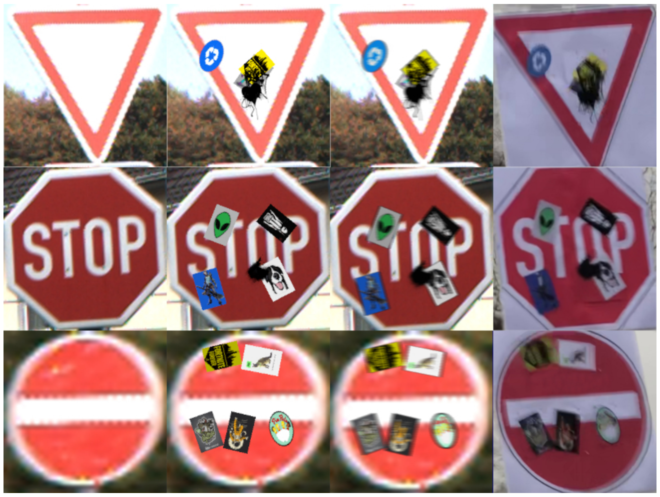

Fig. 1: **Examples of sticker application** with and without our normalization techniques: *from left to right*: 1) the input images from GTSRB [1] without any augmentations; 2) the input images covered with stickers before the proposed adaptations; 3) the input images covered with stickers after being adapted using our adaptation methods; 4) the real images of the same sticker overlay from our field tests.

However, performing such a thorough analysis requires an ordered data-set, in which one can query the data by different severity levels of the attacks. As such a data-set is almost impossible to collect, one can propose to apply such occlusions synthetically, which come with artificial artifacts that could mislead the DNNs due to domain shifts caused by such augmentations, which do not necessarily happen in real world. In order to minimize those artifacts, one needs to normalize the occlusion patches to match the individual images. To the best of our knowledge, there is no controlled environment yet introduced to test the robustness of the traffic sign detectors while considering all the challenges discussed above.

In this paper, we report on the robustness of 11 of the state-of-the-art classification networks, while being challenged against realistic sticker occlusions. This is done

[1] Volkswagen AG {yasin.bayzidi, alen.smajic, fabian.hueger, ruby.moritz}@volkswagen.de, {serin.varghese, peter.schlicht}@cariad.technology,
[2] Technical University of Munich yasin.bayzidi@tum.de, knoll@in.tum.de

by systematically applying stickers from a sticker pool to the traffic signs in order to extract their sensitive regions. To do so, alongside to random application of stickers (RSA), we introduce two methods for finding such regions, namely Saliency-based Sticker Application (SSA) and Monte Carlo Sticker Application (MCSA). Moreover, we apply and test adversarial attacks from the literature in real world along with our proposed methods.

Our contributions are as follows:

- A framework to adapt synthetic stickers to real traffic sign images based on brightness, contrast and blur.
- RSA: the effects of random sticker application on state-of-the-art traffic sign classifiers.
- SSA and MCSA: two sticker application methods to identify the sensitive regions in the input images.
- Cross-DNN analysis: extending the sticker attacks to 11 networks and cross analyse all to recognise the common weakness/strength properties.
- Deployment check: field tests of different sticker application methods in the real world.
- A sign coverage metric to measure the occlusion area as a severity metric.

The rest of the paper is organized as follows: in Section II, the related literature are reviewed; in Section III, our method of adapting synthetic stickers to traffic signs along with our sticker localisation methods are introduced; in Section IV, the experiment setup as well as the results are discussed; finally, the summary of the paper and the main conclusions are reviewed in the final Section V.

II. Related Work

The recent advances in traffic sign recognition are based on DNNs [2], [3], [4], [5], [6], [7], which are specifically prone to be sensitive to changes in the input data. These changes include damage or occlusion occurred to the objects in the real world or to the data during the capturing. Based on that, the challenges of traffic signs detection, such as occluding objects due to camera angel, damage, and aging are reviewed in [12], [13], [14] and different methods to overcome such challenges are suggested. Moreover, multiple attacks are introduced to mislead traffic sign detectors, which mostly rely on occlusion patterns added to the traffic signs [8], [9], [10], [11]. Eykholt et.al. [8] introduced RP2 as a two stage attack, which first localized the sensitive spots and then defined adversarial patches in those regions capable of deploying to the physical world. Similar to that, Brown et.al. [15] proposed adversarial patch (AP), which is a one stage attack that did not rely on specific regions or classes and was therefore location, rotation and scale invariant. However, such methods rely on introducing highly salient textures and are barely tested in real world conditions. For example. they are either tested on very shallow DNNs, such as RP2 [8], or under perfect lighting conditions and very close to the camera, such as AP [15], which is not the case with traffic sign detection that involves far distances, high speeds, uneven road surfaces, or bad weather conditions. Based on that, we argue that deploying such highly salient features to the real world, and preserving them while capturing the images is difficult, which causes such attacks to fail in such challenging environments.

On the other side, few papers suggested to build testing environments, either through image augmentations or real world test tracks to test the traffic sign detectors under the aforementioned challenging conditions [16], [17], [18], [19]. However, none of the aforementioned papers introduced a generic testing environment that conducted the severity of such challenges alongside with cross analysis of different detectors to find commonalities as weakness or strength points of common detectors.

III. Method

The goal of this paper is to study the effect of occlusions caused by realistic stickers on the common traffic sign classifiers, which are normally applied as acts of vandalism without any intention of misleading automated driving (AD) systems. However, they can be applied intentionally on crucial features to mislead the bespoken systems (e.g. to cover the 3 in 30km/h speed limit sign.). In this paper, we introduce methods to overcome the challenges of realistic sticker application mentioned before, and cross analyse multiple state-of-the-art architectures to find commonalities among them. In the following, our approach for adapting the stickers to fit into the traffic signs are explained, which is followed by three methods of realistic sticker localization on traffic signs.

A. Realistic Sticker Occlusions

Applying realistic stickers directly to traffic signs images would cause unrealistic outputs stemming from the diversity in individual image properties such as contrast, brightness, and sharpness. To overcome such a challenge, we employ conventional image processing techniques to extract the input image properties and use for adapting the stickers to fit the input images. To do so, we extract and apply three properties as follows:

Brightness: To estimate the overall brightness of an image, the average of all the pixel intensities is calculated as

$$b_{\mathbf{x}} = \frac{1}{|\mathcal{I}|} \sum_{i \in \mathcal{I}} x_i, \tag{1}$$

where $\mathbf{x} \in \mathbb{G}^{H \times W \times C}$ is the input image with height H, width W, number of color channels C, and $\mathbb{G} = \{0, 1, 2, \ldots, 255\}$. The pixel position i of the input image $\mathbf{x}$ is defined as $i \in \mathcal{I} = \{1, \ldots, H \cdot W \cdot C\}$, and $|\mathcal{I}| = H \cdot W \cdot C$. Based on that, the intensity of the sticker overlay $\mathbf{o}$ is adapted as follows:

$$\mathbf{o}_{\mathbf{x}} = \frac{b_{\mathbf{x}}}{\lambda} \mathbf{o}, \tag{2}$$

where $\mathbf{o}_{\mathbf{x}}$ is the sticker overlay adapted to match the image $\mathbf{x}$ using the brightness $b_{\mathbf{x}}$ calculated based on Equation (1). The $b_{\mathbf{x}}$ is first normalized by dividing by λ which is the normalizing factor, leading to a scalar value in range $[0, 2]$. Multiplying this scalar by the sticker overlay $\mathbf{o}_{\mathbf{x}}$ will lead to an increase in the brightness of $\mathbf{o}_{\mathbf{x}}$ if it is above 1 and vice versa if it is below 1.

Contrast: To adapt the contrast of the sticker overlay $\mathbf{o}$ to the contrast of an image $\mathbf{x}$, we use the standard deviation as follows:

$$\mathbf{o}_{\mathbf{x}} = \frac{\sigma(\mathbf{x})}{\sigma(\mathbf{o})}\mathbf{o}, \tag{3}$$

where $\sigma()$ computes the standard deviation of all the pixel intensities of the given image.

Blur: One can calculate the Laplacian of an image as an indication of blurriness [20], which discovers rapid changes in an image, in other words, edges. The standard deviation of the Laplacian is then taken to rank the sharpness of an image generally, as the Laplacian is calculated locally. Using this, we have:

$$s = \sigma(\Delta(\mathbf{x})), \tag{4}$$

where $\Delta()$ function computes the Laplacian and $\sigma()$ computes the standard deviation. A high s would indicate a sharp image and a low s a blurry one. As the blur factor is applied locally, we define a two dimensional convolution filter of size $\theta - s_{\mathbf{x}} \times \theta - s_{\mathbf{x}}$ which needs to be an odd number, where θ is a hyper-parameter defined as 13. This convolution filter is initialised with zeros except in the middle row, which is initialised with $1/s_{\mathbf{x}}$. This way, a horizontal motion blur is added to the sticker overlay to mimic the motion blur caused by vehicle movement while capturing the video frames. The three aforementioned factors are extracted and applied sequentially to the sticker overlay as a pre-processing step. An example of the sticker applications with and without our normalization method is illustrated in Figure 1, where one can compare the similarity of the adapted stickers and the real world examples of the same stickers.

B. Sticker Localization Methods

The next step towards this process is to control the selection and positioning of the stickers. To do so, a one by one approach is defined, in which after applying a sticker to the traffic sign images, the loss of the DNN is evaluated while the sticker is applied to all the images in the data-set. To do so, we define $\mathbb{S}$ as the set of all the stickers in our sticker pool, $\mathbb{X}$ all the traffic sign images, $\mathbb{Y}$ all the labels used to optimize the sticker positioning algorithms, and $f_{\boldsymbol{\theta}}(\mathbf{x})$ the activation function using the set of parameters $\boldsymbol{\theta}$. Based on that, we define the loss of one sample $x \in \mathbb{X}$ as $\ell_{\mathbf{x}_i}(\boldsymbol{\theta}) = \ell_{\mathbf{x}_i}(f_{\boldsymbol{\theta}}(\mathbf{x}_i), \mathbf{y}_i)$, while i is the index of the selected traffic sign image and its associated label. Having this, the loss of an image while a sticker overlay is applied to it can be calculated as follows:

$$\ell_{\mathbf{x}_i\mathbf{o}}(\boldsymbol{\theta}) = \ell_{\mathbf{x}_i\mathbf{o}}(f_{\boldsymbol{\theta}}(\mathbf{x}_i + \mathbf{o}), \mathbf{y}_i), \tag{5}$$

while $\mathbf{o}$ is the overlay consisting one or more stickers. Adding a new sticker $\mathbf{s}$ to the current sticker overlay $\mathbf{o}$ is donated as $\mathbf{o}_{\mathbf{s}} = \mathbf{o} + s$. To see the effect of adding $\mathbf{s}$, we calculate the average loss increase on all the images in $\mathbb{X}$ as follows:

$$L_{\mathbb{X}_{\mathbf{o}_{\mathbf{s}}}}(\boldsymbol{\theta}) = \frac{1}{n}\sum_{i=1}^{n} \ell_{\mathbf{x}_i\mathbf{o}_{\mathbf{s}}}(f_{\boldsymbol{\theta}}(\mathbf{x}_i + \mathbf{o}_{\mathbf{s}}), \mathbf{y}_i). \tag{6}$$

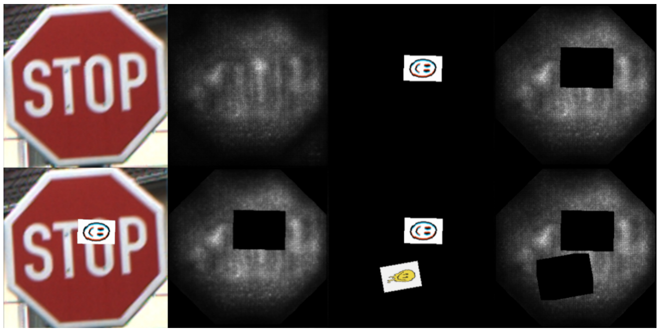

Fig. 2: **Examples of saliency-based sticker application** using saliency maps. *From left to right*: the input image; the saliency extracted using Vanilla [21]; the position of the selected sticker(s); the saliency map excluding the sticker(s) overlay $\mathbf{o}$ and a safe margin around each sticker.

To find the best sticker for the current iteration, the $L_{\mathbb{X}_{\mathbf{o}_{\mathbf{s}}}}(\boldsymbol{\theta})$ is calculated for all the stickers in $\mathbb{S}$ and the one with the highest $L_{\mathbb{X}_{\mathbf{o}_{\mathbf{s}}}}(\boldsymbol{\theta})$ is considered as the best sticker for the current iteration. Based on the number of iterations (i.e. the number of stickers to be added to the sticker overlay $\mathbf{o}$), the process is then repeated to add more stickers having the previous ones already applied. For selection and localisation of the stickers, we have defined three approaches as follows:

Random Sticker Application (RSA): As of its name, this method is just a random application of the stickers, where one selects stickers from our sticker pool randomly and applies them to the traffic signs after the pre-processing mentioned in Section III-A, which is repeated 100 times for each experiment. Furthermore, the traffic sign is then checked to assure that no place on the traffic sign surface is left without being covered with stickers during this process. The main difference of the other two proposed methods with `RSA` is that they aim at increasing the DNN loss. This means, in case of multi-sticker application, the goal would be to find the combination of stickers that achieve the highest loss.

Saliency-based Sticker Application (SSA): In this method, we extract the saliency of each image inferred into the DNN while no stickers are applied to, which returns the most important features of the image, where the DNN has paid attention for its decision. We employed Vanilla [21] to calculate the saliency map $\mathbf{m}$ as follows:

$$\mathbf{m}_{\text{Vanilla}}(\mathbf{x}_0) = \frac{\partial y^{'}}{\partial \mathbf{x}}|_{\mathbf{x}=\mathbf{x}_0}, \tag{7}$$

which is of size $H \cdot W$ and represents the simple derivative of the class score $y^{'}$ with respect to the input image $\mathbf{x}$. Afterwards, we take the $\arg\max$ to extract the peak points of the saliency map as the center points of the stickers. This way, the most important features will be covered with stickers. Furthermore, the scalar values of the saliency map which fall inside the sticker plus a margin around the sticker are removed from the saliency map to avoid multiple stickers to overlap. An example of the SSA for two stickers is shown in Figure 2.

Monte Carlo Sticker Application (MCSA): In this method, all the stickers are applied one by one in random positions, and the one which causes the highest loss gets selected. In other words, to calculate the effect of a new sticker $\mathbf{s}$, it is added to the sticker overlay $\mathbf{o}$ 100 times with 100 random locations on the traffic sign and the one with the highest $L_{\mathbb{X}_{\mathbf{o_s}}}(\boldsymbol{\theta})$ is considered as the best position for that sticker. As mentioned before, $L_{\mathbb{X}_{\mathbf{o_s}}}(\boldsymbol{\theta})$ is then calculated for all the stickers in the $\mathbb{S}$ using the 100 iteration to find the best position for all of the stickers in the current iteration. Finally, the one with the highest $L_{\mathbb{X}_{\mathbf{o_s}}}(\boldsymbol{\theta})$ is selected as the best sticker in the best position for the current iteration. Afterwards, the process is repeated again to choose the second sticker and continues until it reaches the maximum number of applicable stickers defined by the user.

IV. Experiments and Results

A. Experiment Setup

Training: We trained 11 image classification networks including VGG16 and VGG19 [22], InceptionV3 [23], ResNet family with 18, 50, 101 and 152 layers [24], WideResNet50 and WideResNet101 [25], ResNeXt50 and ResNeXt101 [26] on the German Traffic Sign Recognition Benchmark (GTSRB) data-set [1], that includes about $50,000$ images that are split among 43 classes. This way, a broad variation of DNNs in terms of type as well as the number of parameters are tested. The image sizes are 299×299 pixels. We used the Adam optimizer [27] with a learning rate of 0.0001 and a batch size of 64 to train the aforementioned DNNs on four Nvidia Geforce 1080 Ti GPUs for 100 epochs.

Optimisation: We have optimised SSA and MCSA on all of the mentioned DNNs along with RSA averaged over 100 repeats per each DNN. We have implemented and applied the RP2 [8] and AP [15] from the literature as well. These methods are optimised and applied to five classes including 30km/h, yield, stop, no entry, and children crossing, which all represent safety critical classes in three shapes (i.e. triangle, circle, and octagon), and diverse types of critical features, from centralized small ones such as the 30km/h to diverse ones such as the stop sign.

Evaluation Metrics: Besides the mean average precision mAP, we have also introduced a sign coverage metric C, which indicates the area of the traffic sign covered with stickers by the sticker application algorithm. This way, having a lower C is an indication of a stronger attack. C is calculated based on the intersection over union metric as follows:

$$C = \frac{\text{Intersection}(\mathbf{m}, \mathbf{o})}{\text{Union}(\mathbf{m}, \mathbf{o})}, \tag{8}$$

where $\mathbf{m}$ is the class specific mask of the traffic sign and $\mathbf{o}$ is the sticker overlay.

Real World Tests: We conducted outdoor field tests on the MCSA overlays from our methods, as it performed better than RSA and SSA on all the DNNs, and the RP2 [8] and AP [15] from the literature. The videos are captured using a cellphone capable of 4K (i.e. 3840×2160 pixels) video capturing mounted on a vehicle dashboard while driving towards the signs from a range of 50 meters. The accuracy of the DNN under test is first evaluated on a video of the clean traffic sign with a similar setup. Therefore, the results of the field test include both the clean and attacked accuracy of each DNN. However, performing real world field tests on all the combinations of the DNNs, sticker methods, and classes was not feasible for us. Therefore, we hand picked two DNNs for the classes 30km/h, yield, stop and no entry and three DNNs for the class children crossing.

B. Results and Discussion

Comparative Results: We tested our methods both by synthetic attacks through augmentation and field tests in real world. In the first case we only compared our RSA, SSA and MCSA methods, and left the RP2 and AP for the field tests, as their sensitivity to post-processing prevented them to be applied seamlessly to the test images. Based on that, the class based results of RSA, SSA and MCSA are present in Figure 3 representing the five attacked classes and the sign coverage. One can observe, that the MCSA method was able to outperform both RSA and SSA in all of the studied classes, while the RSA method barely deceived the DNNs under test. As shown in this figure, the class 30km/h is the most sensitive one among the five, while the stop sign is the most robust. After visual inspections , we concluded that the most observable reason behind this is the spatial diversity of the features of the two classes. For example, the digit 3 in the class 30km/h is the only feature that distinguishes it from the other speed limit classes such as 20km/h or 80km/h. Similar to that, the class children crossing also consists of features that lie in the center of the traffic sign, which can be covered with 2 to 3 stickers, leading to dropping the accuracy close to zero percent. Therefore, the attacks on all the networks are focused on covering those features.

On the other hand, the class stop sign consists of the word "STOP" along with an octagonal shape which is not similar to any of the other classes in the GTSRB data-set, which causes the MCSA attacks on different DNNs to not focus on any specific feature for this class. Finally, as shown in the last sub-figure of Figure 3, the sign coverage of the SSA and MCSA are very close, which indicates their similarity in selecting the stickers. One can also observe, that although having extremely centralized features in the classes 30km/h and children crossing, RSA did not lead to any significant drop in the performance. This indicates that up to five random stickers (or up to 20 percent coverage) applied by normal people, unless intentionally applied to cover the critical features, do not necessarily lead to any significant performance drop in the common DNNs.

Figure 4 represents the similar results per individual DNN averaged over the five classes. One can observe from this figure, that VGG19 and VGG16 performed the best against our realistic sticker applications. In contrary, ResNet18 appeared to be the most sensitive one. Furthermore, except for ResNet101, there is a linear correlation with the number of layers and robustness in the architectures with

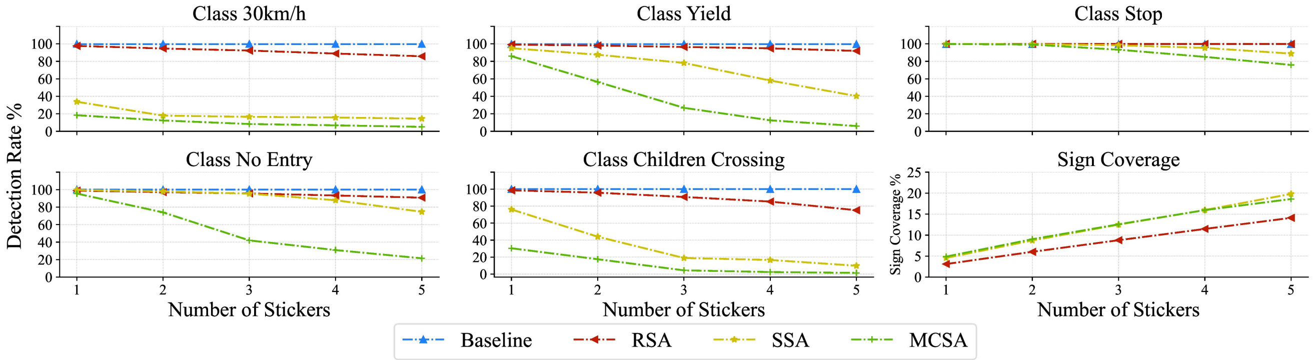

Fig. 3: Average accuracy of all the networks on different attacked classes and the average sign coverage of all the networks and the attacked classes according to number of the stickers. *Blue line*: the average clean accuracy of all the networks; *red line*: the average accuracy of all the networks after applying stickers using `RSA`; *yellow line*: the average accuracy of all the networks after applying stickers using `SSA`; *green line*: the average accuracy of all the networks after applying stickers using `MCSA`. *Top left*: the average accuracy of the networks on the class `30km/h`; *top center*: the average accuracy of the networks on the class `yield`; *top right*: the average accuracy of the networks on the class `stop`; *bottom left*: the average accuracy of the networks on the class `no entry`; *bottom center*: the average accuracy of the networks on the class `children crossing`; *bottom right*: the average sign coverage of different attack methods on all of the networks and the selected classes.

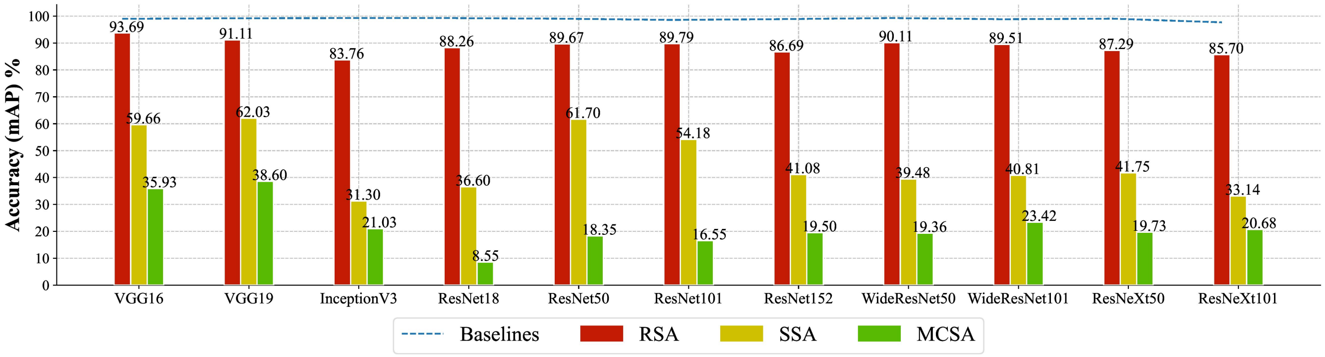

Fig. 4: Average accuracy of all the five classes per each model. *Blue dashed line*: the DNNs accuracy on the clean data; *red bars*: average accuracy of the `RSA` method; *yellow bars*: average accuracy of the `SSA` method; *green bars*: average accuracy of the `MCSA` method. The models are displayed on the x-axis which include: `VGG16` and `VGG19` [22], `InceptionV3` [23], `ResNet` DNNs with 18, 50, 101 and 152 layers [24], `WideResNet` with 50 and 101 layers [25], and `ResNeXt` with 50 and 101 layers [26]. The y-axis indicates the average accuracy based on mean average precision mAP.

size variations. Moreover, the difference in `SSA` and `MCSA` in some of the DNNs are much higher than the other ones. This can be an indication of how good one saliency extraction method, in our case Vanilla [21], can highlight the most important features in the input image, which could be used to compare different saliency methods.

Cross Analysis: The cross analysis of our `MCSA` results are presented in Table I. Each column of this table represents a `MCSA` optimised on the according model and tested on the other models. One can observe that the two models `VGG16` and `VGG19` appear to be the most robust ones among all the 11 models. Therefore, not only they have achieved the best accuracy results on the other 9 DNNs (the top two rows), but also they are the only DNNs that their sticker application achieved better misclassification rate on the other networks than themselves.

DNNs Common Interests: Based on the averaged results in the last row of Table I, one can conclude that except `InceptionV3`, `MCSA` could deceive all the other DNNs by more than 50 percent, while optimised on either of them, out of which, the `ResNet18` was the most extreme case with 30.26 percent. This can be an indication of how probable is to optimize one sticker application on one model and deceive other models with. We have also cross analysed the other attacks including our `SSA`, `RP2` [8] and `AP` [15], which achieved 53.68, 52.68 and 91.41 percent in the best case accordingly. This shows that with our `MCSA` approach, not only we were able to find the common important regions from the input classes that are crucial for one DNN to detect, but also we calculated the probability of such regions being important for other DNNs as well. Such a conclusion can be used in designing/refining traffic sign detectors to ensure a safe prediction (e.g. as the digit "3" is crucial for all the DNNs, one can conclude that there must be a visible 3 in the image, if the detector suggests `30km/h`).

Field Test Results: The field test results are presented in Table II. As shown in this table, except the class `stop` sign, all the other classes are successfully misdetected

Model		VGG [22]		InceptionV3 [23]	ResNet [24]				WideResNet [25]		ResNeXt [26]	
		16	19		18	50	101	152	50	101	50	101
VGG [22]	16	35.93	54.40	64.40	48.98	49.44	54.54	46.09	33.13	46.99	58.90	56.25
	19	54.68	38.60	62.10	39.08	41.21	54.42	43.92	47.59	34.37	57.45	56.40
InceptionV3 [23]		**30.94**	36.52	**21.03**	26.95	31.45	29.02	33.44	34.99	30.50	37.68	36.39
ResNet [24]	18	46.77	47.80	54.16	**8.55**	25.08	25.38	37.67	30.53	28.60	50.83	42.60
	50	40.47	46.94	60.53	21.75	**18.35**	50.82	39.29	33.49	32.58	52.83	47.77
	101	44.79	42.31	59.13	33.77	27.61	**16.55**	37.72	28.90	28.06	49.37	40.11
	152	49.99	50.01	64.40	36.32	30.58	21.79	**19.50**	27.52	34.53	50.93	46.15
WideResNet [25]	50	53.07	47.36	57.22	32.29	27.42	39.04	40.72	**19.36**	37.06	56.29	50.67
	101	37.72	**33.08**	50.59	31.77	27.73	32.17	31.72	27.38	**23.42**	41.64	28.29
ResNeXt [26]	50	40.97	39.03	47.44	28.35	30.33	31.33	27.24	32.03	26.66	**19.73**	36.76
	101	39.75	38.90	45.31	25.11	25.09	35.67	24.44	35.69	25.69	46.83	**20.68**
Average		43.19	43.18	53.30	**30.26**	30.39	35.52	34.70	31.87	31.68	47.50	42.01

TABLE I: The cross analysis results on our MCSA method averaged over all the five classes per individual models. Columns indicate the models, which MCSA is optimised on, and the rows indicate the models, which are evaluated on that optimised MCSA. The numbers indicate the mean average precision in percentage. Cells with gray background indicate the accuracy of that model on the MCSA optimised on the same model. Bold numbers represent the minimum accuracy per each column. The last row represents the averaged accuracy numbers of each column, out of which the bold number in this row indicates the minimum averaged accuracy in this row only.

Baseline & Attacks	30km/h	Yield	Stop	No Entry	Children Crossing
Clean Accuracy	97.85	100.00	100.00	99.64	100.00
RP2 [8]	29.22	67.12	100.00	71.95	1.13
AP [15]	100.00	99.37	100.00	100.00	99.37
MCSA (Ours)	**0.71**	**2.59**	100.00	**5.15**	**0.7**

TABLE II: The averaged field test results of RP2, AP and our MCSA optimized and applied to traffic signs in the real world. The first row indicates the clean accuracy of the aforementioned networks on the clean traffic sign video frames. The following rows indicate their accuracy on the same traffic sign after partially covered by the according method. In each column, lower accuracy indicates the success of the according method compared to the others in that column.

when covered by our MCSA method, which outperforms the adversarial methods from the literature. Furthermore, the AP [15] has not been successful in deceiving the DNNs as such, which explains the fact that introducing only salient features for deceiving DNNs while leaving the important features of the attacked object to be visible would not deceive the DNNs successfully. In fact, after inspecting the results of RP2 [8], we conclude that the partial success of this method in our field tests also relies on its spatial optimization, which covered the aforementioned important features of the attacked objects, and not the salient adversarial features it introduced. Finally, the sign coverage C for MCSA, RP2, and AP are 18.06, 19.9, and 23.96 percent accordingly, which indicates that MCSA covered less area from the traffic signs compared to the other two.

Stickers as Acts of Vandalism: Based on our experiments and the reported results on our RSA method, we conclude that the stickers applied to traffic signs by normal people, if not intentionally engineered to cover specific critical features, would not deceive the tested DNNs as such. However, we have not tested different printing materials such as fluorescent stickers that could cause highly salient textures and dark backgrounds when captured by camera.

MCSA for Data-set Enhancement: Although we did not use any of the sticker overlays in our DNNs training procedure, we suggest that such an experiment would lead to possible increase in the DNNs accuracy. In other words, one can use our methods to enhance the training data-sets to include traffic signs with different combination of stickers on them and use those along with the clean data to train the DNNs. The aim of such an experiment would be to increase the robustness of the traffic sign detectors against the realistic stickers.

V. Conclusions

In this paper, we have reviewed the challenges of applying realistic looking stickers to traffic signs, both as unintentional acts of vandalism and also as intentional act aiming at deceiving particular DNNs. Based on that, we have proposed three sticker content enhancement methods to apply as pre-processing steps towards closing the gap between the traffic signs and the stickers in terms of their difference in brightness, contrast and blurriness. Furthermore, we have also proposed three sticker application methods, namely RSA, SSA and MCSA that can deceive the traffic signs recognition DNNs with realistic looking stickers, which by outperforming the adversarial methods from literature, led us to this final conclusion that deploying highly salient adversarial attacks to the real world would be less crucial than covering the important features by even normal stickers for the detectors performance.

References

[1] S. Houben, J. Stallkamp, J. Salmen, M. Schlipsing, and C. Igel, “Detection of traffic signs in real-world images: The German Traffic Sign Detection Benchmark,” in *International Joint Conference on Neural Networks*, no. 1288, 2013.

[2] D. R. Bruno, D. O. Sales, J. Amaro, and F. S. Osório, “Analysis and fusion of 2d and 3d images applied for detection and recognition of traffic signs using a new method of features extraction in conjunction with deep learning,” in *2018 International Joint Conference on Neural Networks (IJCNN)*, 2018, pp. 1–8.

[3] J. Fulco., A. Devkar., A. Krishnan., G. Slavin., and C. Morato., “Empirical evaluation of convolutional neural networks prediction time in classifying german traffic signs,” in *Proceedings of the 3rd International Conference on Vehicle Technology and Intelligent Transport Systems - VEHITS,*, INSTICC. SciTePress, 2017, pp. 260–267.

[4] J. Guo, J. Lu, Y. Qu, and C. Li, “Traffic-sign spotting in the wild via deep features,” in *2018 IEEE Intelligent Vehicles Symposium (IV)*, 2018, pp. 120–125.

[5] J. Lu, Y. Qu, and X. Yang, "Traffic sign recognition with inception convolutional neural networks," in *Internet Multimedia Computing and Service*, B. Huet, L. Nie, and R. Hong, Eds. Singapore: Springer Singapore, 2018, pp. 487–494.

[6] B. Sanyal, R. K. Mohapatra, and R. Dash, "Traffic sign recognition: A survey," in *2020 International Conference on Artificial Intelligence and Signal Processing (AISP)*, 2020, pp. 1–6.

[7] M. Shahud, J. Bajracharya, P. Praneetpolgrang, and S. Petcharee, "Thai traffic sign detection and recognition using convolutional neural networks," in *2018 22nd International Computer Science and Engineering Conference (ICSEC)*, 2018, pp. 1–5.

[8] K. Eykholt, I. Evtimov, E. Fernandes, T. Kohno, B. Li, A. Prakash, A. Rahmati, and D. Song, "Robust physical-world attacks on machine learning models," *CoRR*, vol. abs/1707.08945, 2017. [Online]. Available: http://arxiv.org/abs/1707.08945

[9] R. Feng, J. Chen, E. Fernandes, S. Jha, and A. Prakash, "Robust physical hard-label attacks on deep learning visual classification," 2021.

[10] C. Sitawarin, A. N. Bhagoji, A. Mosenia, M. Chiang, and P. Mittal, "DARTS: deceiving autonomous cars with toxic signs," *CoRR*, vol. abs/1802.06430, 2018. [Online]. Available: http://arxiv.org/abs/1802.06430

[11] M. Soll, "Informaticup competition 2019: Fooling traffic sign recognition," in *KI 2019: Advances in Artificial Intelligence*, C. Benzmüller and H. Stuckenschmidt, Eds. Cham: Springer International Publishing, 2019, pp. 325–332.

[12] P. Bielik, P. Tsankov, A. Krause, and M. Vechev, *Reliability Assessment of Traffic Sign Classifiers*. Bonn, Germany: Federal Office for Information Security, 2020. [Online]. Available: https://www.bsi.bund.de

[13] A. F. Magnussen, N. Le, L. Hu, and W. E. Wong, "A survey of the inadequacies in traffic sign recognition systems for autonomous vehicles," *International Journal of Performability Engineering*, vol. 16, no. 10, p. 1588, 2020. [Online]. Available: http://www.ijpe-online.com/EN/10.23940/ijpe.20.10.p10.15881597

[14] S. B. Wali, M. A. Abdullah, M. A. Hannan, A. Hussain, S. A. Samad, P. J. Ker, and M. B. Mansor, "Vision-based traffic sign detection and recognition systems: Current trends and challenges," *Sensors*, vol. 19, no. 9, 2019. [Online]. Available: https://www.mdpi.com/1424-8220/19/9/2093

[15] T. B. Brown, D. Mané, A. Roy, M. Abadi, and J. Gilmer, "Adversarial patch," 2018.

[16] H. Lengyel and Z. Szalay, *Traffic sign anomalies and their effects to the highly automated and autonomous vehicles*, 08 2018, pp. 193–204.

[17] H. Lengyel and z. Szalay, "Test scenario for road sign recognition systems with special attention on traffic sign anomalies," in *2019 IEEE 19th International Symposium on Computational Intelligence and Informatics and 7th IEEE International Conference on Recent Achievements in Mechatronics, Automation, Computer Sciences and Robotics (CINTI-MACRo)*, 2019, pp. 000 193–000 198.

[18] D. Temel, T. Alshawi, M. Chen, and G. AlRegib, "Challenging environments for traffic sign detection: Reliability assessment under inclement conditions," *CoRR*, vol. abs/1902.06857, 2019. [Online]. Available: http://arxiv.org/abs/1902.06857

[19] D. Temel, M.-H. Chen, and G. Alregib, "Traffic sign detection under challenging conditions: A deeper look into performance variations and spectral characteristics," *IEEE Transactions on Intelligent Transportation Systems*, vol. PP, pp. 1–11, 08 2019.

[20] X. Wang, "Laplacian operator-based edge detectors," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 5, pp. 886–890, 2007.

[21] K. Simonyan, A. Vedaldi, and A. Zisserman, "Deep inside convolutional networks: Visualising image classification models and saliency maps," in *Workshop at International Conference on Learning Representations*, 2014.

[22] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, Y. Bengio and Y. LeCun, Eds., 2015. [Online]. Available: http://arxiv.org/abs/1409.1556

[23] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[24] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778.

[25] S. Zagoruyko and N. Komodakis, "Wide residual networks," in *Proceedings of the British Machine Vision Conference (BMVC)*, E. R. H. Richard C. Wilson and W. A. P. Smith, Eds. BMVA Press, September 2016, pp. 87.1–87.12. [Online]. Available: https://dx.doi.org/10.5244/C.30.87

[26] S. Xie, R. Girshick, P. Dollar, Z. Tu, and K. He, "Aggregated residual transformations for deep neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[27] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," in *Proc. of ICLR*, San Diego, CA, USA, May 2015, pp. 1–15.