Towards a More Robust and Accurate OCR Model with Adversarial Techniques in HMI Testing Scenarios

Yupeng, Cheng
Nanyang Technological University
Singapore
yupeng.cheng@ntu.edu.sg

Zi Pong, Lim
Continental Corporation
Singapore
zi.pong.lim@continental-corporation.com

Sarthak Ketanbhai, Modi
Nanyang Technological University
Singapore
sarthak005@e.ntu.edu.sg

Yon Shin, Teo
Continental Corporation
Singapore
yon.shin.teo@continental-corporation.com

Yushi, Cao
Nanyang Technological University
Singapore
yushi002@e.ntu.edu.sg

Shang-Wei, Lin
Nanyang Technological University
Singapore
shang-wei.lin@ntu.edu.sg

***Abstract*—Test automation has become increasingly important as the complexity of both design and content in Human Machine Interface (HMI) software continues to grow. Current standard practice uses Optical Character Recognition (OCR) techniques to automatically extract textual information from HMI screens for validation. At present, one of the key challenges faced during the automation of HMI screen validation is the noise handling for the OCR models. In this paper, we propose to utilize adversarial training techniques to enhance OCR models in HMI testing scenarios. More specifically, we design a new adversarial attack objective for OCR models to discover the decision boundaries in the context of HMI testing. We then adopt adversarial training to optimize the decision boundaries towards a more robust and accurate OCR model. In addition, we also built an HMI screen dataset based on real-world requirements and applied multiple types of perturbation onto the clean HMI dataset to provide a more complete coverage for the potential scenarios. We conduct experiments to demonstrate how using adversarial training techniques yields more robust OCR models against various kinds of noises, while still maintaining high OCR model accuracy. Further experiments even demonstrate that the adversarial training models exhibit a certain degree of robustness against perturbations from other patterns.**

***Index Terms*—OCR model, adversarial, HMI testing**

I. Introduction

In automotive terminology, Human-Machine Interface (HMI) refers to the technology and systems that allow interaction and communication between drivers and various electronic systems within a vehicle. The goal of a well-designed HMI is to provide a user-friendly and intuitive interface that enables users to control and interact with different functions of the vehicle, such as infotainment systems, climate control, navigation, safety features, and more [18]. Therefore, a well-designed HMI becomes more important as electronic systems are deployed in vehicles all over the world, among which car dashboards as a particular example. Ensuring the meticulous design of HMI software within the automotive sector is critical, given its direct impact on both user experience and safety outcomes. Conducting comprehensive tests to verify the design and functionalities of HMI software is thus of paramount importance.

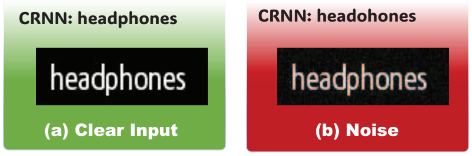


Fig. 1: Influence of random noise against OCR model. (a) clear input of image patch *headphones*. (b) input with a random noise perturbation (maximum perturbation 0.1). The prediction result of a widely used OCR recognition model CRNN [21] is illustrated on the top of each image patch.

HMI testing is the systematic process performed on the HMI products before they are delivered to customers, to ensure optimal user experience and customer satisfaction. As HMI products continue to grow in complexity nowadays and incorporate a greater array of functionalities, the significance of test automation has risen remarkably in addressing the challenges presented by the labor-intensive nature of manual testing procedures [24].

In the software testing workflow, the validation of the HMI screens is a critical component to make sure the design and the layout align with the customers' requirements. The icons and textual information are checked during this process. Current HMI testing automation strategies use Optical Character Recognition (OCR) to extract the letters and icons that appear on the HMI screens. The extracted information will then be verified with the requirement documents provided by the customers[7]. However, as the HMI screens are embedded

979-8-3503-5409-6/24/$31.00 ©2024 IEEE
DOI 10.1109/CAI59869.2024.00252

into the display clusters (car dashboards), testers typically deploy additional hardware such as mounted cameras or frame-grabbing devices to capture the HMI screens and send them back to the computing devices for validation. The signals can be easily affected during this process, resulting in unexpected noises or perturbations in the captured images. Therefore, the accuracy of the OCR model is greatly affected, making the validation process less accurate and requiring more human effort for manual checking.

Fig. 1 shows how noise can influence the text recognition capabilities of an OCR model. The clear input yields a correct recognition result **headphones**. However, when the image patch is subjected to imperceptible random noise, the prediction is misled to **headohones**. The noises are of a low scale and are unable to be identified easily by the naked eye. When this happens during the HMI testing process, human testers need to abort testing to manually identify the root cause for this issue, thus slowing down the overall test process.

A traditional method used in the industry to solve this issue is by augmenting the training data with added noise perturbations (*e.g.*, Gaussian noise) before the training phase of the model-based OCR. However in this case, although the robustness against the added noise (*e.g.*, Gaussian noise) has been improved, the model still lacks resistance to other types of noises [19]. It is also impractical to enumerate all types of noises for image augmentation during the data preparation stage. Therefore, a recognition model with good robustness (against various kinds of noises) while retaining high accuracy is required.

In order to overcome this, we propose utilizing adversarial training techniques on model-based OCR to efficiently improve its robustness. We first build an HMI dataset using the asset files of an actual automotive HMI software, infused with various kinds of noises that occur during the testing process. Based on the HMI dataset, we apply white-box adversarial attacks (a common adversarial attack technique that has unrestricted access to the model and its execution to better discover the limits of the models [2]) on the images to discover the decision boundaries of the OCR model. We then train the OCR model to improve recognition results with adversarial examples (examples that lie around the decision boundaries). To elaborate further, our method utilizes an adversarial attack to discover the weakness of the OCR model and then adopts adversarial training to focus on the weaknesses. Thus the robustness against various kinds of noises can be improved while maintaining high accuracy.

To summarize, our contributions are:

- We proposed using adversarial techniques on OCR models to improve its capabilities to resist noises. Adversarial attack techniques are adopted to discover the adversarial examples of the OCR model and adversarial training techniques are utilized to specifically train the OCR models on the adversarial examples to improve robustness.
- Following real-world industry requirements, we obtained typical HMI screen designs and constructed a dataset that contains HMI textual information. The dataset consists of 9883 HMI image patches. We also apply multiple kinds of perturbation onto the clean HMI dataset to provide a more complete coverage for the potential scenarios.
- Experimental results conducted on the HMI dataset reveal that adversarial training substantially improves the resilience of OCR models to diverse noise types. Further experiments even demonstrate that the adversarial training model exhibits a certain degree of robustness against perturbations from other patterns.

II. Related Works

Optical Character Recognition (OCR) is a technology that translates text from images into machine-readable text. It was first mentioned in 1982 [13], where reading machines were developed as devices to help the blind read.

More recent implementations of OCR include using deep learning techniques such as Multi-Layer Perceptrons (MLP) [9] , Convolutional Neural Networks (CNN)[20]; kernel-based methods such as Support Vector Machines (SVM) [5]; statistical methods such as K Nearest Neighbour (KNN) [16].

A recent survey [14], which has shown an overview of OCR techniques and various phases such as acquisition, pre-processing, segmentation, feature extraction, classification, and post-processing, also mentioned that the employment of OCR systems in practical applications still remains an active area of research.

Adversarial techniques were first introduced by Szegedy *et al.* [23]. The authors created adversarial states to manipulate the network's policy. They showed that even slight state perturbations can potentially lead to very significant differences in terms of performance and decisions. Following that, Goodfellow *et al.* proposed an efficient one-step method for generating adversarial examples, known as the Fast Gradient Sign Method (FGSM) [12]. Kukarin *et al.* [15] demonstrated by using the Iterative Fast Gradient Sign Method (I-FGSM), input-specific adversarial examples can be deployed in the physical world in an untargeted attack if printed out and carefully cropped. Dong *et al.* [10] promoted the Momentum Iterative Fast Gradient Sign Method (MI-FGSM), with the use of momentum in order to enhance the process of creating adversarial instances while using iterative algorithms, thus introducing a broad class of momentum-based iterative algorithms to boost adversarial attacks. Diverse Input I-FGSM (DI2FGSM) [25] is another example of the attacks that directly build on FGSM. The main idea of DI2FGSM is to diversify the input used in each iteration of the iterative FGSM by applying image transformations, such as random resizing and padding, with a fixed probability. This diversification is claimed to facilitate better transferability of the resulting attack in a black-box setup.

As for applications of adversarial attack techniques on OCR models, Song *et al.* [22] have demonstrated that even state-of-the-art deep learning-based OCR models are vulnerable to adversarial images. Chen *et al.* [8] proposed a watermark attack method to produce natural distortion that can yield a

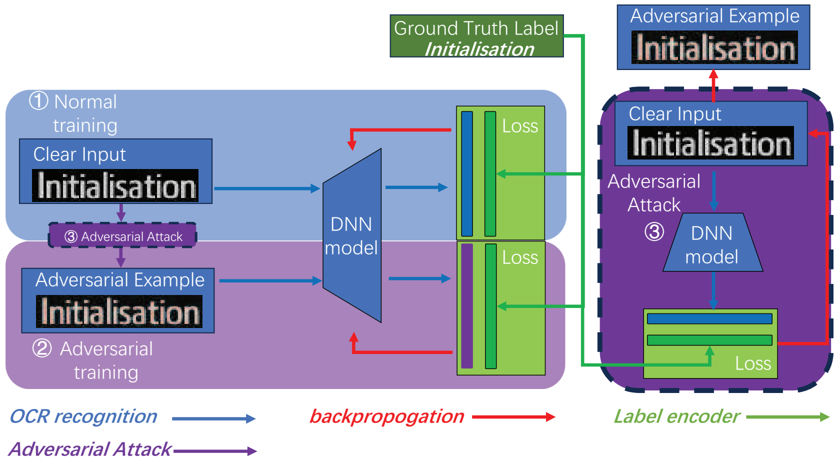


Fig. 2: Pipeline of Adversarial Training and Adversarial Attack. ❶ Normal training process. ❷ Adversarial training process. ❸ Adversarial attack.

set of natural adversarial examples and attain similar attack performance to the state-of-the-art methods in different attack scenarios.

III. Methodology

Our motivation lies in achieving enhanced robustness within the training process while minimizing perturbation levels to a degree imperceptible by testers. In real-life scenarios, random noise typically occurs during graphics rendering, data transmission or as a result of electromagnetic interference. Augmenting the training samples using random noise can also improve the robustness of OCR models. However, the impact of adversarial attacks is not only more efficient but also more effective [3]. To be more precise, when applying adversarial training, the examples that lie around the decision boundaries (hard/adversarial examples) will be selected (by adversarial attack) and augmented for adversarial training. The adversarial strategy ensures that difficult samples are consistently chosen during each augmented training epoch, leading to the model's robustness being improved. On the contrary, when augmenting with random noise, the samples are randomly selected and augmented, leading to a high probability that these samples remain inside the decision boundaries (easy samples), thus making the training less efficient. Fig. 3 illustrates the effects of random noise as well as adversarial attacks on the CRNN model. It is easy to see that, adversarial attacks can achieve a higher attack success with less perturbation, *i.e.*, noise scale. Thus, adversarial attacks are better suited to reveal the vulnerabilities of the model.

To achieve this, we need to discover the decision boundaries of the OCR model and then conduct adversarial training to make the OCR model more robust. The whole pipeline of our method is shown in Fig. 2. ❶ in Fig. 2 represents the normal training process where the clear input is directly taken from the dataset without any augmentations. ❷ in Fig. 2 represents the adversarial training process where the adversarial examples are augmented by the adversarial attack (❸ in Fig. 2). The adversarial attack module is utilized to obtain the adversarial examples. More specifically, it utilizes gradient to discover decision boundaries and generate adversarial noises (added to the input images, referred to as adversarial examples) that mislead the OCR model. After obtaining the adversarial examples, adversarial training is conducted based on these samples to improve the robustness of the OCR models.

A. Adversarial Attack for HMI OCR

Given an image patch, *e.g.*, $\hat{\mathbf{X}}$, we can view it as a combination of a clear input $\mathbf{X}$ and a noise perturbation $\mathbf{N}$ that originated from the data transfer or image acquisition processes:

$$\hat{\mathbf{X}} = \mathbf{X} + \mathbf{N}, \tag{1}$$

When an OCR model is employed to recognize the characters embedded in $\hat{\mathbf{X}}$, a novel objective emerges. This objective aims to mislead the OCR model through the implementation of a well-crafted perturbation map $\mathbf{N}$ and thus discover the decision boundaries of the OCR model.

The target of the adversarial attack is to find $\mathbf{N}$, which maximizes the loss in attack objective function [11], [10]:

$$\arg\max_{\mathbf{N}}(\ell(f_\theta(\mathbf{X}+\mathbf{N}),\mathbf{G})), \text{ subject to } \quad \|\mathbf{N}\|_\infty \le \epsilon, \tag{2}$$

where $f_\theta(\cdot)$ represents the OCR model, $\mathbf{G}$ represents the ground truth label, and $\ell(\cdot)$ is the loss function for character recognition, *e.g.*, the CTC loss. ϵ refers to the maximum perturbation.

Generally, we solve Eq. (2) by sign gradient descent. Note the number of attack iteration as T, and each pixel value in $\mathbf{N}$ is updated for every iteration t:

$$\mathbf{N}_t = \mathbf{N}_{t-1} + \alpha \ \mathrm{sign}(\nabla \ell_{\mathbf{N}_{t-1}}), \tag{3}$$

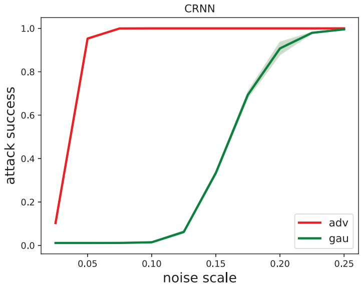


Fig. 3: Evaluating the Impact of Gaussian Noise (gau) and Adversarial Attacks (adv) on the CRNN OCR Recognition Model. "Noise scale" is the maximum perturbation ϵ of each pixel. Higher "attack success" refers to a larger influence on the model.

where α is the step size of each iteration. $\mathrm{sign}(\cdot)$ represents the signum function. $\nabla\ell_{\mathbf{N}_{t-1}}$ denotes the gradient of $\mathbf{N}_{t-1}$ with respect to the objective function. By solving Eq. (2), the adversarial examples ($\hat{\mathbf{X}}$) with the highest potential for crossing decision boundaries can be identified. These adversarial examples are used for adversarial training to improve the robustness of the OCR model. [1]

Fig. 2 ❸ illustrates the comprehensive attack process pipeline. It starts with the *Clear Input* $\mathbf{X}$, which is initially fed into the DNN model to generate prediction logits (blue bar). Subsequently, the Loss function (green block) is computed by integrating the logits with the encoded ground truth label $\mathbf{G}$ (green bar). Ultimately, by employing the backpropagation technique (red lines), we acquire the gradient of the input layer, *i.e.*, $\nabla\ell_{\mathbf{N}_{t-1}}$. After T iterations of updates, the ultimate adversarial example $\hat{\mathbf{X}}$ is derived. Note that the adversarial attack process bears a resemblance to training. However, in contrast to updating the model parameters, the attack employs gradients to generate adversarial noise in the input layer.

B. Adversarial Training

We now introduce adversarial training in this section. It utilizes the capability of adversarial examples to expose the weaknesses of the model, thus comprehensively enhancing the model's robustness. This approach goes beyond the limitations imposed by the characteristics of typical noise distributions.

Specifically, given a clear input $\mathbf{X}$ and its corresponding label $\mathbf{G}$, a universal objective function of a normal training process can be represented as:

$$\min_{\theta} \mathbb{E}_{(\mathbf{X},\mathbf{G})\sim\mathcal{D}}[\ell(f_\theta(\mathbf{X}),\mathbf{G})], \tag{4}$$

where $\mathcal{D}$ is an underlying data distribution. As shown in Fig. 2, the normal training process (blue) updates the parameters θ in the OCR model by minimizing the loss.

Adversarial training aims to minimize the loss of the most difficult examples. By incorporating Eq. (2) into Eq. (4), the objective function Eq. (4) is rewritten as:

$$\min_{\theta} \mathbb{E}_{(\mathbf{X},\mathbf{G})\sim\mathcal{D}}[\max_{\mathbf{N}\in\Omega} \ell(f_\theta(\mathbf{X}+\mathbf{N}),\mathbf{G})], \tag{5}$$

where Ω represents the perturbation space. By incorporating adversarial examples during the training phase, adversarial training has been empirically established as one of the most effective methods for enhancing the robustness of a vulnerable model [17].

Once we have obtained the adversarial noise $\mathbf{N}_{i-1}$ for a clean sample $\mathbf{X}$, we proceed to update the parameters of the target model θ using gradient descent in i-th step:

$$\theta_i = \theta_{i-1} + \eta\ \nabla_\theta \ell(f_{\theta_{i-1}}(\mathbf{X}+\mathbf{N}_{i-1}),\mathbf{G}), \tag{6}$$

where η is the learning rate.

As shown in Fig. 2 ❷ **Adversarial training** (purple block), during the training process, we can conduct an adversarial attack module (Fig. 2 ❸) and utilize the adversarial example for the parameters updating.

In Section IV, we will perform experiments to showcase the effectiveness of adversarial training in enhancing the robustness of the vulnerable model when compared with models trained under normal conditions and data augmentation.

IV. Experiments

In this section, we provide an in-depth analysis of our experimental outcomes aimed at showcasing the potency of OCR adversarial attacks. We begin by presenting the construction process of the HMI dataset in Section IV-A. Subsequently, we elucidate the outcomes of our adversarial attack evaluations in Section IV-B. Finally, we showcase the effectiveness of adversarial training in Section IV-C. Section IV-D illustrates and analyses some visualization results.

a) Dataset: In order to comprehensively assess the efficiency of OCR models on HMI image patches, we constructed a specialized HMI dataset utilizing a designated font asset. It is important to note that we intentionally refrain from utilizing publicly available generic OCR recognition dataset for our evaluation tests. The rationale behind this decision will be elaborated upon in the subsequent Section IV-A.

b) Model: In order to validate the efficacy of OCR adversarial training in mitigating the effects of perturbations, we opted for a robust OCR recognition framework, which is the Convolutional Recurrent Neural Network (CRNN) [21], incorporating a straightforward *ResNet34* backbone architecture. More precisely, we employ three distinct training methodologies for the ResNet34-based CRNN model: normal training (**NorTrain**), adversarial training (**AdvTrain**), and training augmented with Gaussian noise (**GauTrain**). Additionally, we trained a model with Gaussian blur augmentation (**GauBlurTrain**), which serves as a baseline for evaluating the robustness of **AdvTrain** against perturbations from other patterns.

[1] In the experiment, we choose $\alpha = \epsilon/T$.

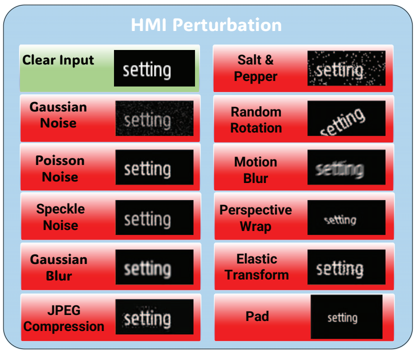


Fig. 4: Examples of our HMI dataset and the corresponding perturbations of word ‘setting’.

c) Perturbation: As we want to evaluate various noise types in the testing scenario, we choose random Gaussian noise [6] and Salt&Pepper noise [6] in our experiments to measure the robustness of models as they have totally different distributions. In the case of Gaussian noise, each noise pixel is generated based on a Gaussian distribution. The “noise scale” in this noise refers to the maximum perturbation. On the other hand, in Salt&Pepper noise, certain pixel values within the image are replaced with corrupted values, which can either be the maximum value 255 or the minimum value 0. The “noise scale” in this noise indicates the ratio of corrupted pixels. Note that adversarial noise is also regarded as a perturbation in the following experiments and the “noise scale” in this noise means the maximum perturbation. Furthermore, for a more comprehensive assessment of the impact of various perturbations, we conduct experiments incorporating Gaussian Blur and Motion Blur to evaluate their effects on the Additive Adversarial Training model, i.e., **AdvTrain**. The term “noise scale” in Gaussian blur and Motion blur refers to the standard deviation and kernel size of the blur kernel, respectively. The kernel size of Gaussian blur is fixed at 5.

d) Metric: In assessing the effectiveness of the adversarial attack, we employ the attack success rate and noise scale as metrics to evaluate the performance of a perturbation technique. The formulation of calculating the attack success rate is:

$$Succ.Rate = \frac{Acc_{clr} - Acc_{pert}}{Acc_{clr}}, \tag{7}$$

where Acc_{clr} indicates the recognition accuracy of the OCR model on the clear inputs, and Acc_{pert} refers to the accuracy of the OCR model on the perturbated inputs. Note that a lower attack success rate indicates a better robustness of a model.

e) Implementation Detail: In the HMI testing scenarios, since we focus on the impact of cases that remain imperceptible to testers yet significantly affect DNN models, we set the maximum perturbation ϵ to values ranging from 0.02 to 0.25 for the Gaussian and adversarial perturbation. We also choose the values for the noise scale to be from 0.025 to 0.2 for Salt&Pepper noise. The attack iteration T is 10. The standard deviation of Gaussian blur ranges from 0.5 to 1.0, and the kernel size of motion blur ranges from 1 to 7.

A. HMI dataset

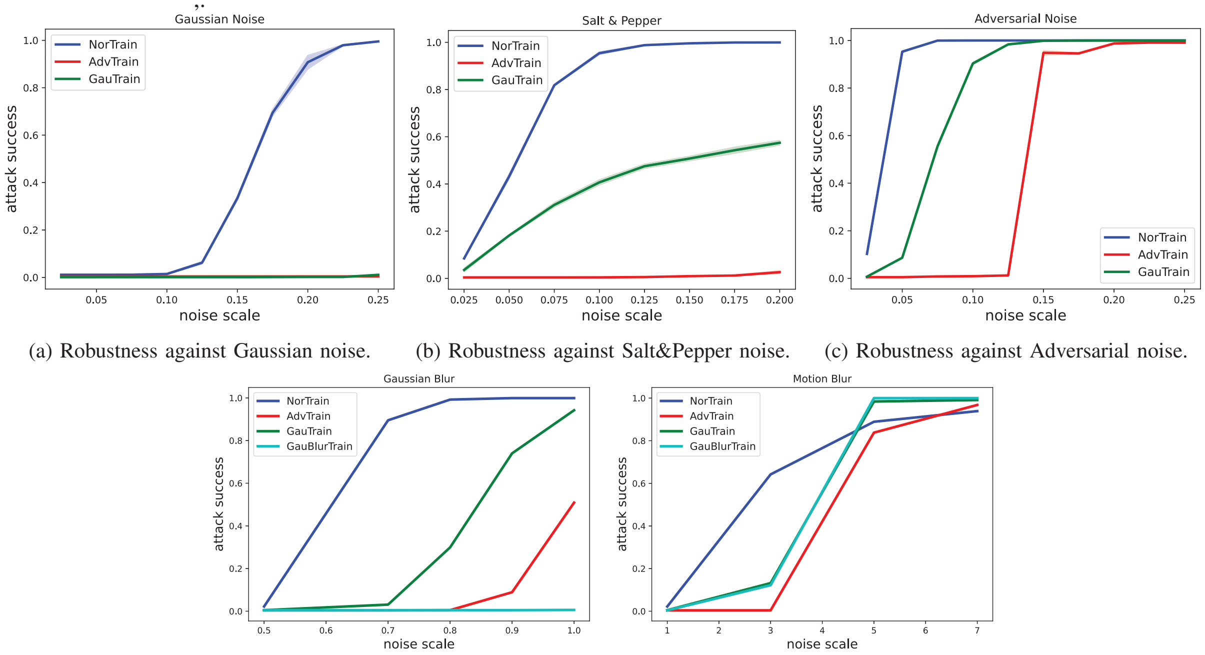
The OCR model trained in a conventional environment does not consider variations commonly encountered in the HMI testing scenario, such as changes in lighting conditions and text distortions. Consequently, this leads to its inability to achieve the highest recognition accuracy for the simple HMI image patches using the smallest network structure.

Furthermore, the text recognition disparities between general OCR and HMI testing scenarios are multi-fold. In typical OCR recognition problems, the primary goal is to accurately identify the text content. However, in the context of HMI testing scenarios, the presence of variations such as rendering errors, random noise and distortions are forbidden. In such situations, the model’s requirement is not to be robust against these variations but rather to be aware of the discrepancies that are not permissible. It then serves to alert testing personnel to potential issues, such as problems with the display color module or image transmission module. Conversely, the model’s true necessity for robustness lies in a specific domain, *e.g.*, imperceptible perturbations, which go unnoticed by testing personnel. Thus, the exploration of HMI OCR model characteristics necessitates limited robustness.

Guided by the principle of Occam’s razor [4], we employ a compact network, *e.g.*, ResNet34, and a proprietary HMI dataset to assess the impact of adversarial attacks and general perturbations. Specifically, we generate over 9883 image patches that follow the HMI testing scenario. The words are picked from a publicly available dataset [1]. Afterwards, we apply multiple kinds of perturbation onto the clean HMI dataset to provide a more complete coverage for the potential scenarios. Fig. 4 illustrates several perturbation examples of the word “setting”, including Gaussian noise, Poisson noise, Speckle Noise, Gaussian Blur, JPEG Compression, Salt and Pepper, Random Rotation, Motion Blur, Perspective Wrap, Elastic Transform, and Padding. Moreover, the different perturbation scales are also applied in our dataset.

B. Evaluation of adversarial attack

First we showcase the potency of our attack by assessing the performance of adversarial examples generated for a typically well-trained OCR model using our HMI dataset. It is important to emphasize that a stronger perturbation inherently yields a greater impact on the OCR model’s predictions. Therefore, for a meaningful comparison, we evaluate the attack success rate while maintaining consistent levels of perturbation. To achieve this, we fine-tune the noise scale for each type of perturbation, resulting in attack success rate - noise scale curves that facilitate clear and visually informative comparisons. Note that Salt&Pepper noise is not included in this experiment due to its inability to align with the perturbation evaluation criteria used for the previous two types of noise. Specifically, we slide the ϵ in Eq. (2) from 0.02 to 0.25 to tune the success rate. Fig. 3 shows the comparison results on our HMI dataset.



(a) Robustness against Gaussian noise. (b) Robustness against Salt&Pepper noise. (c) Robustness against Adversarial noise.

(d) Robustness against Gaussian blur. (e) Robustness against Motion blur.

Fig. 5: (a-c) Evaluation of the influence of Gaussian, Salt & Pepper, and adversarial noises against three trained models, **NorTrain**, **AdvTrain**, and **GauTrain**. (d-e) The effect of blur perturbations, *e.g.*, Gaussian Blur and Motion Blur, against Gaussian Blur augmented training model **GauBlurTrain** and additive adversarial training model **AdvTrain**.

In general, higher perturbation leads to lower accuracy for the OCR model. The model can resist perturbation with a low noise scale. For example, the attack success rates of adversarial attack and Gaussian noise are around 0% with the noise scale lower than 0.02. However, the attack success rate rises rapidly with the increase of noise scale. Moreover, the adversarial attack shows high effectiveness in disrupting the recognition accuracy of the OCR model. Concretely, in the blue curve of adversarial attack, there is a rapid rise of attack success rate from 1% to 97% when the noise scale increases from 0.02 to 0.05. However, the attack success rate of Gaussian noise remains around 0% within a noise scale of 0.05. It only achieves 87% with a noise scale of 0.2. The findings suggest that adversarial attacks can conduct a more substantial influence on OCR models through perturbations that are even less perceptible. Visualization results will be illustrated to demonstrate this conclusion in Section IV-D.

C. Evaluation of adversarial training

We can integrate the attack within the training process to enhance the robustness of DNNs against additive perturbations. This involves generating adversarial examples during each epoch and subsequently updating the DNN parameters using these augmented samples. To demonstrate the advantages of adversarial attacks in enhancing DNN robustness, we implement three distinct training strategies as introduced in Section III-B. Specifically, we denote the ordinarily trained model with Eq. (4) as **NorTrain**, the model trained with Gaussian noise augmentation as **GauTrain**, the model trained with Gaussian blur augmentation as **GauBlurTrain**, and the adversarial training model **AdvTrain**.

We first train **AdvTrain** and **GauTrain** according to the previous descriptions. Together with **NorTrain**, we evaluate the robustness of the three models against random Gaussian noise, difficult/Adversarial noise, and Salt&Pepper noise. This evaluation involves adjusting the maximum perturbation ϵ from 0.02 to 0.25 for the former two types of noise and the noise scale from 0.02 to 0.2 for Salt&Pepper noise. Fig. 5 depicts the evaluation outcomes of the three models, yielding the following observations: Firstly, the **NorTrain** model lacks robustness against all types of noise. Its attack success rates nearly reached 100% in the Gaussian noise with a noise scale of 0.225, in Salt&Pepper noise with a noise scale of 0.125, and in Adversarial noise with a noise scale of 0.075. Secondly, the introduction of Gaussian noise augmentation during training notably enhances the model's resilience against Gaussian noise. As shown in Fig. 5 (a), the attack success rate of Gaussian noise against **GauTrain** remains steady at 1% even with a heavy noise scale of 0.25. Nonetheless, Fig. 5 (b)&(c) indicate that the **GauTrain** model remains susceptible to Salt&Pepper noise and adversarial noise, albeit exhibiting slightly improved robustness compared to the **NorTrain** model. Its attack success rate nearly reached 50% in Salt&Pepper noise with a noise scale of 0.125, and in Adversarial noise with a noise scale of 0.075. Thirdly, adversarial training effectively mitigates the influence of all kinds of noise investigated here. As depicted in Fig. 5 (a), the **AdvTrain** model demonstrates comparable robustness against Gaussian noise to **GauTrain**. Specifically, the attack success rate of Gaussian noise against **AdvTrain** remains under 2% even with a noise scale of 0.25. As for the robustness against Salt&Pepper noise, **AdvTrain** maintains an attack success rate within 2% with a noise scale of 0.2. In terms of adversarial noise perturbations, **AdvTrain** displays resistance in comparison to both **NorTrain** and **GauTrain**. Specifically, the attack success rate of adversarial attacks remains at 0% with a noise scale of 0.1, while it exceeds 90% in the other two models at the same perturbation level.

To further evaluate the robustness of **AdvTrain** against other perturbation patterns, we also test its recognition performance on Gaussian blur and Motion blur. As shown in Fig. 5 (d) and (e), **GauBlurTrain** exhibits the lowest attack success rate when confronted with Gaussian blur, demonstrating the effectiveness of training augmented with Gaussian blur. **AdvTrain** also achieves a comparable robustness with **GauBlurTrain**. It maintains a 0% attack success rate with the Gaussian blur with a standard deviation of 0.8. However, when confronted with motion blur, the **GauBlurTrain** model exhibits low robustness against this perturbation, even weaker than the additive adversarial training model **AdvTrain**. Specifically, as shown in (e), when facing motion blur with a kernel size of 3, **AdvTrain** still suppress the attack success rate to 0%, whereas this perturbation attains an approximate 15% attack success rate against the **GauBlurTrain** model.

D. Qualitative evaluation

In this section, we first visually demonstrate the robustness of the three models against Gaussian noise, Salt&Pepper noise, and adversarial noise using an example word ‘delayed’. Gaussian noise is uniformly selected with a maximum perturbation of $\epsilon = 0.125$. On the other hand, adversarial noise is crafted from three models, *i.e.*, **NorTrain**, **GauTrain**, and **AdvTrain**. It also employs a maximum perturbation $\epsilon = 0.125$. Lastly, the noise scale of Salt&Pepper noise is 0.1. In addition, we present the recognition results of motion-blurred images using **AdvTrain** and **GauBlurTrain** to assess their efficacy in handling motion blur. The final results are presented in Fig. 6. In summary, we make the following observations:

❶ All models correctly recognize the clear input. ❷ In terms of additive noise, as shown in (a), Gaussian noise starts to impact the detection results of the **NorTrain** model. For example, ‘delayed’ is falsely recognized as ‘delayedd’. However, the deviations from the ground truth labels are not substantial. On the other hand, **GauTrain** and **AdvTrain**, due to their consideration of additive noise during the training phase, exhibit robustness to such variations. ❸ Salt&Pepper noise exhibits a distinctly different characteristic distribution from random Gaussian noise. This noise type directly sets some pixel values to 0 or 255, making it more pronounced and significantly impactful on the models. Notably, both **NorTrain** and **GauTrain** models experience instances of misclassification. For instance, the word ‘delayed’ is erroneously identified as ‘delayedj’. ❹ Adversarial noise significantly impacts **NorTrain**, resulting in more severe recognition deviations. For instance, ‘delayed’ is recognized as ‘uabaxadllll’. Although **GauTrain** also fails to accurately recognize these cases, its recognition result ‘oelavecl’ is notably closer to the ground truth labels, showcasing a certain degree of robustness against Adversarial noise. Ultimately, **AdvTrain**, benefiting from the integration of difficult adversarial examples during the training process, exhibits enhanced resistance to adversarial attacks. ❺ In terms of blur perturbation, as shown in (b), the **AdvTrain** correctly recognizes the motion-blurred word with kernel size of 3, while **GauBlurTrain** failed by recognizing it as ‘deloyed’. When the kernel size increases to 5, the recognition result of **AdvTrain**, *i.e.*, ‘dulnyud’, is more similar to the ground truth ‘delayed’ than the recognition result of **GauBlurTrain**, *i.e.*, ‘ulilijuul’. These results indicate the significant robustness of the additive adversarial training model, **AdvTrain**, against perturbations with other patterns.

V. Conclusion

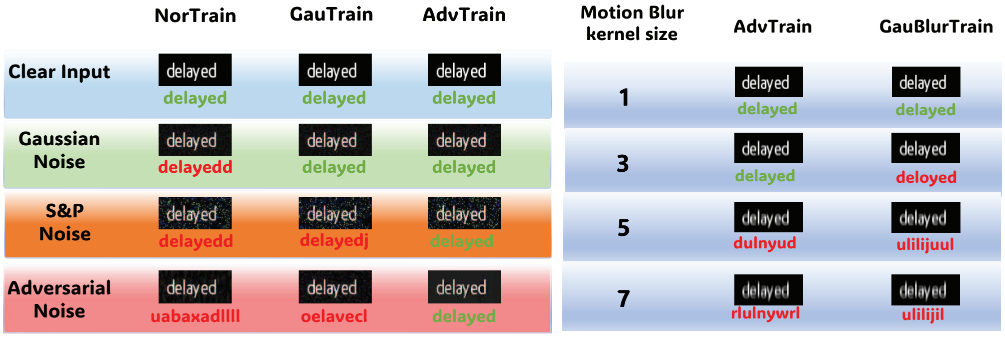
This paper has explored the promising topic of utilizing adversarial training techniques to enhance the robustness of OCR models in HMI testing scenarios, whilst maintaining high model accuracy. By delving into the mechanisms of adversarial attacks and their impact to OCR models, we demonstrated their potential to expose vulnerabilities and weaknesses within these models.

Our findings showed that OCR models trained using adversarial training techniques perform better than the vanilla model, and even the models trained with noise data augmentation. In the experiments, the adversarial training model also demonstrated a certain level of robustness even against perturbations from other patterns. It underscored the significance of adversarial training as a mitigation strategy, that fortifies the OCR models against perturbations added during the collection phase of the HMI images, and also contributed to a deeper understanding of the models and the HMI images.

As the field of adversarial machine learning continues to evolve, the insights and methodologies presented in this paper serve as a stepping stone for further research and innovation. Ultimately, by harnessing the power of adversarial attacks to bolster the reliability and effectiveness of OCR models, we move closer to achieving a more dependable and automated HMI testing strategy in an increasingly digitized world.

Acknowledgements

This study is supported under the RIE2020 Industry Alignment Fund – Industry Collaboration Projects (IAF-ICP) Funding Initiative, as well as cash and in-kind contribution from the industry partner(s).



(a) Robustness against additive noise.

(b) Robustness against motion blur.

Fig. 6: The recognition results of word ‘delayed’ are provided, including the clear version and perturbed samples. The recognition result for each image is shown at the bottom. Correct predictions are indicated by green text, while incorrect predictions are shown in red. (a) From top to bottom, each row corresponds to the clear input, Gaussian noise, and adversarial noise, respectively. The columns labeled **NorTrain**, **GauTrain**, and **AdvTrain** indicate the model that generated the recognition results in their respective columns. (b) From the top row to the bottom row, the three rows respectively represent a motion-blurred image with kernel sizes of 1, 3, 5, and 7 (1 indicates clear image). The columns labeled **AdvTrain** and **GauBlurTrain** indicate the model that generated the recognition results in their respective columns.

REFERENCES

[1] Google 10000 english. https://github.com/first20hours/google-10000-english. 5

[2] Anish Athalye, Nicholas Carlini, and David Wagner. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In *International conference on machine learning*, pages 274–283. PMLR, 2018. 2

[3] Yonatan Belinkov and Yonatan Bisk. Synthetic and natural noise both break neural machine translation. *arXiv preprint arXiv:1711.02173*, 2017. 3

[4] Anselm Blumer, Andrzej Ehrenfeucht, David Haussler, and Manfred K Warmuth. Occam’s razor. *Information processing letters*, 24(6):377–380, 1987. 5

[5] Abdelhak Boukharouba and Abdelhak Bennia. Novel feature extraction technique for the recognition of handwritten digits. *Applied Computing and Informatics*, 13(1):19–26, 2017. 2

[6] Ajay Kumar Boyat and Brijendra Kumar Joshi. A review paper: noise models in digital image processing. *arXiv preprint arXiv:1505.03489*, 2015. 5

[7] Yushi Cao, Yan Zheng, Shang-Wei Lin, Yang Liu, Yon Shin Teo, Yuxuan Toh, and Vinay Vishnumurthy Adiga. Automatic hmi structure exploration via curiosity-based reinforcement learning. In *2021 36th IEEE/ACM International Conference on Automated Software Engineering (ASE)*, pages 1151–1155. IEEE, 2021. 1

[8] Lu Chen and Wei Xu. Attacking optical character recognition (ocr) systems with adversarial watermarks. *arXiv preprint arXiv:2002.03095*, 2020. 2

[9] Dan Claudiu Cireşan, Ueli Meier, Luca Maria Gambardella, and Jürgen Schmidhuber. Deep, big, simple neural nets for handwritten digit recognition. *Neural computation*, 22(12):3207–3220, 2010. 2

[10] Yinpeng Dong, Fangzhou Liao, Tianyu Pang, Hang Su, Jun Zhu, Xiaolin Hu, and Jianguo Li. Boosting adversarial attacks with momentum. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9185–9193, 2018. 2, 3

[11] Yinpeng Dong, Tianyu Pang, Hang Su, and Jun Zhu. Evading defenses to transferable adversarial examples by translation-invariant attacks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4312–4321, 2019. 3

[12] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In *International Conference on Learning Representations*, 2015. 2

[13] H Herbert. The history of ocr, optical character recognition. *Manchester Center, VT: Recognition Technologies Users Association*, 1982. 2

[14] Noman Islam, Zeeshan Islam, and Nazia Noor. A survey on optical character recognition system. *arXiv preprint arXiv:1710.05703*, 2017. 2

[15] Alexey Kurakin, Ian Goodfellow, and Samy Bengio. Adversarial examples in the physical world. *arXiv preprint arXiv:1607.02533*, 2016. 2

[16] J Pradeep, E Srinivasan, and S Himavathi. Neural network based recognition system integrating feature extraction and classification for english handwritten. *International journal of Engineering*, 25(2):99–106, 2012. 2

[17] Kevin Roth, Yannic Kilcher, and Thomas Hofmann. Adversarial training is a form of data-dependent operator norm regularization. *Advances in Neural Information Processing Systems*, 33:14973–14985, 2020. 4

[18] Albrecht Schmidt, Anind K Dey, Andrew L Kun, and Wolfgang Spiessl. Automotive user interfaces: human computer interaction in the car. In *CHI’10 Extended Abstracts on Human Factors in Computing Systems*, pages 3177–3180. 2010. 1

[19] Ali Shafahi, Mahyar Najibi, Zheng Xu, John Dickerson, Larry S Davis, and Tom Goldstein. Universal adversarial training. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 5636–5643, 2020. 2

[20] Eman Shaikh, Iman Mohiuddin, Ayisha Manzoor, Ghazanfar Latif, and Nazeeruddin Mohammad. Automated grading for handwritten answer sheets using convolutional neural networks. In *2019 2nd International conference on new trends in computing sciences (ICTCS)*, pages 1–6. Ieee, 2019. 2

[21] Baoguang Shi, Xiang Bai, and Cong Yao. An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition. *IEEE transactions on pattern analysis and machine intelligence*, 39(11):2298–2304, 2016. 1, 4

[22] Congzheng Song and Vitaly Shmatikov. Fooling ocr systems with adversarial text images. *arXiv preprint arXiv:1802.05385*, 2018. 2

[23] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. In *International Conference on Learning Representations*, 2014. 2

[24] J Wagner, R Bruntsy, K Kastery, D Eagany, and D Anthony. A vision for automotive electronics hardware-in-the-loop testing. *International journal of vehicle design*, 22(1-2):14–28, 1999. 1

[25] Cihang Xie, Zhishuai Zhang, Yuyin Zhou, Song Bai, Jianyu Wang, Zhou Ren, and Alan L Yuille. Improving transferability of adversarial examples with input diversity. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2730–2739, 2019. 2