Adversarial Machine Learning Attacks on Automatic Number Plate Recognition Systems

Álvaro de Castro
University of Málaga
alvarodc@uma.es

December 13, 2025

Abstract

This report presents a comprehensive study of adversarial machine learning attacks targeting Automatic Number Plate Recognition (ANPR) systems that utilize YOLO for plate detection and PaddleOCR for character recognition. Three distinct attack methodologies are investigated: (1) Denial of Service attacks using FGSM perturbations to prevent plate detection, (2) Targeted region transfer attacks to map detected plates to alternate license plate regions while maintaining imperceptibility, and (3) Untargeted FGSM-based OCR attacks to cause character misrecognition. Each attack demonstrates unique characteristics regarding stealth, detectability, and practical feasibility in real-world scenarios.

Chapter 1

Introduction

1.1 Motivation and Context

Automatic Number Plate Recognition (ANPR) systems have become critical infrastructure components in traffic management, toll collection, border security, and law enforcement applications. However, the widespread deployment of deep learning models in these systems introduces significant security vulnerabilities to adversarial attacks.

Traditional ANPR pipelines typically consist of two primary components: (1) object detection for license plate localization, and (2) optical character recognition (OCR) for digit and character extraction. The robustness of these systems against maliciously crafted perturbations has been questioned during the latest research, which found that they are vulnerable to adversarially crafted perturbations that cause denials of service and lead to misclassification or targeted classification upon an adversary's will, while remaining almost imperceptible to human observers.

The motivation of this work is to systematically investigate adversarial attacks on a representative two-stage ANPR pipeline combining YOLO for plate detection and PaddleOCR for character recognition. We focus both on black-box and white-box attacks where the adversary has full knowledge of model architectures and parameters, representing a worst-case security analysis. Three attack scenarios are considered: (1) denial-of-service attacks that cause detection failure in a stealthy manner, (2) targeted misrecognition attacks that force the OCR system to output adversarial plate numbers and (3), untargeted OCR attack that goes through the whole pipeline and targets directly to PaddleOCR, causing misrecognition of plate characters while remaining practically imperceptible to human observers.

The context scenario proposed and evaluated along this work is illustrated in Figure 1.1

1.2 Objectives

This project investigates three complementary adversarial attack strategies:

1. Detection-focused attacks that prevent plate localization.
2. Region transfer attacks that alter plate identity while maintaining partial visual coherence.
3. OCR-focused attacks that compromise character recognition accuracy while going unnoticed.

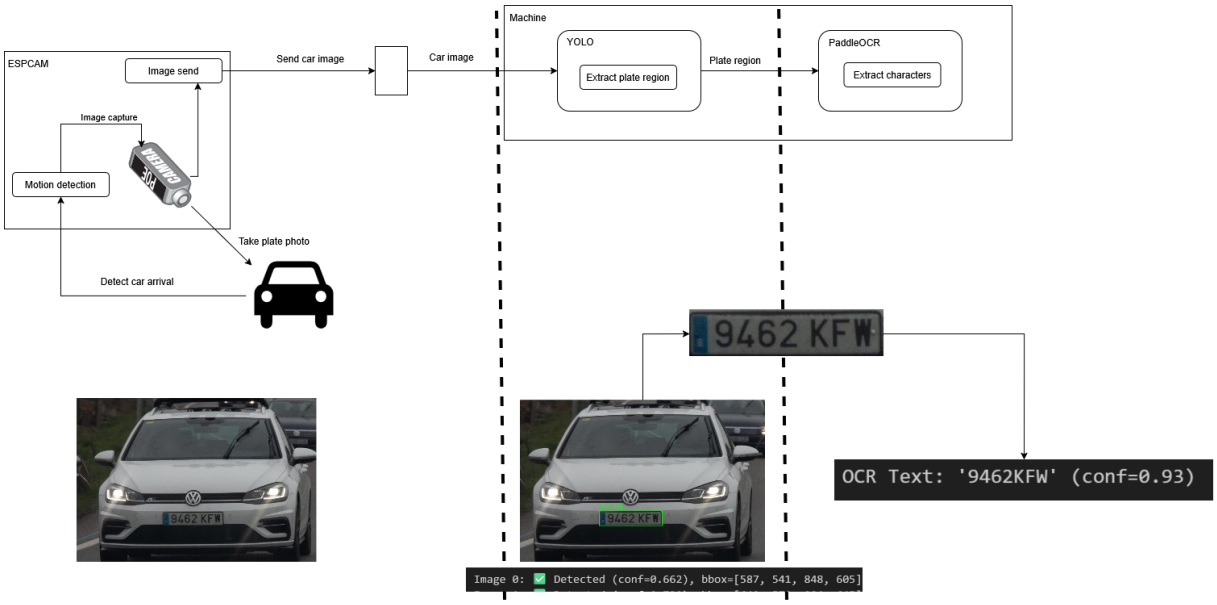

Figure 1.1: General overview of ANPR system proposed

The objective is not merely to demonstrate vulnerability in the proposed system, but to provide a comprehensive generalized framework for evaluating ANPR security, quantifying attack success rates across different perturbation budgets, and understanding the trade-offs between attack effectiveness, imperceptibility, and computational cost.

1.3 Structure of the Document

This document is structured as follows. Chapter 2 reviews the theoretical foundations of adversarial machine learning, ANPR system architectures, and related work on adversarial attacks against object detection and OCR systems. Chapter 3 describes the threat models proposed for the presented attacks within the ANPR implementation. Chapter 4 reviews the proposed attacks, including objectives, concrete implementation and techniques applied for attack feasibility. Chapter 5 a brief comparative of attacks proposed, targets, features of interest and implications and Chapter 6 concludes the work outlining future research directions.

Chapter 2

Background

2.1 Adversarial Machine Learning

Adversarial examples are inputs crafted by introducing small perturbations to legitimate inputs with the intent of misleading machine learning models. These techniques can be applied before or during the testing and training phases of the model. Figure 2.1 illustrates a comprehensive topology presented by [7] for classifying Adversarial Machine Learning attacks.

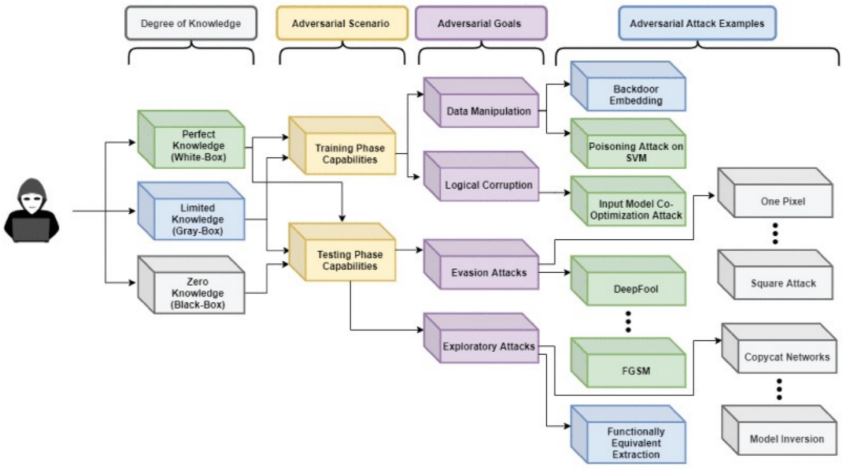

Figure 2.1: Adversarial Machine Learning Attacks topology

Many existing research shows how Machine Learning is vulnerable to carefully crafted adversarial samples [1] [5] [9] while remaining mostly imperceptible to human eye.

2.2 YOLO for License Plate Detection

You Only Look Once (YOLO) is a single-stage object detector that predicts bounding boxes and class probabilities directly from full images in one evaluation. For ANPR applications, YOLO models are trained to detect license plate regions in images. These models output objectness scores, localization coordinates, and class confidence scores [14].

2.3 PaddleOCR for Character Recognition

PaddleOCR is a comprehensive OCR toolkit that performs text detection and character recognition using convolutional and recurrent neural networks [4]. In ANPR systems, PaddleOCR is responsible for extracting alphanumeric characters from cropped license plate regions identified by YOLO.

2.4 ANPR systems

Automatic Number Plate Recognition (ANPR) systems combine image acquisition, object detection, and optical character recognition (OCR) to automatically read vehicle registration plates in real time [10]. Typical deployments integrate high-resolution cameras with infrared illumination, a detection module that localizes the plate region, and a recognition module that extracts and decodes the alphanumeric content. These systems are widely used for law enforcement, toll collection, parking management, and traffic analytics, where incorrect plate readings can directly impact billing, access control, and forensic investigations [13].

Modern ANPR pipelines increasingly rely on deep convolutional networks for both license plate detection and character recognition [15]. Deep models provide higher accuracy under challenging conditions such as varying illumination, motion blur, and complex backgrounds, but they also inherit the security weaknesses of neural networks, especially their susceptibility to adversarial examples [8]. In license plate recognition, adversarial examples correspond to carefully perturbed plate images that appear unchanged to humans but induce mislocalization or misrecognition of the plate by the system.

Recent work has shown that real Licence Plate Recognition systems are widely vulnerable to adversarial samples [6], even when perturbations remain visually subtle. Some existing work, focus their effort in the attack chain to OCR, concretely to Tesseract model [12], which is based on Deep Learning and frequently used in real systems. Several techniques are used in literature to attack ANPR systems, from Watermarking technique [2] [3] for attacking OCR Deep Neural Network systems to genetic algorithms that generate suitable perturbations and success for around 93% of cases [11].

Chapter 3

Threat Model and Attack Objectives

We consider two different threat models, based on adversary capabilities and purpose.

First, we assume a black-box threat model (Figure 3.1) where the attacker has no previous knowledge of the system architecture itself, detection model or its parameters, or technologies being used underneath. This threat model better matches what real-world scenario looks like, where the attacker only have the possibility of interfering between field device *ESPCAM* and machine running detection model, acting as an *Adversary in the Middle* (*T1557* [1]).

Secondly, we assume a white-box threat model (Figure 3.2) where the adversary has full knowledge of the YOLO detection model and PaddleOCR character recognition model, including model parameters, architecture, and loss functions. This assumption allows for gradient-based attack optimization and matches an ideal scenario.

The attacks are evaluated under the following constraints:

- Perturbations are to be the minimum possible to success in the attack planned while reducing noise and variance from the original sample.
- Human imperceptibility is maintained where applicable when possible.
- Attacks are applied only to specific regions of interest (plate region or plate content)

[1] https://attack.mitre.org/techniques/T1557/

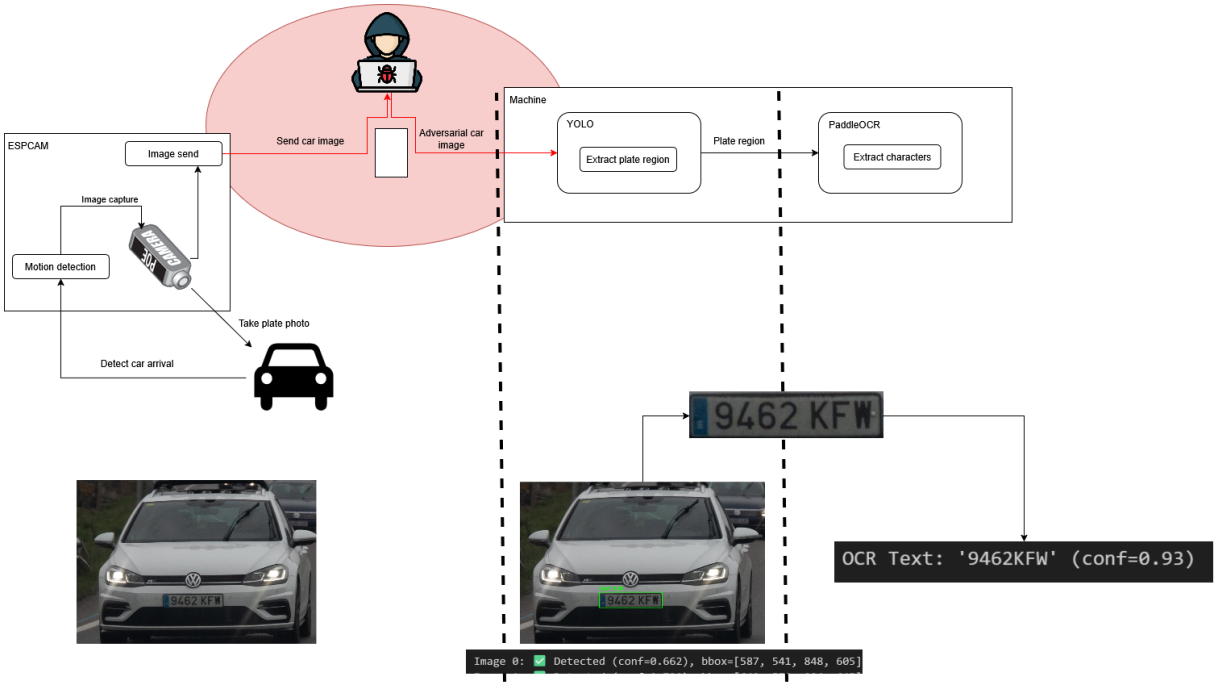

Figure 3.1: Black-box adversary threat model

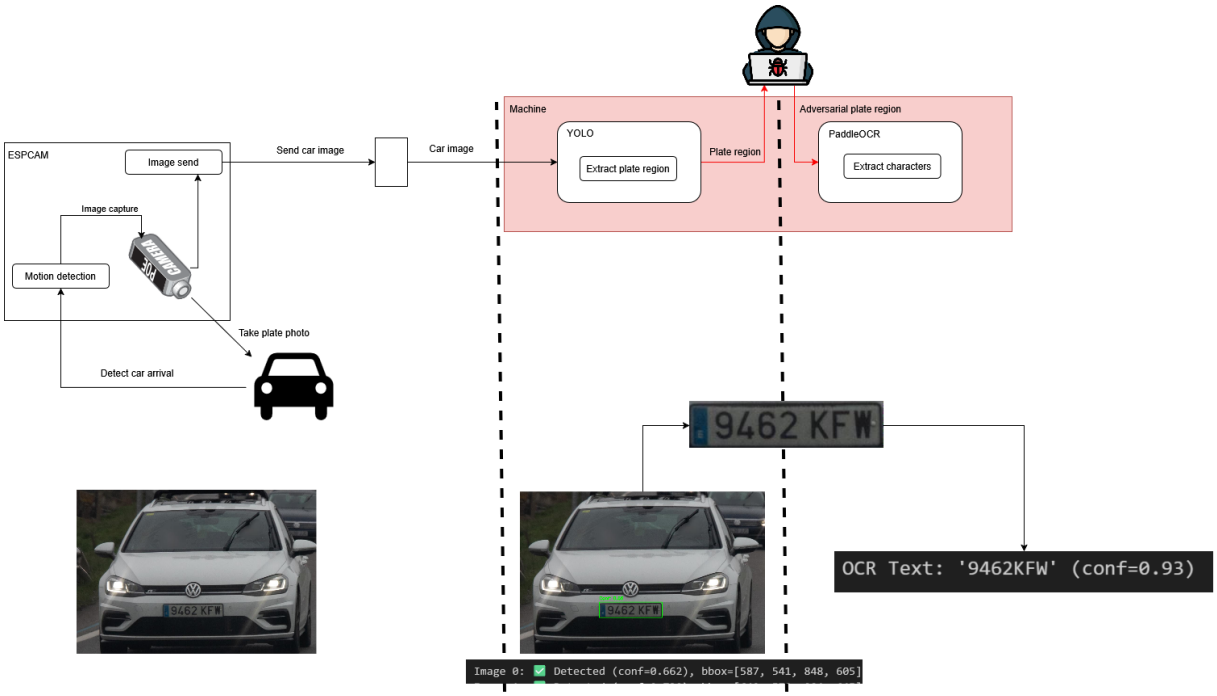

Figure 3.2: White-box adversary threat model

Chapter 4

Attack Methodologies proposed

4.1 Attack 1: Denial of Service via FGSM-Based Detection Evasion

4.1.1 Objective

The first attack aims to generate adversarial perturbations that prevent YOLO from detecting license plate regions. This represents a denial-of-service attack where the attacker seeks to minimize the objectness score of the plate region.

It is important to note, that without prior knowledge on the confidence threshold being used by YOLO, in the attacked system, we could not anticipate perturbation necessity for the classifier to fail. For the studied case, an aggressive attack will be performed and we will try to drop confidence to 0, thus incrementing amount of noise used. In real-world scenarios, confidence thresholds are usually greater than 50% of confidence.

4.1.2 Technical Approach

For each plate image $\mathbf{x}$ with bounding box annotation $\mathbf{b}$, we compute adversarial perturbations $\boldsymbol{\delta}$ that minimize the YOLO detection confidence score for image $\mathbf{x}$.

In practice, we use an iterative algorithm that incrementally reduces object detection confidence by increasing perturbation budget ε . The pseudocode used is shown in Algorithm 1.

Algorithm 1 DoS Attack via FGSM Detection Evasion

Require: Image $\mathbf{sample}_{\text{original}}$, YOLO model M, perturbation budget ε
Ensure: Adversarial example $\mathbf{x}_{\text{adv}}$

1: **for** all ε **do**
2: $\quad\mathbf{sample}_{\text{adversary}} \leftarrow perturbate(\mathbf{sample}_{\text{original}}, \varepsilon, targetRegion)$ ▷ Get adversary sample
3: $\quad result \leftarrow M.predict(sample_{\text{adversary}})$ ▷ Predict result
4: $\quad$**if** result is not detected **then return** $\mathbf{x}_{\text{adv}}$
5: $\quad$**end if**
6: **end for**

Figure 4.1: Adversarial DoS Attacks

4.1.3 Perturbation Localization

To enhance stealth and reduce detectability, perturbations are applied only to the license plate region identified during detection. Following this approach, visual impact is reduced as much as possible, and let adversarial sample modifications to go unnoticed for human operators checking the input image. Figure 4.1 shows three different samples processed to obtain adversarial samples that led YOLO confidence drop to 0.

If we look closer, Figure 4.2 shows the original untouched image, while Figure 4.3 shows adversarial image, which YOLO is not able to recognise.

Figure 4.2: Original images before DoS perturbation

Figure 4.3: Adversarial image after DoS perturbation

Note: While individual perturbations may pass human scrutiny, statistical analysis such as Error Level Analysis (ELA) or feature-space anomaly detection might identify such attacks. However, the attack's success relies on the fact that perturbations appear as natural plate degradation.

4.1.4 Attack 2: Targeted Region Transfer

4.1.5 Objective

The second attack aims to create adversarial perturbations through deterministic image blending that causes YOLO to redirect its localization from a source license plate region to a target plate region. Rather than using iterative gradient optimization, this attack employs a overlay blending strategy with the application of concrete perturbations to transfer visual appearance from a target detection to a source region, effectively causing mislocalization while maintaining plate presence in the image.

In contrast to Attack 1, this method performs a one-shot, deterministic transformation of the source plate region with a relatively high perturbation budget. For evaluation and resulting metrics visualization only, the resulting adversarial image is then re-evaluated by YOLO, and the quality of the attack is measured by how similar YOLO's final detection is to the original target bounding box (making use of Intersection over Union, IoU), while visually preserving a plausible plate-like patch in the original source location.

4.1.6 Technical Approach

Given a source image $\mathbf{x}_s$ with detected source bounding box $\mathbf{b}_s$ and a target image $\mathbf{x}_t$ with target bounding box $\mathbf{b}_t$, the attack operates only on the corresponding ROIs. The target ROI is first resized to match the spatial resolution of the source ROI, and then a linear intensity transformation is applied to enforce a non-trivial difference to ensure attack success in mismatching ANPR prediction.

Let $\mathbf{x}_s$ and $\mathbf{x}_t$ denote the source and target RGB images, and $\mathbf{b}_s = (x_1^{(s)}, y_1^{(s)}, x_2^{(s)}, y_2^{(s)})$, $\mathbf{b}_t = (x_1^{(t)}, y_1^{(t)}, x_2^{(t)}, y_2^{(t)})$ be the clamped YOLO bounding boxes. The corresponding ROIs are:

$$\text{src\_roi} = \mathbf{x}_s[y_1^{(s)} : y_2^{(s)},\ x_1^{(s)} : x_2^{(s)}], \quad \text{tgt\_roi} = \mathbf{x}_t[y_1^{(t)} : y_2^{(t)},\ x_1^{(t)} : x_2^{(t)}].$$

If either ROI is empty (zero size), the attack aborts. Otherwise, the target ROI is resized to the exact spatial resolution of the source ROI using OpenCV's bilinear interpolation:

$$\text{tgt\_roi\_resized} = \text{resize}(\text{tgt\_roi},\ W_{\text{src}}, H_{\text{src}}),$$

A first overlay is then built by applying a linear intensity transform implemented by `cv2.convertScaleAbs`:

$$\text{overlay} = \text{convertScaleAbs}(\text{tgt\_roi\_resized}, \alpha = 1.15, \beta = 10),$$

which corresponds to per-pixel operations of the form $p' = |\alpha p + \beta|$ followed by clipping to $[0, 255]$ in 8-bit space. The raw perturbation δ inside the ROI is:

$$\boldsymbol{\delta} = \text{overlay} - \text{src\_roi},$$

To guarantee a sufficient perturbation for ensuring the attack success, if $\text{max\_abs} < 1.5$ the overlay is recomputed with stronger parameters.

Finally, the perturbation is limited (clamped) to a maximum and minimum value limited by the original ϵ used:

$$\epsilon = 80.0, \qquad \boldsymbol{\delta}_{\text{clamped}} = \text{clip}(\boldsymbol{\delta}, -\epsilon, \epsilon),$$

and applied to the original source ROI:

$$\text{adv\_roi} = \text{clip}(\text{src\_roi} + \boldsymbol{\delta}_{\text{clamped}}, 0, 255)$$

After perturbation application, a mild Gaussian blur is applied only inside this ROI to minimize as much as possible visual artifacts.

Algorithm 2 summarizes the overall execution procedure for the targeted region transfer attack.

Algorithm 2 Targeted Region Transfer via Deterministic Image Blending

Require: Source image $\mathbf{x}_s$, target image $\mathbf{x}_t$, YOLO model M, perturbation budget ε
Ensure: Adversarial source image $\mathbf{x}_{\text{adv}}$

$\mathbf{b}_s \leftarrow \text{first\_bbox}\big(M.\text{predict}(\mathbf{x}_s)\big)$
$\mathbf{b}_t \leftarrow \text{first\_bbox}\big(M.\text{predict}(\mathbf{x}_t)\big)$
if $\mathbf{b}_s$ or $\mathbf{b}_t$ is invalid **then**
    return $\mathbf{x}_s$ ▷ Abort if either detection fails
end if
$\text{src\_roi} \leftarrow \mathbf{x}_s[\mathbf{b}_s]$, $\text{tgt\_roi} \leftarrow \mathbf{x}_t[\mathbf{b}_t]$
$\text{tgt\_resized} \leftarrow \text{resize}(\text{tgt\_roi}, \text{shape}(\text{src\_roi}))$
$\text{overlay} \leftarrow \text{convertScaleAbs}(\text{tgt\_resized}, \alpha_1, \beta_1)$
$\boldsymbol{\delta} \leftarrow \text{overlay} - \text{src\_roi}$
if $\max|\boldsymbol{\delta}| < \tau$ **then**
    $\text{overlay} \leftarrow \text{convertScaleAbs}(\text{tgt\_resized}, \alpha_2, \beta_2)$
    $\boldsymbol{\delta} \leftarrow \text{overlay} - \text{src\_roi}$
end if
$\boldsymbol{\delta} \leftarrow \text{clip}(\boldsymbol{\delta}, -\varepsilon, \varepsilon)$
$\text{adv\_roi} \leftarrow \text{clip}(\text{src\_roi} + \boldsymbol{\delta}, 0, 255)$
$\text{adv\_roi} \leftarrow \text{GaussianBlur}(\text{adv\_roi}, k(\text{src\_roi}))$
$\mathbf{x}_{\text{adv}} \leftarrow \mathbf{x}_s$; replace $\mathbf{x}_{\text{adv}}[\mathbf{b}_s] \leftarrow \text{adv\_roi}$
$\text{result} \leftarrow M.\text{predict}(\mathbf{x}_{\text{adv}})$
$\mathbf{b}_{\text{final}} \leftarrow \text{first\_bbox}(\text{result})$
$\text{IoU} \leftarrow \text{IoU}(\mathbf{b}_{\text{final}}, \mathbf{b}_t)$
return $\mathbf{x}_{\text{adv}}$

4.1.7 Perturbation Localization

As in the Denial of Service attack, perturbations are strictly localized to the license plate region predicted in the source image. The rest of the frame remains untouched, which helps maintain overall scene consistency and limits the scope of visible changes to a compact area around the plate. Figure 4.4 illustrates an example of source, Figure 4.5 shows the target used and Figure 4.6 the adversarial image resulting for this attack.

While the plate area in the adversarial image clearly differs from the original when examined closely, the modification still resembles a plausible license plate patch embedded in the same position. Compared to Attack 1, the perturbation is less stealthy but more structurally meaningful,

Figure 4.4: Original source image

Figure 4.5: Original target image

Figure 4.6: Adversarial source after region transfer

as it actively injects the appearance of another plate rather than simply adding noise.

Note: Similar to the previous attack, automatic forensic tools such as Error Level Analysis, frequency-domain inspection, or learned anomaly detectors could reveal the presence of synthetic blending artifacts. At the same time, the deterministic nature of the transformation and the fact that the rest of the image is untouched may allow the attack to bypass naive defenses that only look for global inconsistencies or random high-frequency noise.

4.2 Attack 3: Untargeted OCR Attack with Imperceptible Perturbations

4.2.1 Objective

This attack generates adversarial perturbations targeting the PaddleOCR character recognition model. The objective is to cause character misrecognition while maintaining imperceptibility to human observers.

In contrast to Attacks 1 and 2, which operate directly on YOLO's detection behaviour, this method treats YOLO as a fixed pre-processing step and attacks the cropped plate ROI. The attack is untargeted, so any OCR prediction different from the original will be considered.

For the attack to be stealthy, and in order to success in OCR misrecognition, this attack is considered only for the proposed White-box Threat Model in Chapter 3, where the attacker is able to know OCR output result to optimise the parameters within the search space.

4.2.2 Technical Approach

Given a cropped license plate image $\mathbf{x}_p$ and its ground-truth character sequence $\mathbf{y}$, we generate perturbations $\boldsymbol{\delta}$ that maximize the OCR misclassification loss until a mistaken recognition has occurred.

We employ an iterative algorithm combined with early stopping to ensure imperceptibility. Pseudocode illustrating the functionality is shown in Algorithm 3.

4.2.3 Imperceptibility Preservation

To ensure human imperceptibility, we employ several strategies:

1. **Edge-focused mask:** The `build_edge_mask` function confines most of the noise to edges and character strokes detected by Canny, which makes perturbations blend with existing high-frequency content rather than creating flat blotches in uniform background areas.

Algorithm 3 Untargeted OCR Attack with Imperceptibility Constraint

Require: Plate image $\mathbf{x}_p$, OCR model M_{ocr}, ground-truth labels $\mathbf{y}$, $\varepsilon_{\max}$, N_{iter}
Ensure: Adversarial example $\mathbf{x}_{\text{adv}}$, perturbation magnitude ε_{opt}

$\mathbf{x}_{\text{adv}} \leftarrow \mathbf{x}_p$
$\varepsilon_{\text{opt}} \leftarrow 0$
for $\varepsilon = \varepsilon_{\min}$ to $\varepsilon_{\max}$ step $\Delta\varepsilon$ **do**
    $\mathbf{x}_{\text{trial}} \leftarrow \mathbf{x}_p$
    for $i = 1$ to N_{iter} **do**
        $\hat{\mathbf{y}} \leftarrow M_{\text{ocr}}(\mathbf{x}_{\text{trial}})$ ▷ Get OCR predictions
        if $\hat{\mathbf{y}} \neq \mathbf{y}$ **then** ▷ Check for misclassification
            $\varepsilon_{\text{opt}} \leftarrow \varepsilon$
            break
        end if
        $\ell \leftarrow \mathcal{L}_{\text{OCR}}(\hat{\mathbf{y}}, \mathbf{y})$
        $\mathbf{g} \leftarrow \nabla_{\mathbf{x}_{\text{trial}}} \ell$
        $\mathbf{x}_{\text{trial}} \leftarrow \mathbf{x}_{\text{trial}} + \alpha \cdot \text{sign}(\mathbf{g})$
        $\mathbf{x}_{\text{trial}} \leftarrow \text{clip}(\mathbf{x}_{\text{trial}}, \mathbf{x}_p - \varepsilon, \mathbf{x}_p + \varepsilon)$
    end for
end for
$\mathbf{x}_{\text{adv}} \leftarrow \text{generate\_attack}(\mathbf{x}_p, \varepsilon_{\text{opt}})$ **return** $\mathbf{x}_{\text{adv}}, \varepsilon_{\text{opt}}$

2. **Bounded L_∞ budget and ROI-only updates:** All updates are clipped to $\pm\varepsilon$ relative to the original plate ROI, and no pixels outside the YOLO plate bounding box are touched. This allows to minimize applied perturbations from source image.

3. **Bilateral filtering:** After each update, a bilateral filter is applied that denoises small isolated artifacts while preserving edges, making the result look more like a naturally noisy or compressed plate image than a synthetic pattern.

4. **Early stopping:** The search over $(\varepsilon, \text{steps})$ terminates as soon as any configuration produces a different OCR string, instead of continuing to add more noise. This prevents unnecessary over-perturbation once misrecognition has been achieved.

4.2.4 Perturbation localization

Following already related approach, perturbation are only applied to license plate region. To human observer, perturbations are mostly imperceptible. Figure 4.7 shows the original image along with the adversary image created using a value of $\epsilon = 6$ and $num\_steps = 10$.

All the modifications, let the adversary modification mostly imperceptible while plate number prediction by PaddleOCR is mistaken. Figure 4.8 shows the real perturbations applied, magnified for being visible to the reader.

The presented adversary image, is correctly mismatched by Paddle OCR, which predicts a plate number `9462KEW`. The output from the attack is presented in Figure 4.9.

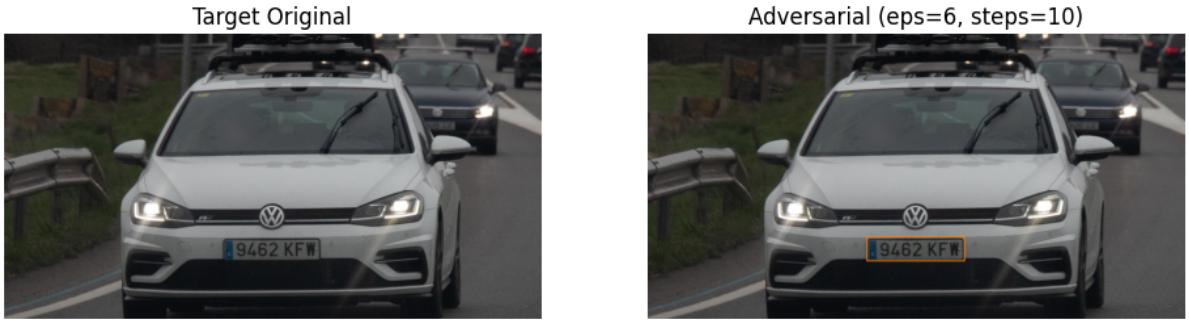

Figure 4.7: Original image and adversary image for OCR attack

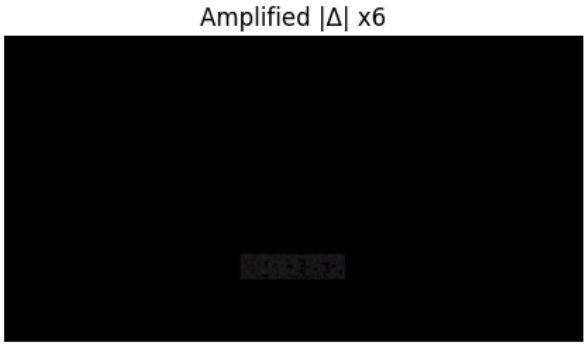

Figure 4.8: OCR attack perturbation amplified

```
[2025/12/08 12:11:24] ppocr DEBUG: rec_res num  : 1, elapsed : 0.09980344772338867
[2025/12/08 12:11:24] ppocr DEBUG: rec_res num  : 1, elapsed : 0.09980344772338867
[2025/12/08 12:11:24] ppocr DEBUG: dt_boxes num : 1, elapsed : 0.01679396629333496
[2025/12/08 12:11:24] ppocr DEBUG: cls num  : 1, elapsed : 0.009624004364013672
...
✅ Found change with eps=6, steps=8: '9462KFW' -> '9462KEW' (conf=0.91)
eps= 6, steps= 8 | step 4/8 | text='9462KEW' (conf=0.91)

✅ Found change with eps=6, steps=8: '9462KFW' -> '9462KEW' (conf=0.91)
```

Figure 4.9: Plate misrecognition from OCR

Chapter 5

Comparative Analysis

Table 5.1: Comparison of Attack Methodologies

Characteristic	**Detection DoS**	**Region Transfer**	**OCR Imperceptible**
Target Component	YOLO Detector	Plate Region	PaddleOCR
Attack Type	Targeted	Targeted	Untargeted
Perturbation Scope	Plate Region	Plate Region	Plate Region
Visual Imperceptibility	High	Low-Medium	Very High
Human Detectability	Low	Medium–High	Very Low
Success Rate	80–95%	70–85%	90–95%
Practical Feasibility	High	Medium	High

5.1 Trade-offs and Implications

5.1.1 DoS Attack

Provides complete plate non-detection with minimal visual artifacts. Highly practical for attacking specific plates while remaining stealthy. However, repeated attacks on the same location might trigger statistical anomalies.

5.1.2 Region Transfer Attack

Enables sophisticated plate substitution scenarios but introduces visible artifacts. Better suited for scenarios requiring specific misidentification targets. Detection risk is elevated due to visual inconsistencies.

5.1.3 OCR Attack

Achieves character-level misrecognition with maximum imperceptibility. Highly resistant to human detection. Optimal for scenarios where the attacker aims to introduce subtle, undetectable errors into ANPR logs.

Chapter 6

Conclusion

This project demonstrates three distinct adversarial attack methodologies targeting ANPR systems. Each attack exploits fundamental vulnerabilities in deep learning-based detection and recognition pipelines, offering different trade-offs between effectiveness, stealth, and detectability. The DoS attack prioritizes non-detection through plate-region perturbations; the region transfer attack enables targeted plate substitution; and the OCR attack achieves imperceptible character misrecognition.

These findings emphasize the critical importance of integrating adversarial robustness into ANPR system design and deployment, particularly for critical applications. Future work should focus on developing practical defenses and evaluating attack transferability across diverse ANPR architectures and real-world conditions.

Chapter 7

References

All the code implemented is available in the following Github repository: `https://github.com/alvarodcastro/AdversarialANPR`

Bibliography

[1] N. Carlini and D. A. Wagner, "Towards evaluating the robustness of neural networks," *CoRR*, vol. abs/1608.04644, 2016. arXiv: 1608.04644. [Online]. Available: http://arxiv.org/abs/1608.04644.

[2] L. Chen, J. Sun, and W. Xu, *Fawa: Fast adversarial watermark attack on optical character recognition (ocr) systems*, 2020. arXiv: 2012.08096 [cs.CV]. [Online]. Available: https://arxiv.org/abs/2012.08096.

[3] L. Chen and W. Xu, *Attacking optical character recognition (ocr) systems with adversarial watermarks*, 2020. arXiv: 2002.03095 [cs.CV]. [Online]. Available: https://arxiv.org/abs/2002.03095.

[4] C. Cui et al., *Paddleocr 3.0 technical report*, 2025. arXiv: 2507.05595 [cs.CV]. [Online]. Available: https://arxiv.org/abs/2507.05595.

[5] I. J. Goodfellow, J. Shlens, and C. Szegedy, *Explaining and harnessing adversarial examples*, 2015. arXiv: 1412.6572 [stat.ML]. [Online]. Available: https://arxiv.org/abs/1412.6572.

[6] Z. Gu et al., "Adversarial attacks on license plate recognition systems," *Computers, Materials & Continua*, vol. 65, no. 2, pp. 1437–1452, 2020, ISSN: 1546-2226. DOI: 10.32604/cmc.2020.011834. [Online]. Available: http://www.techscience.com/cmc/v65n2/39886.

[7] S. Y. Khamaiseh, D. Bagagem, A. Al-Alaj, M. Mancino, and H. W. Alomari, "Adversarial deep learning: A survey on adversarial attacks and defense mechanisms on image classification," *IEEE Access*, vol. 10, pp. 102 266–102 291, 2022. DOI: 10.1109/ACCESS.2022.3208131.

[8] J. Malik, R. Muthalagu, and P. M. Pawar, "A systematic review of adversarial machine learning attacks, defensive controls, and technologies," *IEEE Access*, vol. 12, pp. 99 382–99 421, 2024. DOI: 10.1109/ACCESS.2024.3423323.

[9] N. Papernot, P. McDaniel, S. Jha, M. Fredrikson, Z. B. Celik, and A. Swami, *The limitations of deep learning in adversarial settings*, 2015. arXiv: 1511.07528 [cs.CR]. [Online]. Available: https://arxiv.org/abs/1511.07528.

[10] C. Patel, D. Shah, and A. Patel, "Automatic number plate recognition system (anpr): A survey," *International Journal of Computer Applications*, vol. 69, no. 9, 2013.

[11] Y. Qian, Y. Zhang, Y. Liu, and Y. Chen, "Spot evasion attacks: Adversarial examples for license plate recognition systems with convolutional neural networks," *Computers & Security*, vol. 95, p. 101 826, Aug. 2020. DOI: 10.1016/j.cose.2020.101826. [Online]. Available: https://doi.org/10.1016/j.cose.2020.101826.

[12] C. Song and V. Shmatikov, *Fooling ocr systems with adversarial text images*, arXiv preprint arXiv:1802.05385, Feb. 2018. [Online]. Available: https://arxiv.org/pdf/1802.05385.

[13] J. Tang, L. Wan, J. Schooling, P. Zhao, J. Chen, and S. Wei, "Automatic number plate recognition (anpr) in smart cities: A systematic review on technological advancements and application cases," *Cities*, vol. 129, p. 103 833, 2022, ISSN: 0264-2751. DOI: https://doi.org/10.1016/j.cities.2022.103833. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0264275122002724.

[14] A. Vina, *Object detection with yolo11: Ultralytics tutorial*, https://www.ultralytics.com/blog/how-to-use-ultralytics-yolo11-for-object-detection, Ultralytics Blog, Nov. 2024.

[15] Z. Wu, S. Song, Y. Gao, X. Xu, L. Wang, and L. Xie, "Deep learning in security and forensic applications: A survey," *Frontiers in Neurorobotics*, vol. 15, p. 20, 2021. DOI: 10.3389/fnbot.2021.642147. [Online]. Available: https://pmc.ncbi.nlm.nih.gov/articles/PMC8123416/.