Training Cycle For Telecommunications Engineers

Option : SysTIC

Graduation Project Report

Enhancing Safety and Reliability of Object Detection in Aerial Imagery through Explainable AI

Realized by:

Malek BEN HASSINE

Professional Supervisors:

Dr. Reza Bahmanyar

Dr. Houda Chaabouni

Academic Supervisor:

Dr. Hajer Tounsi

Work proposed and fulfilled in collaboration with:

Academic year : 2024–2025

Higher School of Communications of Tunis – SUP'COM

Elghazala Technopark, Ariana 2083, Tunisia

Tel: +216 70 240 900 – Website: `https://www.supcom.tn`

Abstract

Object detection in aerial imagery plays a critical role in safety-critical applications including disaster response, surveillance, and autonomous systems, yet modern deep learning detectors operate as black boxes, providing limited insight into their decision-making processes. This lack of interpretability presents significant challenges for building trustworthy AI systems where understanding why a model makes specific predictions is as important as achieving high accuracy. This project addresses this gap by systematically investigating explainable AI (XAI) techniques for aerial object detection and developing a novel method that overcomes fundamental limitations of existing approaches.

Working with the EAGLE dataset containing 215,986 annotated vehicles across 748 high-resolution aerial images, we trained YOLOv11l, achieving mAP_{50} of 0.78. We conducted a comprehensive evaluation of both black-box XAI methods (D-RISE, D-CLOSE, SODEx) and white-box methods (Grad-CAM, HiResCAM) adapted for object detection. Our analysis revealed an unavoidable trade-off: black-box methods achieve faithful explanations but require 1,000–5,000 forward passes per explanation, while white-box methods offer single-pass efficiency but produce spatially inaccurate explanations due to sparse gradients and naive upsampling that ignore true receptive fields.

To overcome this limitation, we propose **Receptive-Field-Based HiResCAM**, which explicitly reconstructs each grid cell's receptive field through input-image gradient computation, accounting for complex architectural elements including Feature Pyramid Networks and multi-scale detection heads. Quantitative evaluation using insertion–deletion metrics demonstrates that our method achieves faithfulness scores of 0.589–0.868, matching or exceeding black-box methods while being an order of magnitude faster computationally. The generated explanations reveal that the detector relies primarily on vehicle roofs, shadows, and contextual information, providing actionable insights for systematic model improvement.

This work makes several key contributions: (1) the first comprehensive evaluation of XAI techniques specifically for small object detection in aerial imagery, (2) a novel explanation method that breaks the conventional efficiency–faithfulness trade-off through principled receptive field reconstruction, (3) demonstration that white-box-level efficiency and black-box-level faithfulness are simultaneously achievable, and (4) practical insights enabling interpretable, trustworthy object detection systems suitable for mission-critical deployments.

Keywords: Explainable AI, Object Detection, Aerial Imagery, Deep Learning, YOLOv11, Grad-CAM, Receptive Field, Computer Vision, Interpretability, Remote Sensing

Acknowledgments

I would like to thank everyone who supported me throughout this work.

My sincere gratitude goes to the faculty and staff at the ***Higher School of Communications of Tunis - SUP'COM*** *for providing me with the knowledge and guidance that shaped my academic journey.*

I am especially grateful to my supervisors ***Dr. Reza BAHMANYAR****,* ***Dr. Houda Chaabouni****, and* ***Dr. Hajer TOUNSI*** *for their invaluable mentorship, patience, and expertise throughout this project.*

I also thank everyone who contributed to making this internship successful, and the jury members for their time and thoughtful evaluation of this work.

Contents

List of Figures

List of Tables

List of Abbreviations

Abbreviation	Definition
DLR	German Aerospace Center (Deutsches Zentrum für Luft- und Raumfahrt)
EAGLE	Earth observation data for object detection in Aerial images
FPN	Feature Pyramid Network
GAP	Global Average Pooling
Grad-CAM	Gradient-weighted Class Activation Mapping
HiResCAM	High-Resolution Class Activation Mapping
ICT	Information and Communication Technology
IoU	Intersection over Union
LIME	Local Interpretable Model-agnostic Explanations
mAP	mean Average Precision
MFPP	Morphological Fragmental Perturbation Pyramid
NMS	Non-Maximum Suppression
R&D	Research and Development
RGB	Red Green Blue
RISE	Randomized Input Sampling for Explanation
SHAP	SHapley Additive exPlanations
SLIC	Simple Linear Iterative Clustering
SODEx	Surrogate Object Detector Explainer
SPFF	Spatial Pyramid Fusion Feature
XAI	Explainable Artificial Intelligence
YOLO	You Only Look Once

General Introduction

Object detection in aerial imagery has become indispensable for critical applications ranging from disaster response and environmental monitoring to autonomous systems and defense operations. Modern deep learning models, particularly YOLO-based detectors, achieve impressive accuracy in identifying and localizing objects from high-altitude perspectives. However, these models operate as complex "black boxes," providing little insight into their decision-making processes. In safety-critical scenarios where detections influence potentially life-impacting actions, this opacity creates serious challenges: practitioners cannot validate predictions, identify systematic failures, or build trust in automated systems.

This lack of interpretability is especially problematic in aerial imagery, where objects are small (often just a few pixels), appear in diverse contexts, and are viewed from unconventional angles. Understanding why a detector predicts a specific bounding box—or fails to detect an object entirely—requires explainable AI (XAI) techniques capable of revealing which visual features and spatial regions drive model decisions. While XAI methods exist for image classification, their applicability to object detection, particularly for small objects in aerial scenes, remains largely unexplored.

This report presents a comprehensive investigation into explainable AI for aerial object detection, systematically evaluating existing techniques and developing a novel approach that overcomes their fundamental limitations. The work is structured as follows:

Chapter 1 establishes the project foundation by introducing the host institutions—the German Aerospace Center (DLR) and the Digital Research Center of Sfax (CRNS)—and their research capabilities. It presents the motivation for interpretability in aerial object detection and introduces the EAGLE dataset, containing 215,986 annotated vehicles across 748 high-resolution aerial images, which serves as our experimental testbed throughout this work.

Chapter 2 provides the technical foundation through a comprehensive survey of the state of the art. It examines two categories of XAI techniques: black-box methods (D-RISE, D-CLOSE, SODEx) that generate explanations through systematic input perturbation, and white-box methods (Grad-CAM, HiResCAM) that leverage internal model gradients and activations. The chapter then analyzes the YOLO object detection architecture, particularly YOLOv11's multi-scale prediction mechanism and single-cell detection paradigm, revealing architectural characteristics that critically impact XAI applicability.

Chapter 3 transitions from theory to practice through preliminary experiments. After preprocessing the EAGLE dataset and training YOLOv11l (achieving $\text{mAP}_{50} \approx 0.78$), we systematically apply and evaluate both black-box and white-box XAI methods. Through extensive qualitative analysis of true positives, false positives, and false negatives, we uncover critical limitations: black-box methods achieve faithful explanations but require thousands of forward passes per explanation, while white-box methods fail fundamentally due to sparse gradients and naive upsampling strategies that ignore true receptive fields. This reveals an unavoidable efficiency–faithfulness trade-off in existing approaches.

Chapter 4 presents our main contribution: **Receptive-Field-Based HiResCAM**, a novel XAI method designed specifically for object detectors. Rather than naively upsampling activation heatmaps, we explicitly reconstruct each grid cell's receptive field through input-image gradient computation, accounting for complex architectural elements including Feature Pyramid Networks and multi-scale detection heads. Comprehensive quantitative evaluation using insertion–deletion metrics and qualitative analysis demonstrates that our method successfully breaks the efficiency–faithfulness trade-off, achieving explanation quality matching black-box methods (scores: 0.589–0.868) with the computational efficiency of white-box approaches (single forward pass vs. thousands).

The **General Conclusion** synthesizes our findings, discusses practical implications for building trust-

worthy AI systems in aerial surveillance, and outlines future research directions including extensions to transformer-based detectors and integration with active learning frameworks.

Through this systematic investigation—from foundational understanding through comprehensive evaluation to novel method development—this work demonstrates that faithful, efficient explainability for aerial object detection is not only feasible but essential for deploying interpretable, accountable AI systems in mission-critical applications.

Chapter 1

General Overview

Introduction

This chapter establishes the project foundation by presenting: (1) the host institutions—the German Aerospace Center (DLR) and Digital Research Center of Sfax (CRNS)—and their research capabilities in aerospace and digital technologies; (2) the critical need for interpretability in safety-critical aerial object detection applications; and (3) the EAGLE dataset containing 215,986 annotated vehicles across 748 high-resolution aerial images, which serves as our primary experimental testbed.

1.1 Host Companies

This part introduces the two hosting institutions—the German Aerospace Center (DLR) and the Digital Research Center of Sfax (CRNS)—emphasizing their objectives, recent achievements, and contributions to the field of remote sensing research and technology.

1.1.1 German Aerospace Center (DLR)

Presentation

Established in 1969, the German Aerospace Center (DLR) functions as Germany's premier national research and technology hub for aeronautics, space, energy, transportation, security, and defense. Employing more than 11,000 staff across 30 sites, DLR plays a central role in Germany's aeronautics and space research landscape. The center comprises 54 research institutes and facilities, promoting scientific advancement and technological innovation to tackle societal, economic, and industrial challenges. Additionally, DLR carries out the German space program on behalf of the federal government through the German Space Agency.

services

The German Aerospace Center (DLR) provides a wide range of research and technological services in the following areas:

- **Aeronautics Research:** Development of innovative aircraft technologies, flight systems, and air traffic management solutions.
- **Space Exploration & Technology:** Designing and operating satellites, spacecraft, and instruments for scientific and commercial space missions.
- **Energy Research:** Advancing renewable energy technologies, energy efficiency, and sustainable energy systems.
- **Transport & Mobility:** Research on future transportation solutions, including intelligent traffic systems and sustainable mobility.

- **Security & Defense:** Development of technologies and strategies for civil and national security applications.
- **Remote Sensing & Earth Observation:** Monitoring and analyzing the Earth's environment and climate using satellite and airborne sensors.
- **Scientific & Technological Innovation:** Providing R&D support, prototypes, and solutions to address societal, industrial, and economic challenges.

1.1.2 Digital Research Center of Sfax (CRNS)

Presentation

The Digital Research Center of Sfax (CRNS) in Tunisia was founded in July 2013 as a public institution focused on the development of the information and communication technology (ICT) sector. The center is home to highly skilled senior researchers and PhD students specializing in computing and telecommunications, with a strong focus on applied research, technological innovation, and digital transformation. CRNS aims to promote research and development activities, facilitate innovation and technology transfer by leveraging research outcomes and expertise, and tackle key digital challenges in areas such as smart grids, e-health, smart agriculture, cybersecurity, and Industry 4.0.

services

The Digital Research Center of Sfax (CRNS) provides research and technological services in the following areas:

- **Applied Research & Development:** Conducting practical research in computing and telecommunications to solve real-world problems.
- **Technology Innovation:** Developing new digital solutions and innovative technologies for various sectors.
- **Digital Transformation Support:** Assisting industries and organizations in adopting and implementing digital technologies.
- **Technology Transfer & Valorization:** Translating research results into practical applications, products, and services.
- **Sector-Specific Solutions:**
  - **Smart Grids:** Optimizing energy distribution and management.
  - **E-Health:** Developing digital solutions for healthcare services.
  - **Smart Agriculture:** Implementing ICT for modern agricultural practices.
  - **Cybersecurity:** Researching and creating secure digital systems.
  - **Industry 4.0:** Supporting the digitalization and automation of industrial processes.

1.2 Problematic

Modern deep learning object detectors—especially YOLO-based architectures—achieve remarkable accuracy when identifying vehicles, buildings, and other objects in aerial imagery. However, these models function as opaque computational systems: they take an input image and produce bounding boxes with confidence scores through millions of non-linear transformations. As a result, practitioners cannot determine why a particular object was detected or why another was missed. This lack of transparency creates serious challenges in safety-critical domains, where validation, failure diagnosis, and trust are essential.
In disaster-response imagery, for example, analysts cannot easily verify whether a detected vehicle is supported by meaningful visual evidence or whether the model relied on spurious correlations. When systematic errors occur, the relevant visual cues or dataset biases remain hidden. These issues are

amplified in aerial imagery, which presents unique interpretability challenges compared to ground-level computer vision. Aerial objects are often tiny—sometimes only 20–100 pixels—meaning that subtle cues such as roof texture, shadow orientation, and contextual relationships may be decisive. The top-down viewpoint removes many familiar features, and objects appear in cluttered, diverse backgrounds where context may matter as much as appearance. Yet these detections influence critical decisions such as resource allocation in disasters, surveillance threat assessment, and autonomous navigation—situations where errors have real-world consequences.

Despite these needs, most explainable AI (XAI) methods were developed for image classification, not object detection, and have not been systematically evaluated for small-object detection in aerial imagery. Current object-detection XAI approaches fall into two categories, each with fundamental limitations that create an unavoidable trade-off between computational efficiency and faithfulness.

Black-box perturbation methods (e.g., D-RISE, D-CLOSE, and SODEx) mask regions of the input image and measure how occluding those regions affects detection confidence. These techniques can yield faithful explanations but require 1,000–5,000 forward passes per explanation. At this cost, explaining 10,000 detections would require tens of millions of model evaluations, making large-scale analysis, real-time use, or interactive debugging impractical.

White-box gradient methods (e.g., Grad-CAM and HiResCAM) use gradients and internal activations to produce explanations in a single forward and backward pass, enabling explanation times under 200 ms. However, these methods are fundamentally incompatible with modern object detectors. Naively upsampling activation-space heatmaps ignores the true receptive fields of feature-map cells, and sparse gradients from single-cell detections lead to spatially inaccurate visualizations. They also fail to account for architectural complexities such as Feature Pyramid Networks, multi-scale detection heads, and feature-fusion mechanisms found in detectors like YOLOv11.

This trade-off—faithful explanations that are computationally expensive versus efficient explanations that are spatially unreliable—limits the practical deployment of interpretable object-detection systems. No prior work has thoroughly evaluated both black-box and white-box methods specifically for small objects in aerial imagery, identified the architectural reasons gradient-based approaches fail with YOLO-family models, or proposed a principled solution that reconstructs true receptive fields rather than simply upsampling activation heatmaps.

This thesis addresses these gaps by investigating the following research questions:

1. How do existing XAI techniques perform when adapted to aerial object detection with YOLO architectures?
2. What architectural incompatibilities cause gradient-based methods to fail?
3. Can we design a new method that achieves black-box-level faithfulness with white-box-level efficiency by incorporating detector-specific architectural details?
4. Do the resulting explanations offer actionable insights for model improvement and failure analysis?

We answer these questions through extensive experiments on the EAGLE dataset and by introducing **Receptive-Field-Based HiResCAM**, a novel XAI method designed specifically for modern object detectors. Instead of naively upsampling activation heatmaps, our method explicitly reconstructs the true receptive field of each contributing grid cell using input-image gradient computation. It accounts for architectural elements such as Feature Pyramid Networks and multi-scale detection heads, enabling faithful, high-resolution explanations while preserving single-pass computational efficiency. In doing so, it breaks the long-standing efficiency–faithfulness trade-off and advances the practical interpretability of object detection in safety-critical aerial applications.

1.3 Dataset

In this project,we used the EAGLE (Earth observation data for object detection in aerial images) dataset, published by the German Aerospace Center (DLR) consists of 748 aerial image with size of 5616×3744 px acquired during several flight campaigns carried out between 2006 and 2019 in various time of day and year with different weather and illumination conditions [2]

The EAGLE contains 215, 986 annotated vehicles, ranging from 1 to 3,567 annotations per image in all possible orientations. Each object instance is annotated with detailed information, including bounding boxes, visibility conditions (totally, partly, or hardly visible), and orientation clarity (clear or unclear) , figure 1.1 shows a few samples from the EAGLE dataset.

Figure 1.1: Image samples from EAGLE dataset with ground sampling truth from 5 cm to 45 cm per pixel and size 5616 $\times$ 3744 px

Conclustion

This chapter established the institutional framework, research motivation, and data foundation. The EAGLE dataset's diversity in lighting conditions, object scales, and spatial resolutions (5-45 cm per pixel) provides an ideal testbed for investigating XAI techniques in challenging small-object detection scenarios. With this foundation established, Chapter 2 surveys existing XAI techniques and object detection architectures that form the technical basis for our experimental investigations.

Chapter 2

State Of The Art

Introduction

In this chapter,we examine black-box XAI methods (D-RISE, D-CLOSE, SODEx) that generate explanations through input perturbation, and white-box methods (Grad-CAM, HiResCAM) that leverage internal gradients and activations. We then analyze the YOLO family, particularly YOLOv11's architecture, multi-scale detection heads, and single-cell prediction mechanism—design choices that critically impact XAI applicability.

2.1 X-AI Techniques

2.1.1 blackbox X-AI Techniques

Black-box explainable AI (XAI) techniques aim to interpret the behavior of machine-learning models without requiring access to their internal architecture or parameters. Instead of analyzing how the model is constructed, these methods treat the model as a black box: inputs are perturbed and the resulting changes in the outputs are examined. By observing these input–output relationships, black-box techniques infer which features influence predictions and to what extent.
The core principle behind these approaches is *local sensitivity analysis*. By systematically altering specific input components either individually or in combinations and measuring the magnitude and direction of the corresponding prediction shift, these methods approximate the contribution of each feature. This provides transparent, model-agnostic explanations even for highly complex models such as deep neural networks or ensemble learners.
Common black-box XAI methods include LIME, model-agnostic SHAP, counterfactual explanations, D-RISE ,D-CLOSE..
Together, these techniques offer intuitive insights into which factors drive model decisions, without exposing or relying on the model's internal design.

2.1.1.1 D-RISE

D-RISE [8] is a black-box explainable AI (XAI) method designed specifically to generate visual explanations for object detectors. It extends the input-perturbation strategy of the RISE method, which was originally developed for image classifiers, and adapts it to the more complex output structure of object detection models.
Similar to RISE, D-RISE operates by applying random perturbation masks to the input image. Each mask partially occludes different regions of the image, and the perturbed image is then passed through the object detector. Rather than relying on internal model information, the method observes how these structured perturbations influence the detector's predictions. This enables a model-agnostic explanation process that depends solely on input–output behavior.
A key challenge with explaining object detectors is that their outputs are not single class probabilities but sets of bounding boxes with associated confidence scores. To address this, D-RISE focuses on a *specific target bounding box* that we want to explain. For each perturbed image, the method

measures the change in the detector's output with respect to this target bounding box using an appropriate similarity metric . Masks that preserve the target object's detectability (high similarity) receive higher weights, while those that significantly reduce the score or localization accuracy receive lower weights(low similarity) . This is derived by the idea that if important regions are masked , the similarity between the original bounding box and the detected bounding box (after hiding those regions) is low .
By aggregating the contributions of all perturbation masks, D-RISE produces a saliency map that highlights the regions of the image most influential for detecting the chosen bounding box. As a result, it provides intuitive, spatially grounded explanations for complex, high-dimensional object detection models, while maintaining full model-agnosticism . The figure 2.2 shows the whole pipeline of D-RISE.

Random Masking

To estimate pixel-level importance in a black-box setting, the RISE method uses randomized masking and evaluates how the model's confidence score changes when different subsets of pixels are visible.
Let $f : I \to \mathbb{R}$ be a black-box model producing a scalar score for an input image I. We define the image space as

$$I = \{I \mid I : \Lambda \to \mathbb{R}^3\},$$

where $\Lambda = \{1, \ldots, H\} \times \{1, \ldots, W\}$ is the set of pixel coordinates and each $I(\lambda)$ contains RGB values. The model f may be a classifier, detector, or captioning model providing a confidence score for the masked input.
Let $M : \Lambda \to \{0, 1\}$ be a random binary mask sampled from a distribution $\mathcal{D}$. The masked input is obtained by an element-wise multiplication $I \odot M$. The importance of a pixel λ is defined as the expected model score over all masks that keep this pixel visible:

$$S_{I,f}(\lambda) = \mathbb{E}_M\big[\, f(I \odot M) \mid M(\lambda) = 1 \,\big]. \tag{2.1}$$

Expanding the conditional expectation over all possible masks $m : \Lambda \to \{0, 1\}$:

$$S_{I,f}(\lambda) = \sum_m f(I \odot m)\, P[M = m \mid M(\lambda) = 1] \tag{2.2}$$

$$= \frac{1}{P[M(\lambda) = 1]} \sum_m f(I \odot m)\, P[M = m, M(\lambda) = 1]. \tag{2.3}$$

Since

$$P[M = m, M(\lambda) = 1] = \begin{cases} 0, & m(\lambda) = 0, \\ P[M = m], & m(\lambda) = 1, \end{cases}$$

we can write

$$S_{I,f}(\lambda) = \frac{1}{P[M(\lambda) = 1]} \sum_m f(I \odot m)\, m(\lambda)\, P[M = m]. \tag{2.4}$$

Using the fact that $P[M(\lambda) = 1] = \mathbb{E}[M(\lambda)]$, this becomes:

$$S_{I,f} = \frac{1}{\mathbb{E}[M]} \sum_m f(I \odot m)\, m\, P[M = m]. \tag{2.5}$$

Equation 2.5 shows that the saliency map is a weighted sum of random masks, where each mask is weighted by the model's confidence score on the corresponding masked image.
Monte Carlo Approximation: Since enumerating all possible masks is infeasible, RISE uses Monte Carlo sampling. Given N masks $\{M_1, \ldots, M_N\}$ drawn from $\mathcal{D}$, we approximate the importance map as:

$$S^{\mathrm{MC}}_{I,f}(\lambda) \approx \frac{1}{N\, \mathbb{E}[M]} \sum_{i=1}^{N} f(I \odot M_i)\, M_i(\lambda). \tag{2.6}$$

Thus, the method produces a black-box saliency map by averaging random masks, weighted by the model scores they produce. This requires no access to model parameters, gradients, or internal features, making it suitable for explaining arbitrary black-box vision models.

Mask Generation

A naïve approach to random masking would set each pixel of the mask independently to 0 or 1. However, this leads to two main problems. First, small independent pixel changes can introduce *adversarial effects*, causing drastic and unstable changes in the model's confidence scores. Second, independently sampling each pixel creates an enormous mask space of size $2^{H \times W}$, which requires an impractically large number of samples to obtain a reliable Monte Carlo estimate of the saliency map. To overcome these issues, RISE (same as D-RISE in the mask generation process) generates *structured, low-resolution* random masks and then upscales them smoothly to the image size. This approach reduces the combinatorial mask space, decreases sampling variance, and avoids sharp edges that may cause adversarial behavior.
The mask generation process consists of three steps:

1. **Sampling low-resolution binary masks:** Generate N binary masks of size $h \times w$, where $h \ll H$ and $w \ll W$. Each element is independently set to 1 with probability p, and to 0 otherwise. Sampling at this coarse resolution significantly reduces the number of possible masks.

2. **Upsampling using bilinear interpolation:** Each low-resolution mask is upsampled to approximately the image size using bilinear interpolation. This smooths the masks, avoiding sharp edges and adversarial artifacts, and produces continuous mask values in $[0, 1]$. The upsampled mask initially has resolution $(h+1)C_H \times (w+1)C_W$, where $C_H = \lfloor H/h \rfloor$ and $C_W = \lfloor W/w \rfloor$ are the cell sizes in the mask.

3. **Random spatial shifts:** To increase variability and reduce grid-aligned artifacts, each upsampled mask is randomly shifted by up to (C_H, C_W) pixels in both spatial directions. Finally, a crop of size $H \times W$ is extracted to match the original image.

Through this three-step process, RISE produces diverse yet smooth masks that enable stable, model-agnostic explanations by probing the black-box model with structured perturbations.
Finally, The technique multiplies the final masks with the original image (element-wise multiplication) to get the original image hidden with many masks , figure 2.1 shows a few samples from different masks.

Figure 2.1: Image masked with different masks using RISE algorithm

Similarity metric

To quantify how closely a detection proposal aligns with the target bounding box, D-RISE defines a similarity metric that considers three key aspects of the detection vectors: spatial overlap, class prediction similarity, and objectness confidence. This metric ensures that only proposals that are both spatially and semantically similar to the target contribute significantly to the explanation.

- **Spatial similarity:** The Intersection over Union (IoU) is used to measure the overlap between the bounding boxes of the target vector d_t and a proposal d_j:

$$s_L(d_t, d_j) = \text{IoU}(L_t, L_j), \tag{2.7}$$

  where L_t and L_j denote the bounding box coordinates of the target and proposal, respectively.

- **Class probability similarity:** To capture how similar the proposal region looks to the network, the cosine similarity between the class probability vectors P_t and P_j is computed:

$$s_P(d_t, d_j) = \frac{P_t \cdot P_j}{\|P_t\| \, \|P_j\|}. \tag{2.8}$$

- **Objectness similarity:** For networks that produce an objectness score (e.g., YOLOv3), the similarity also incorporates the objectness score O_j of the proposal. When explaining high-confidence detections, the target objectness O_t is set to 1, so the contribution is simply

$$s_O(d_t, d_j) = O_j. \tag{2.9}$$

  For networks without an objectness score (e.g., Faster R-CNN), this term can be omitted.

The overall similarity between a target and a proposal is computed as the product of these three components:

$$s(d_t, d_j) = s_L(d_t, d_j) \cdot s_P(d_t, d_j) \cdot s_O(d_t, d_j). \tag{2.10}$$

Using the product ensures a logical "AND" behavior: if any component is low (e.g., poor spatial overlap or dissimilar class probabilities), the overall similarity decreases accordingly. This metric allows D-RISE to accurately weigh proposals when aggregating their contributions to the explanation of the target detection.

Detector Inference

The final saliency map for each target detection is obtained by aggregating the contributions of masked inputs weighted by their similarity to the target. The process can be summarized in the following steps:

1. **Generate RISE masks:** Create N random masks $M = \{M_i \mid 1 \leq i \leq N\}$ using the structured random masking procedure.

2. **Prepare target detections:** Convert the set of target detections to be explained into detection vectors $D_t = \{d_t \mid 1 \leq t \leq T\}$. The detector only needs to be run once on each masked image to generate explanations for all T target detections simultaneously.

3. **Run the detector on masked images:** For each masked image $I \odot M_i$, run the object detector f to produce N_p proposals, $D_i^p = \{d_j^i \mid 1 \leq j \leq N_p\}$.

4. **Compute similarity weights:** For each target detection d_t and each masked image M_i, compute the maximum similarity between d_t and all proposals in D_i^p using the similarity metric s:

$$w_i^t = \max_{1 \leq j \leq N_p} s(d_t, d_j^i), \quad 1 \leq i \leq N, \, 1 \leq t \leq T. \tag{2.11}$$

5. **Aggregate saliency maps:** Compute the saliency map H_t for each target detection d_t as a weighted sum of all masks:

$$H_t = \sum_{i=1}^{N} w_i^t \, M_i. \tag{2.12}$$

All operations, including the similarity computations and weighted aggregation, can be efficiently implemented using vectorized tensor operations, such as element-wise multiplication, maximum along an axis, and weighted summation along an axis. The resulting saliency maps highlight the regions of the input image that most strongly influence the detection of each target object, providing intuitive, spatially grounded explanations for black-box object detectors.

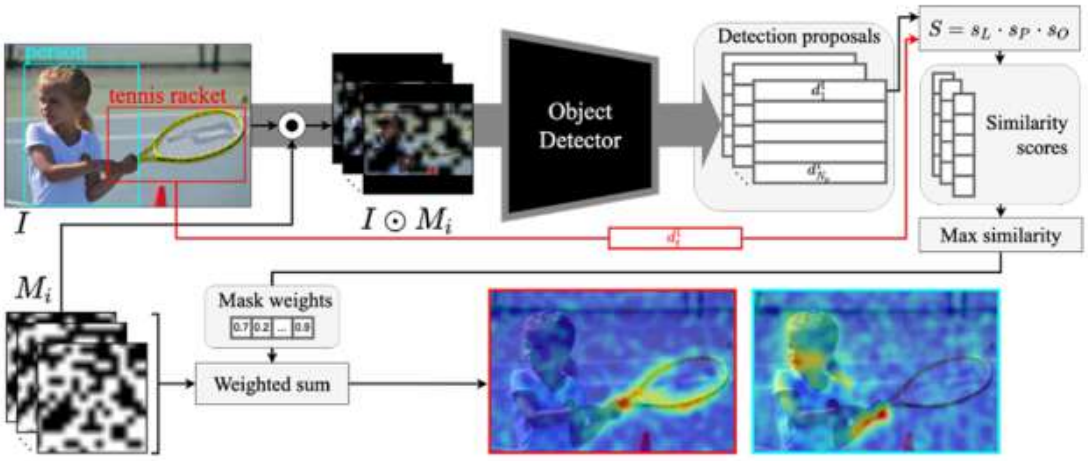


Figure 2.2: D-RISE Pipeline

2.1.1.2 D-CLOSE

Detector-Cascading Multiple Levels of Segments to Explain (D-CLOSE) is a perturbation-based explainability method designed for object detection models, including both one-stage and two-stage detectors. Its goal is to generate faithful saliency maps that highlight the image regions contributing to individual detection results.

The main difference between D-CLOSE and D-RISE lies in the mask generation process. While D-RISE extends the original RISE framework by sampling purely random binary masks, D-CLOSE extends the Morphological Fragmental Perturbation Pyramid (MFPP) strategy. Specifically, the input image is segmented into multiple levels of superpixels using SLIC with different granularities. At each level, random perturbation masks are generated by probabilistically activating or deactivating superpixel regions. This multi-scale perturbation design enables D-CLOSE to capture both fine-grained object features and broader contextual information.

Each masked image is subsequently forwarded through the object detector, and the resulting predictions are compared with the target detection using the same similarity metric as D-RISE. This metric combines the Intersection-over-Union (IoU) between predicted and target bounding boxes, the objectness score, and the cosine similarity between class confidence vectors. The resulting similarity score is used as a weight for the corresponding perturbation mask.

To address the non-uniform spatial distribution introduced by random mask sampling, D-CLOSE computes a density map that normalizes the contribution of each pixel, leading to smoother and less noisy explanations. A saliency map is then generated for each superpixel level. Finally, these multi-level saliency maps are fused in a cascading manner, progressively combining detailed feature maps with more general semantic and contextual information. This fusion process does not rely on internal network activations or threshold-based filtering.

Overall, D-CLOSE provides a model-agnostic and multi-scale explanation framework for object detection that preserves the robust similarity scoring of D-RISE while significantly enhancing the spatial structure of the generated explanations through MFPP-based perturbations and hierarchical feature fusion , figure 2.5 provides the full architecture of D-CLOSE.

Mask Generation

D-CLOSE inherent from MFPP algorithm to generate perturbation masks using multi-level superpixel segmentation based on Simple Linear Iterative Clustering (SLIC). The input image is segmented into L different levels, where each level corresponds to an increasing number of superpixels, ranging from coarse to fine granularity. Lower levels consist of fewer, larger segments that capture global structure

and coarse object regions, while higher levels contain more, smaller segments that preserve fine object details.
For each segmentation level, random binary masks are generated by sampling the superpixel regions. Specifically, each superpixel is independently activated with a fixed probability and deactivated otherwise, producing a set of random perturbation masks. These masks are then resized using bilinear interpolation and randomly cropped to match the original image resolution. The multi-level design allows the perturbation process to capture both large contextual regions and fine-grained object features, which is essential for constructing accurate multi-scale saliency maps.

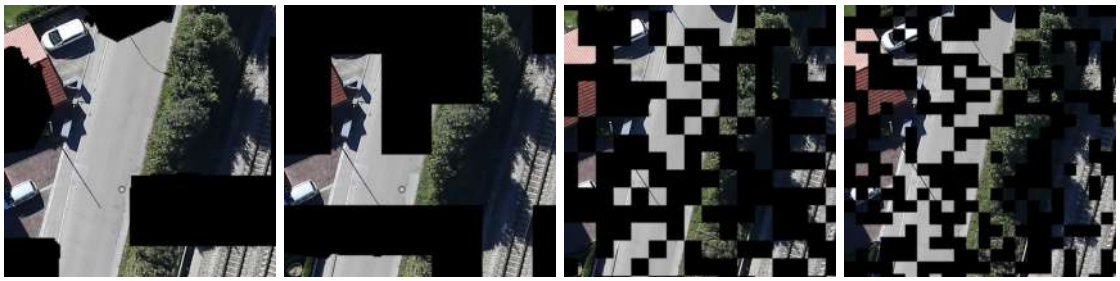


Figure 2.3: Image masked with different levels using MFPP algorithm

Detector Inference and Saliency Map per Level

Figure 2.4 illustrates the cascading fusion process used in D-CLOSE to infer the final saliency map from multiple segmentation levels. After generating saliency maps at different superpixel levels (from coarse to fine), each level produces a feature map with a different semantic resolution.
The fusion process starts by normalizing each saliency map to the range $[0, 1]$. The normalized maps are then combined in a hierarchical manner. At each cascade stage, two adjacent saliency maps are first added together to aggregate complementary information. The result is then multiplied point-wise with the saliency map from the finer level. This point-wise multiplication acts as an attention mechanism that emphasizes regions consistently highlighted across multiple scales.

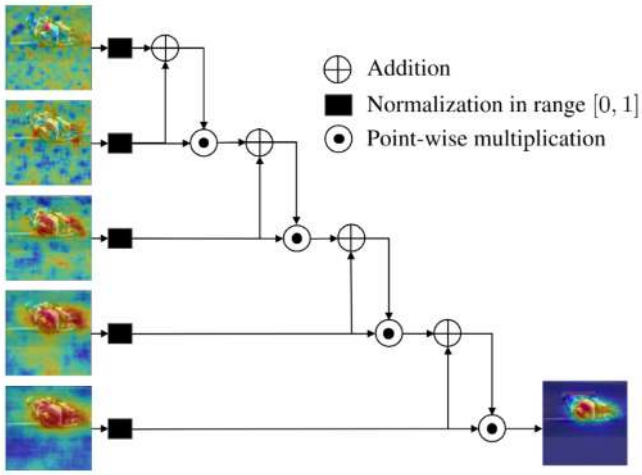


Figure 2.4: Cascading Fusion Process

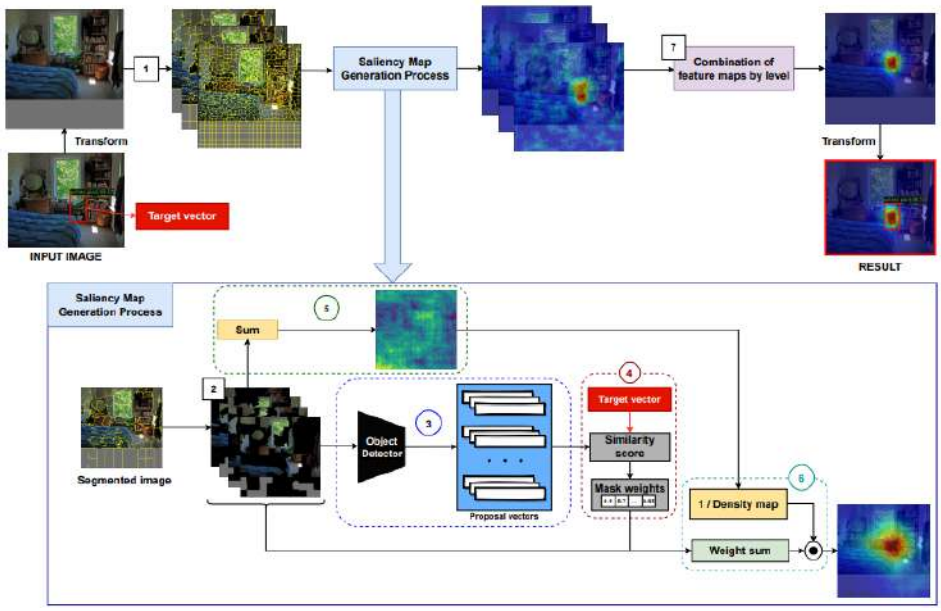


Figure 2.5: D-CLOSE Pipeline

2.1.1.3 SODEx

SODEx (Surrogate Object Detector Explainer) is an explainability technique for object detectors that leverages **surrogate classifiers** to make standard classifier-based XAI methods applicable. Instead of directly explaining the complex detector, SODEx transforms the detection problem into a classification problem for a specific detection of interest.
SodeX is an explanation framework that transforms the object detection explanation problem into a classification explanation problem through the use of a surrogate model. Instead of directly explaining the object detector, SodeX replaces it with a surrogate classifier that mimics the behavior of the detector for a specific object under explanation.
Given an input image, the surrogate model first applies a YOLO-based object detector to obtain all predicted bounding boxes and their corresponding confidence scores. Among these detections, the surrogate searches for the bounding box that has the highest Intersection-over-Union (IoU) with the target object to be explained, provided that the IoU exceeds a predefined threshold. The confidence score associated with this selected bounding box is then used as the output of the surrogate model.
Through this process, the surrogate model effectively maps an input image to a single probability value, similarly to a standard image classifier. This transformation enables the direct application of any explainable AI (XAI) method originally designed for image classification, (usually LIME) , to explain object detection predictions.
By decoupling the explanation process from the internal structure of the object detector, SodeX provides a flexible, model-agnostic framework that allows existing classification-based XAI (in this project we've used LIME explainer) techniques to be reused for object detection tasks. in the figure 2.6 , the full algorithm is explained.

LIME

LIME (Local Interpretable Model-agnostic Explanations) is an explainability technique used to interpret the predictions of black-box models, such as image classifiers. The version of LIME applied to images relies on **SLIC segmentation** to divide an input image into multiple superpixels.
To generate explanations, LIME creates many perturbed samples of the original image by randomly hiding some of the superpixels. Each perturbed image is then passed through the classifier to obtain a prediction. This process results in a dataset consisting of multiple images with different combinations of visible superpixels along with their corresponding predicted outputs.
SODEX then trains a simple, interpretable linear model on this dataset. The weights of the linear model indicate the contribution of each superpixel to the classifier's prediction. By analyzing these

weights, one can identify the most important superpixels, thus successfully explaining the decision of the image classifier.

Bounding Box Selection and Confidence Threshold

For the bounding box under explanation in the original image (bounding box b) , SODEx finds the predicted bounding box b' in each perturbed image that has the **maximum Intersection over Union (IoU)** with b. Then, the confidence score of this matched bounding box is used as the output of the surrogate classifier. Finally, Only boxes with confidence above a threshold τ are considered; detections below this threshold are ignored to avoid noisy explanations.

Surrogate Classifier and Saliency Map

1. The object detector is effectively replaced by a **binary classifier** that outputs the confidence of the detection being present or absent.
2. Any standard classifier XAI technique (e.g., LIME, SHAP, Integrated Gradients) can then be applied to this surrogate classifier , in our project only LIME technique is used.
3. The resulting saliency map highlights the image regions that most influence the detection of interest.

Algorithm 1 Surrogate Binary Classifier (SBC)

```
function SBC_oue(I)                                   ▷ Object under explanation (oue)
    objects ← YOLO.FIND_OBJECTS(I)
    if objects is empty then
        return 0
    end if
    iou_max ← −1
    c_score ← 0
    for all object ∈ objects do
        iou ← IOU(object.bbox, oue.bbox)
        if iou > IOU_MIN ∧ iou > iou_max ∧ object.class = oue.class then
            iou_max ← iou
            c_score ← object.score
        end if
    end for
    return c_score
end function
```

Algorithm 2 Surrogate Object Detection Explainer (SODEx)

```
function SODEx(obj)
    seg_alg ← QUICKSHIFT                          ▷ or another segmentation algorithm
    classifier ← SBC_obj
    explanation ← LIME.EXPLAIN(classifier, seg_alg, obj)
end function
```

Figure 2.6: SODEx Algorithm

2.1.2 White Box X-AI

White-box explainable AI (XAI) methods interpret a model's behavior by directly examining its internal architecture such as its layers, activations, and gradients. Unlike black-box techniques, which rely solely on input perturbations and observe how predictions change, white-box methods "open up" the model and analyze the mechanisms that drive its decision-making. Examples of such approaches include gradient-based visualization tools like Grad-CAM and HiResCAM, which highlight the regions of the input that most strongly influence the model's output.

2.1.2.1 GradCAM

Gradient-weighted Class Activation Mapping (Grad-CAM) is a white-box XAI technique that visualizes the regions of an input image that most strongly influence a model's prediction. It accomplishes this by examining the gradients flowing into the last convolutional layer of a convolutional neural network (CNN), thereby requiring access to the model's internal architecture , figure2.7 shows the full pipeline of GradCAM. Grad-CAM operates through the following steps:

1. **Forward pass through the network** The input image is fed through the model, and the activations of the last convolutional layer are recorded. Let these feature maps be denoted by A^k, where k indexes each channel. Convolutional layers are chosen because they preserve spatial information that is lost in deeper fully connected layers.

2. **Compute the gradient of the target class score** For a target class c, the gradient of the score y^c with respect to the feature maps A^k is computed:

$$\frac{\partial y^c}{\partial A^k} \tag{2.13}$$

   These gradients capture how sensitive the class score is to each activation.

3. **Global average pooling of gradients** The gradients are spatially averaged to obtain a single importance weight for each channel:

$$\alpha_k^c = \frac{1}{Z} \sum_i \sum_j \frac{\partial y^c}{\partial A_{ij}^k}, \tag{2.14}$$

   where Z is the number of spatial locations (height $\times$ width). The weights α_k^c represent the contribution of each feature map to class c.

4. **Weighted combination of feature maps** The class-discriminative localization map is obtained by forming a weighted linear combination of the feature maps:

$$L_{\text{Grad-CAM}}^c = \text{ReLU}\left(\sum_k \alpha_k^c A^k\right) \tag{2.15}$$

   The ReLU operation preserves only the features that positively influence the class score.

5. **Upsampling to input resolution** The resulting heatmap is low resolution due to the spatial size of the convolutional layer. It is therefore upsampled (e.g., via bilinear interpolation) to match the size of the input image and is typically overlaid on the original image to provide a visual explanation.

Grad-CAM thus highlights the regions of the image that the model relies on when predicting a specific class. For example, when identifying a "cat" in an image, Grad-CAM often emphasizes the cat's face or other distinctive features. This makes it a valuable tool for understanding and validating the behavior of deep neural networks.

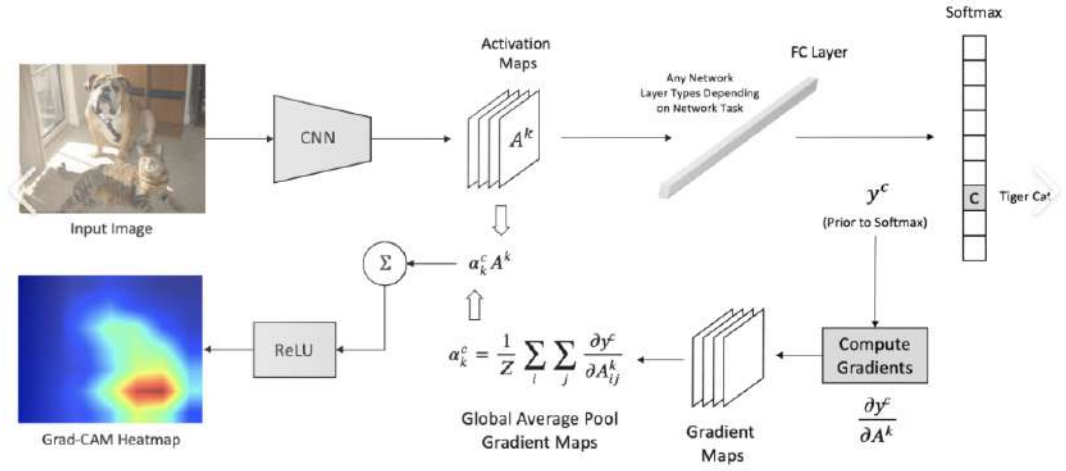


Figure 2.7: GradCAM pipeline

2.1.2.2 HiresCAM

High-Resolution Class Activation Mapping (HiResCAM) is a white-box XAI technique designed to produce higher-resolution and more faithful saliency maps compared to Grad-CAM. HiResCAM addresses limitations of Grad-CAM, such as low spatial resolution and reliance on channel-wise gradient averaging, by computing pixel-wise contributions of activations to the class score . Figure 2.8 shows the difference in weight's calculation between gradCAM and HiresCAM. HiResCAM operates through the following steps:

1. **Forward pass to obtain feature maps**
   The input image is passed through the model, and the feature maps A^k of a target convolutional layer are recorded. Convolutional layers are used due to their spatial structure, which preserves location-specific information.

2. **Compute gradients of the class score**
   For the target class c, compute the gradients of the class score with respect to each spatial activation:

$$G_{ij}^k = \frac{\partial y^c}{\partial A_{ij}^k}. \tag{2.16}$$

   Unlike Grad-CAM, HiResCAM does not perform global average pooling on these gradients, retaining fine spatial details.

3. **Pointwise multiplication**
   Compute the element-wise product of activations and their corresponding gradients:

$$M_{ij}^k = A_{ij}^k \cdot G_{ij}^k. \tag{2.17}$$

   This step preserves spatial resolution and reflects the contribution of each activation to the class score.

4. **Aggregation across channels with ReLU**
   The class-specific relevance map is obtained by summing over all channels, followed by a ReLU:

$$L_{\text{HiResCAM}}^c(i,j) = \text{ReLU}\left(\sum_k M_{ij}^k\right). \tag{2.18}$$

5. **Normalization and upsampling**
   The resulting relevance map can be normalized and upsampled to match the input image dimensions for visualization. Due to its pixel-wise computation, HiResCAM naturally provides higher spatial resolution than Grad-CAM.

HiResCAM produces fine-grained saliency maps that highlight exactly which regions of the image contributed to the model's prediction. For example, when predicting the class "cat", HiResCAM may highlight fur textures, contours of ears, and subtle edges, offering a more precise explanation than Grad-CAM. This makes HiResCAM especially valuable for applications where detailed spatial understanding is critical.

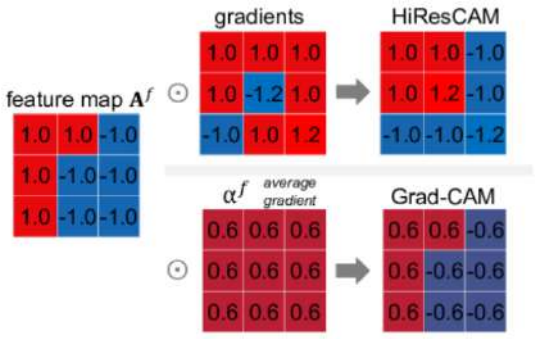


Figure 2.8: Difference in Weights calculation between GradCAM and HiresCAM

2.2 Object-Detector

2.2.1 YOLO

You Only Look Once (YOLO) is a family of real-time object detection algorithms widely used in computer vision applications such as autonomous driving, surveillance, and robotics. YOLO formulates object detection as a single-stage regression problem: the input image is fed into the backbone to produce feature maps , each spatial (x,y) grid cell in the feature maps predicts bounding boxes along with class probabilities. Because all predictions are generated in a single forward pass through the network, YOLO achieves high detection speed while maintaining competitive accuracy.
Over the years, several versions of YOLO have been developed to improve accuracy, robustness, and efficiency. Models from YOLOv1 to YOLOv3 introduced the core architecture and multi-scale feature prediction. YOLOv4 incorporated additional training and architectural optimizations such as CSPDarknet53, Mosaic augmentation, and advanced data augmentation techniques. YOLOv5, implemented in PyTorch, improved usability and modularity while offering multiple model sizes. More recent iterations, including YOLOv6, YOLOv7, and YOLOv8, introduced further improvements in speed, optimization methods, and task-specific variants.
Most modern YOLO versions (such as YOLOv8 and YOLOv11) are released in multiple scaled variants, enabling users to choose a suitable model depending on available hardware and performance requirements. These commonly include:

- **YOLO-n (Nano)**: Extremely lightweight and fast; designed for mobile and edge devices.
- **YOLO-s (Small)**: Small model with a good balance between speed and accuracy; suitable for embedded GPUs.
- **YOLO-m (Medium)**: Balanced accuracy and computational cost; useful for general-purpose applications.
- **YOLO-l (Large)**: Higher accuracy at the cost of increased computational demand; ideal for high-performance GPUs.

- **YOLO-x (Extra-Large)**: Largest and most accurate variant; often used for offline processing or research applications.

These scaled versions offer flexibility, allowing developers to choose the appropriate trade-off between computational efficiency and detection accuracy for a given application.

2.2.2 YOLOv11

YOLOv11 is designed for real-time detection with high accuracy. The model processes an input image in a single forward pass, predicting objects of different sizes efficiently by leveraging advanced feature extraction, multi-scale detection, and feature fusion mechanisms

2.2.2.1 pre-processing

YOLOv11 is designed for **real-time detection** with high accuracy. The model processes an input image in a single forward pass, predicting objects of different sizes efficiently by leveraging advanced feature extraction, multi-scale detection, and feature fusion mechanisms.
When an image is fed into YOLOv11, it first passes through the **backbone network**, which extracts hierarchical **feature maps** containing spatial and semantic information. Each **cell in these feature maps** (the output of the backbone network) is responsible for predicting objects located in its **receptive field** which means the region of the input image that influences that cell. A 1×1 convolution is applied to each cell to produce a vector of length $k = 4 + \text{num\_of\_classes}$, where:

- 4 represents the bounding box coordinates (x_c, y_c, w, h)
- num_of_classes represents class probabilities $(p_{c1}, p_{c2}, \dots)$

in the figure 2.9 , the feature maps extracted by the backbone are operated with 1*1*k convolution producing a matrix of detections , each d_i is a vector of length= k with k = 4 (xc,yc ,w,h) + num of classes $(p_{c_1}, p_{c_2}, \dots, p_{c_n})$. The final matrix of detections is 3d matrix with shape (H,W,k) , H and W is the spatial dimension of the feature maps produced by the backbone and k is the number of the target outputs (x_c,y_c,h,w,C_1...C_n).
To handle objects of varying sizes, YOLOv11 employs **multi-scale detection heads**,This multi-scale approach ensures that small objects, which may be lost in coarser feature maps, are captured accurately, while large objects are detected reliably in lower-resolution maps. it basically consists of extracting features from three different resolutions of the backbone outputs, in other words , the backbone outputs 3 features maps with different spatial dimensions (H,W) :

- $(H, W) = (80, 80)$ for small objects $\rightarrow$ number of detections: $80 \times 80 = 6400$: last layer is 16 dedicated for small objects.
- $(H, W) = (40, 40)$ for medium objects $\rightarrow$ number of detections: $40 \times 40 = 1600$:last layer is 19 dedicated for medium size objects.
- $(H, W) = (20, 20)$ for large objects $\rightarrow$ number of detections: $20 \times 20 = 400$:last layer is 22 dedicated for big objects.

This results in 8400=6400+1600+400 detections in one image.As mentioned before , a detection is a vector with length = 4 (xc,yc ,w,h) + num of classes $(p_{c_1}, p_{c_2}, \dots, p_{c_n})$. , resulting in so many overlapping bounding boxes with different confidence scores , so here comes the importance of **post-processing** to let only the most accurate and appropriate bounding boxes as the final detections.
YOLOv11 also integrates advanced feature fusion mechanisms in its backbone and neck parts . Figure 2.10 shows the full architecture of yolov11:

- **Feature Pyramid Network (FPN):** Combines features from multiple backbone layers, merging low-level spatial details with high-level semantic information.
- **SPFF (Spatial Pyramid Fusion Feature):** Enhances multi-scale feature representation by fusing features in a structured manner, improving detection in complex scenes.
- **C3K2:** Module used to capture objects of different scales.

These architectural enhancements make YOLOv11 highly effective in detecting objects across a range of sizes and challenging conditions.

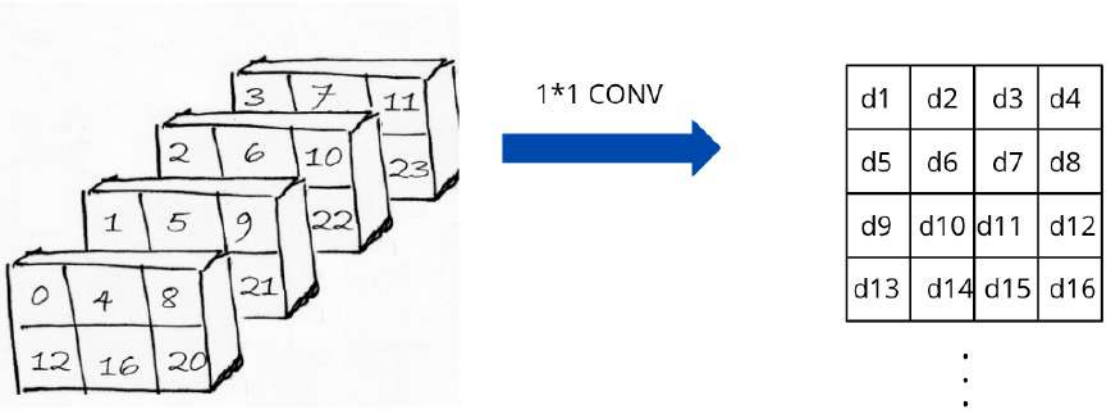


Figure 2.9: Producing detections with $1 \times 1 \times k$ convolution applied to the produced feature maps

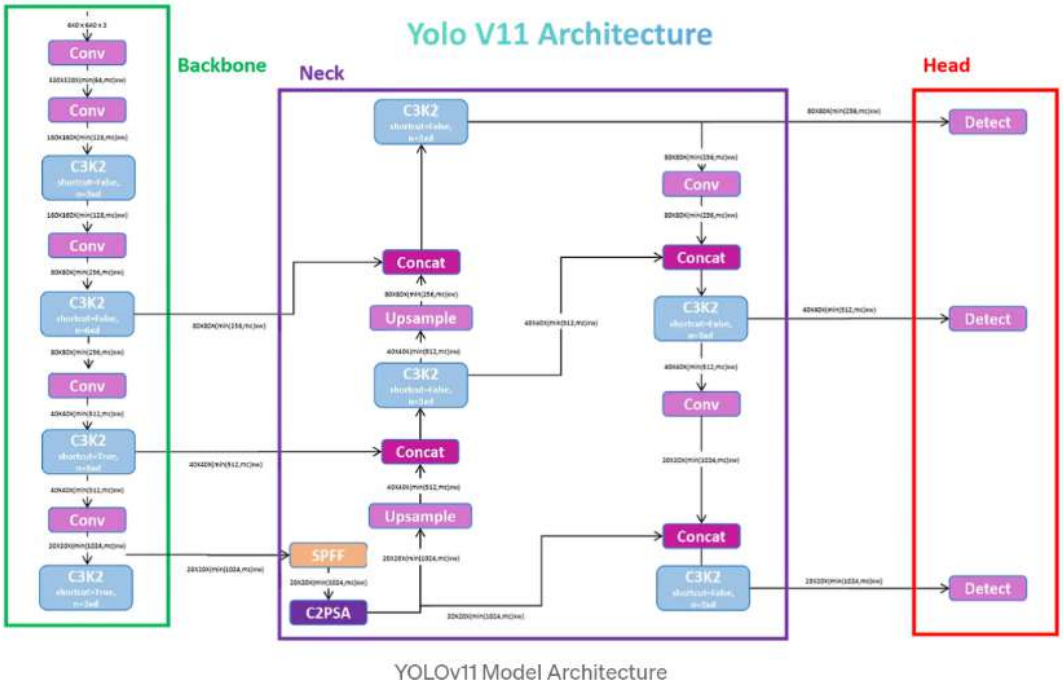


Figure 2.10: YOLOv11 full architecture in pre-process (before NMS, thresholding)

2.2.2.2 Post-Processing: Confidence Thresholding and Non-Maximum Suppression

After generating raw predictions, YOLOv11 applies **post-processing** to refine the results and remove redundant or low-confidence detections.

1. **Confidence Thresholding:** Predictions with confidence scores below a certain threshold are discarded. This ensures that only predictions likely to correspond to real objects are considered.

2. **Non-Maximum Suppression (NMS):** Among overlapping bounding boxes for the same object. If the IOU(bounding-box1,bounding-box2)>0,45 the model considers that these two bounding boxes target the same object , in this case only the box with the highest confidence is kept

and the others are suppressed . This eliminates duplicate detections and provides a clean set of final bounding boxes.

By combining confidence thresholding and NMS, YOLOv11 produces accurate and reliable final detections during inference, ready for practical applications such as surveillance, autonomous driving, and industrial inspection.

conclusion

This review revealed a critical gap: most XAI techniques were designed for image classification, not object detection. Object detectors produce thousands of bounding boxes with spatial parameters rather than single class probabilities—a fundamentally different output structure. YOLOv11's architecture, which generates 8,400 initial predictions from single grid cells across three scales (80×80, 40×40, 20×20), poses unique challenges for gradient-based explanations. These architectural incompatibilities motivate our experimental investigation in Chapter 3, where we systematically evaluate how existing methods perform on aerial vehicle detection.

Chapter 3

Preliminary Experiments

Introduction

This chapter applies existing XAI techniques to YOLOv11 trained on the EAGLE dataset. After preprocessing the data into 640×640 patches and training YOLOv11l (achieving $\text{mAP}_{50} \approx 0.78$), we systematically test black-box methods (D-RISE, D-CLOSE, SODEx) and white-box methods (Grad-CAM, HiResCAM) after adapting them for object detection. Through extensive qualitative analysis of true positives, false positives, and false negatives, we identify critical limitations that reveal fundamental trade-offs between computational efficiency and explanation faithfulness.

3.1 Data Pre-processing

In this project we used YOLO [4] model for aerial object detection, chosen for its simplicity and efficiency, the original EAGLE dataset annotations were converted into the YOLO format. This format represents each object with its class label and normalized bounding box parameters (class,x_c,y_c,w,h). During this transformation, additional informations provided by EAGLE dataset such as bounding box orientation, visibility condition, and orientation clarity were discarded, as YOLO does not process these attributes.

To prepare the imagery for training, each high-resolution image was cropped into multiple patches of size 640×640 pixels, matching the input size of the YOLO model. A 10% overlap between patches was applied to both augment the dataset and ensure that objects located near patch borders were fully captured in at least one crop. This approach preserved maximum spatial detail while adapting the data to the model's requirements, figure 3.1 shows some patched samples.

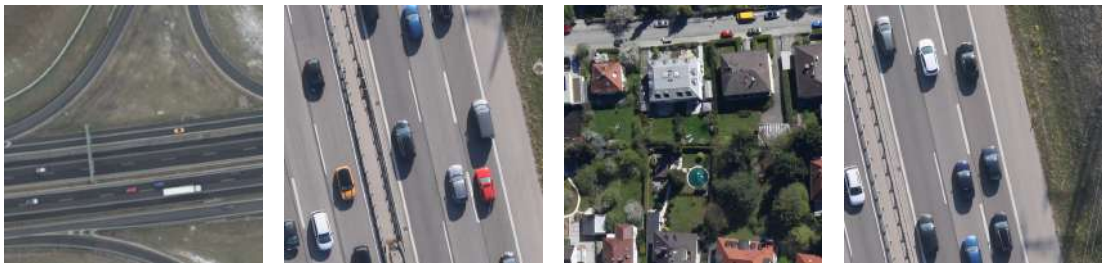


Figure 3.1: Patched Cropped from eagle dataset with width=height = 640 and 5% overlap

3.2 Model Training

For the object detection task, the YOLOv11l (large) is selected for its higher accuracy and high performance, and is trained on the patched version of the EAGLE dataset, there are two classes (small

vehicles , big vehicles).
The preprocessing stage ensured that all images were cropped to 640×640 pixels, matching the model's expected input dimensions. Training was conducted for 100 epochs, allowing the model to learn robust feature representations from the aerial imagery , figure 3.2 shows the model's loss-metrics evaluation during the training process over 100 epoch , and for qualitative evaluation after the training the figure 3.3 shows the model's detections compared with the ground truth .

Box Loss

The box loss quantifies the error between the predicted bounding boxes and the ground-truth annotations. Both the training and validation curves show a consistent downward trend, indicating that the model progressively improves its localization accuracy. The validation curve follows the training curve closely, suggesting limited overfitting.

Classification Loss

Classification loss measures the model's ability to assign the correct class label to each detected object. A smooth reduction is observed across epochs in both training and validation curves. The initially high validation classification loss decreases and converges toward the training curve, demonstrating stable learning and good generalization.

DFL Loss

The Distribution Focal Loss (DFL) is used to refine bounding-box regression by modeling distances as probability distributions. The steady decline in both training and validation DFL curves indicates that the model is consistently improving its bounding-box precision over time.

Precision and Recall

Precision reflects the proportion of correct detections out of all model predictions, while recall represents the proportion of true objects detected out of all ground-truth objects. Both metrics show a clear upward trajectory throughout training. Precision increases as the model makes fewer false positives, and recall improves as fewer true objects are missed. Minor fluctuations early in training are expected and stabilize as the model converges.

Mean Average Precision

The metrics mAP50 and mAP50–95 capture overall detection performance at different Intersection over Union (IoU) thresholds. mAP50, which uses a more lenient IoU threshold, rises rapidly and approaches convergence. mAP50–95, a stricter and more comprehensive evaluation metric, increases more gradually but consistently. The upward trends in both metrics indicate that the model improves not only its detection capability but also the accuracy of predicted bounding boxes.

The training and validation curves in figure 3.2 show steady convergence over 100 epochs. Both the box, classification, and distribution focal losses decrease consistently for training and validation, indicating effective learning and minimal overfitting. Precision and recall improve rapidly during the first 50 epochs before stabilizing, with final values around 0.8 and 0.7, respectively. Similarly, the mean Average Precision (mAP50) and mAP50-95 metrics exhibit strong upward trends, reaching approximately 0.78 and 0.45, reflecting reliable detection performance across the dataset. Overall, the model demonstrates stable training dynamics and satisfactory generalization on the validation set.

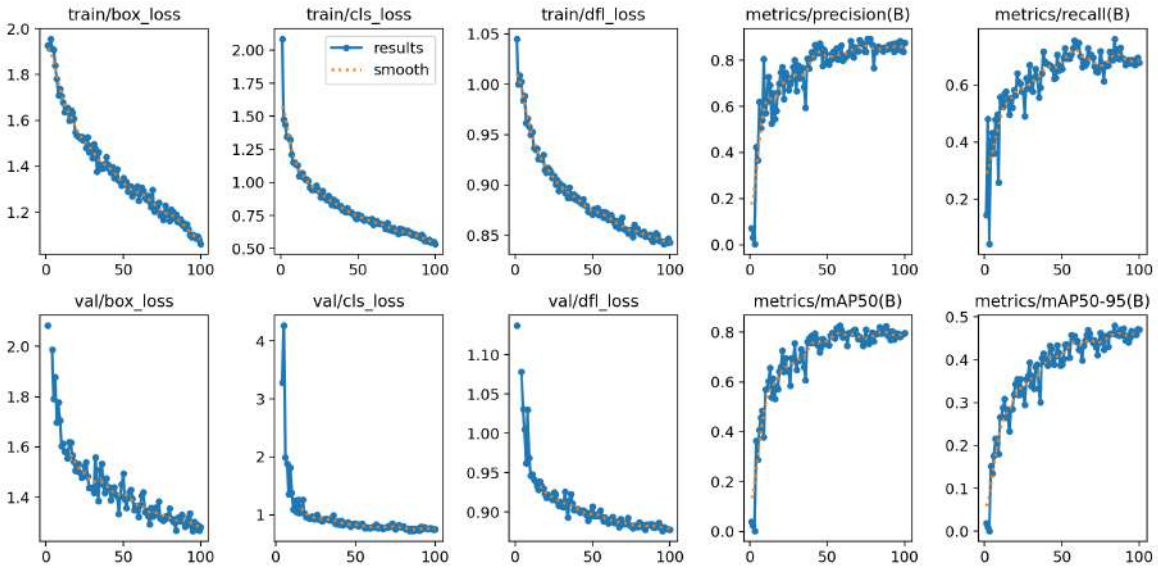


Figure 3.2: Qualitative Evaluation : Training and Validation Performance of YOLOv11l on Patched-Eagle Dataset.

Figure 3.3: Quantitative evaluation , blue boxes : model's detections , green boxes : ground truth

3.3 Blackbox X-AI Implementation

3.3.1 results

In the results , the dark red regions are the most important regions for the model to make his detections , they're the ones that are highlighted by the xAI technique , and the blue regions are the less important ones , all values are normalized using min-max normalization so that values range from 0 to 1  , however in some cases in D-RISE explanations when the whole background is green which is the average value and there is only specific region that is dark blue , that means this dark blue region is important in a way that it has negative effect on the confidence score (the specefic dark-blue region reduces the confidence score of the target under-explanation) , this will be more discussed in the 3.3.2 section .

True Positive :

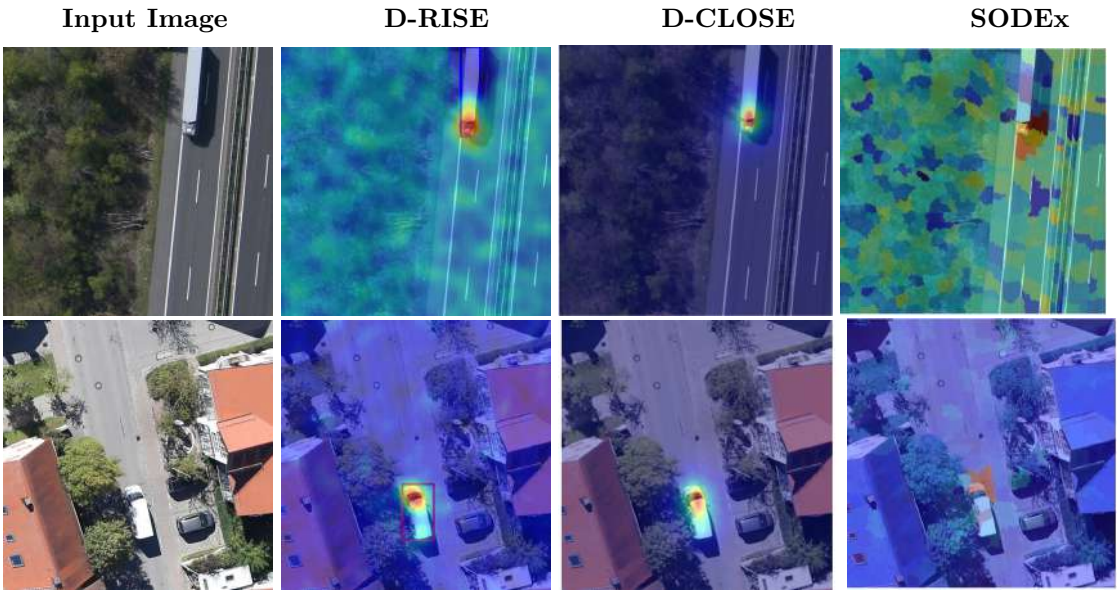


Figure 3.4: XAI visualizations highlighting the vehicle's front region and part of its shadow as the most influential areas for detection.

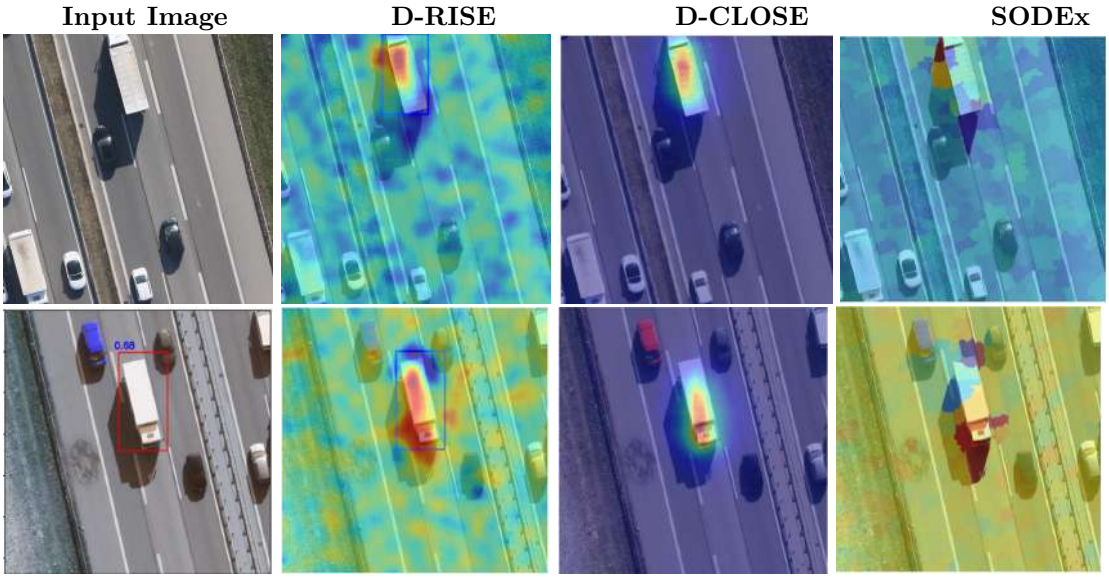


Figure 3.5: explanations tell us that the model has focused mainly on the vehicule's roof and its shadow

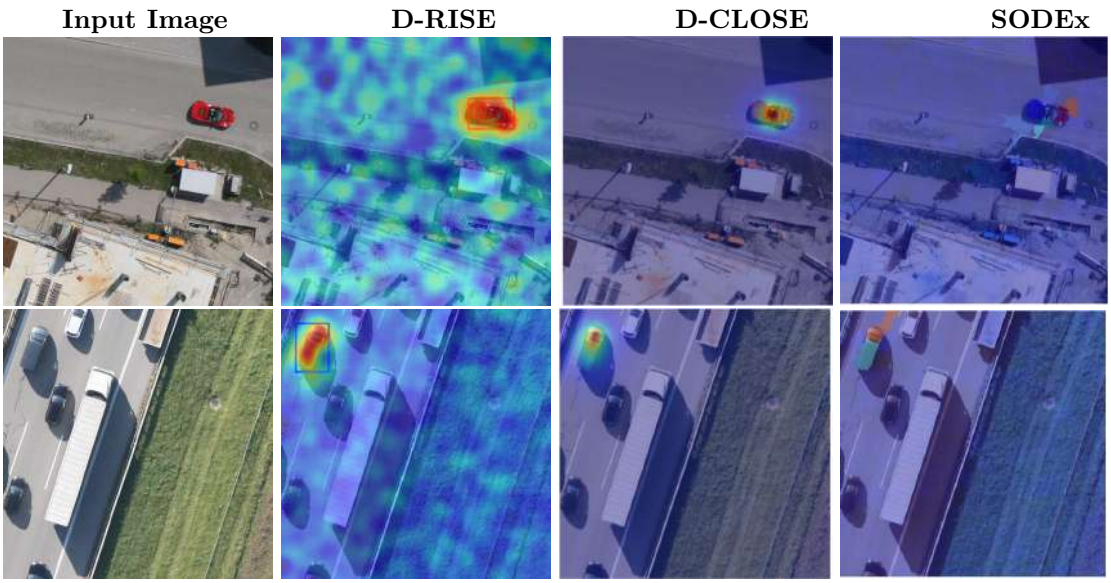


Figure 3.6: The X-AI techniques tell us that the parts of the vehicles are equally important(there is no specific important part)

False Positive

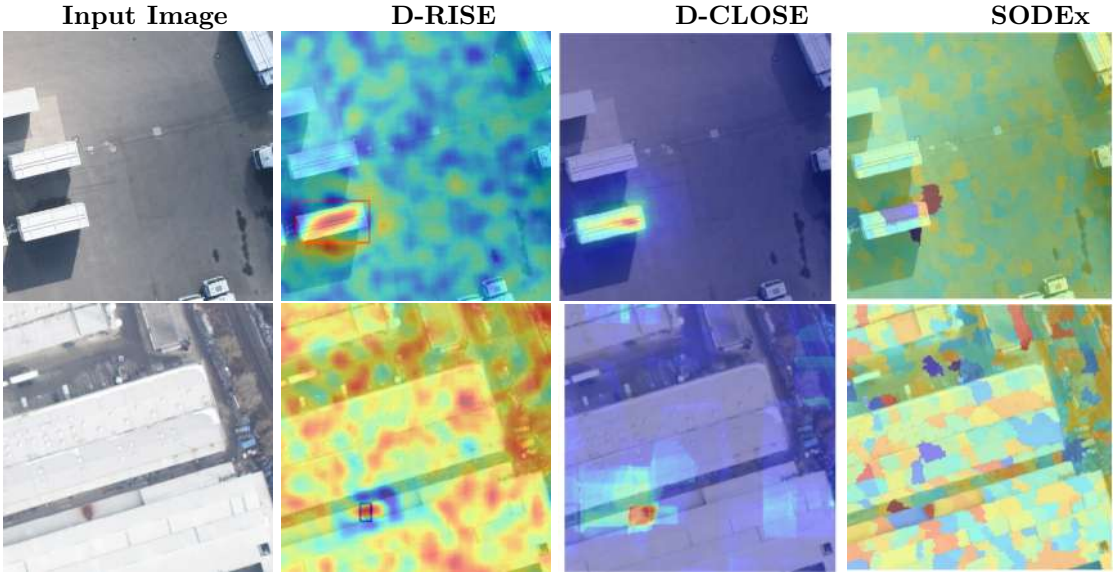


Figure 3.7: The regions that have affected these false positive detections are "white parts" and the shadows, which are similar the white trucks' features

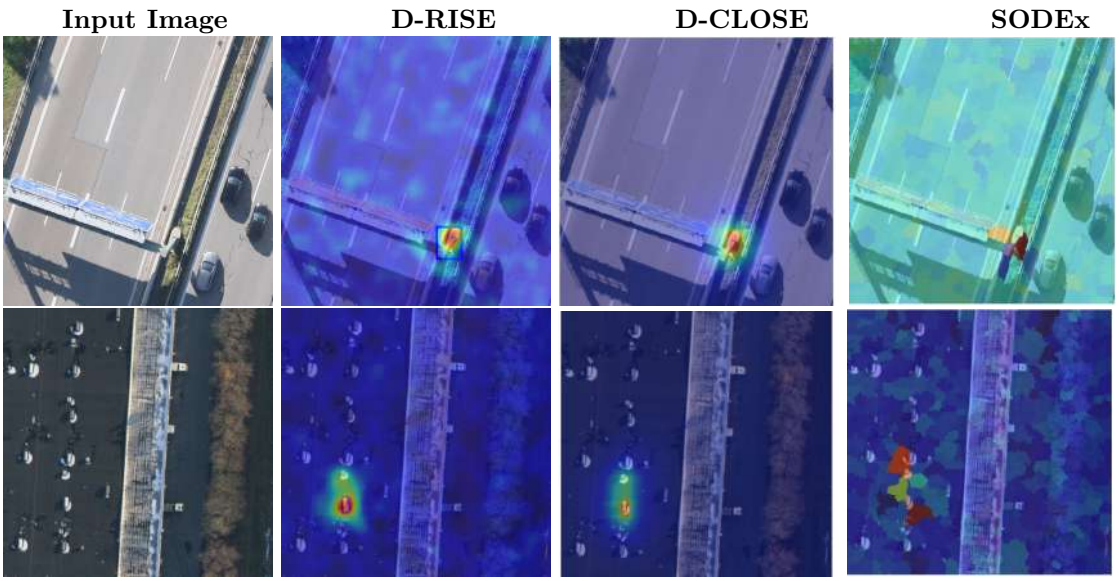


Figure 3.8: The regions that have affected these false positive detections are the objects shadows, object's head and some others that are similar to cars' features

False Negative

In the following explanations , the Dark Blue regions are the ones that have affected the confidence score of the bounding box negatively , in other words , these regions have lowered the confidence of the bounding box below 0.25 so that the model couldn't detect the object, only D-RISE xAI technique that is used , and this is explained in 3.3.2.

Figure 3.9: the buildings in front of the cars has affected the confidence scores of the target bounding boxes negatively ,so the model couldn't detect the vehicles

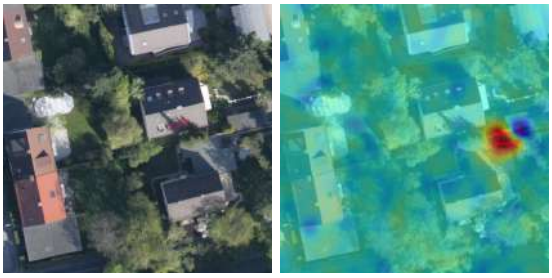


Figure 3.10: The context in which the vehicles are placed is the cause of their non-detection.

3.3.2 Discussion

D-RISE Discussion :

- In the typical case where the model produces a **high-confidence detection**, most regions in the image have a **non-negative impact** on the confidence score. Under this condition:
  - Regions that strongly support the detection (positively affecting **Confidence** $\times$ **IoU**) appear in dark red.
  - Regions that are less important for the detection appear in dark blue, not because they harm the prediction, but because their contribution is the lowest relative to the rest.

  In contrast, for **low-confidence detections**, some regions may have a negative impact on the confidence score. In these cases:
  - Truly negative contributions appear in dark blue.
  - Positive contributions remain in dark red.
  - Regions with little influence fall between the two extremes and are shown in mid-range colors such as green.

  These color variations result from applying the **min–max normalization**, defined as:

$$X_{\text{norm}} = \frac{X - X_{\text{min}}}{X_{\text{max}} - X_{\text{min}}}.$$

  This normalization scales the contribution values so that the highest positive values map to red, the lowest values map to blue, and intermediate values are represented by colors between them , that's why in low confidence objects , the background is usually green (the average value) , and the region that affected the confidence negatively is highlighted by dark blue (the lowest value) and regions that affected the confidence positively are highlighted by dark red (highest values).

- The model mainly focuses on many parts of the vehicle (head , roof , shadow..) , the roof , glasses and shadows are the most important ones (they're highlighted with dark red) .
- vehicle's shadow and its roof top are so important for the model to make detections , either for false positive or true positive
- at the explanation of the first row in figure 3.4 , the dark blue region at the back of the truck has negative effect on the confidence score , meaning that if this part is removed , there would be another bounding box highlighting the same truck with higher Confidence , to validate this , the dark-blue region is hidden with black pixels.

Figure 3.11: Hidden Image

⇒ In figure 3.11,the bounding box is smaller than the original one with confidence so much higher than the original one , this can be interpreted as the model has trained much more on the smaller trucks.So the trucks with such long back-roof are unfamiliar to the model that's why it's been highlighted as dark blue in the explanation (affects the confidence negatively) . To overcome this limitation,training the model on much more objects of truck that has such long back-roof will be a solution so that the model gets familiar with these type of long trucks.

- Most of the false-positive explanations show that the model focuses on the rooftop (in the case of trucks), shadows, and the frontal region . For example, The Explanations in figure 3.7, the model concentrated heavily on the white parts , which closely resembles that of a truck, leading to an incorrect detection. Training the model on additional objects that share similar rooftop characteristics may help reduce these types of false-positive detections.
- The explanations in figure 3.8 , show that the dark red region covers the head's vehicle (and its shadows) , these regions contributed the most to the false-positive detection , this can be interpreted that these features are so much similar to the small vehicle's features , that's why the model mistakenly detected them.
- in False Negative samples , we must focus on the dark blue regions rather than the dark-red regions , because these are the regions that negatively affected the confidence score so that the model couldn't detect the object.
- most of the dark blue regions cover the context in which the car is located , the explanations show that the buildings in front of the cars are the ones that highly affected the confidence score negatively , so that the model couldn't detect the vehicles. to validate this , these dark blue regions are hidden with black pixels. *RightAroow*

Figure 3.12: Hidden Image

⇒After Hiding the Building which is covered by dark blue in the explanation , we can see that the model successfully detected the object with confidence 0.51 , this can be interpreted as the model uses the contextual information to make his detections , The model is trained on vehicles on the road or in the garage.. , so the building in front of the vehicle was unfamiliar region for the model , to overcome this limitation , training the model on more different contexts , in other words , putting the vehicles on diverse locations , (behind buildings , near trees ...)

D-CLOSE Discussion :

heatmap explanation:

- D-CLOSE uses a **cascading fusion process** as shown in figure 1.1, which preserve only the regions that have positively affecting the detection (IOU × confidence) .
- **Dark red:** Regions that strongly support the detection.

- **Blue regions:** Regions that are not important, but *do not indicate negative contribution.*
- Unlike D-RISE, D-CLOSE **does not highlight regions that reduce confidence**.
- Effect: Saliency maps are **more focused on critical regions**, but provide **less information about negative contributions**.

Implications for High- and Low-Confidence Detections:

- **High-confidence detections:**
  * Dark red regions highlight the most important features for detection (e.g., roof, shadows, headlights).
  * Blue regions indicate regions with low contribution but are not harmful.
- **Low-confidence detections:**
  * Since D-CLOSE discards regions with negative impact, explanations provide **limited insights**.
  * Regions that contributed negatively to low confidence are ignored, making it difficult to analyze failure causes.

Analysis of True Positives:

- Model mainly relies on **vehicle roofs, shadows, and headlights** for detection.
- These regions appear in **dark red**, showing the strongest positive contribution.
- Blue regions are of minor importance, providing context but not affecting the score of the target box,which is the object under explanation.

Limitations for False Positives:

- False positives are influenced by features resembling learned object characteristics (e.g., small vehicle roofs similar to truck roofs).
- D-CLOSE highlights the **supporting features (dark red)** but does **not indicate features that may have reduced confidence**.
- Therefore, it is **less informative for diagnosing why a misclassification occurred**.
- **Mitigation strategies:**
  * Train on objects with similar positive features (e.g., roof shapes) to reduce false positives.
  * D-CLOSE can guide model improvement for highlighted features.

Limitations for False Negatives:

- In low-confidence or false-negative detections, the most informative regions are those with **negative contributions**.
- D-CLOSE **ignores these regions**, so explanations **cannot reveal why the model missed the object**.
- Example: A car hidden behind a building or occluded by other objects would have its negatively contributing regions discarded, leaving almost no insight.
- **Consequence:** D-CLOSE is **not suitable for analyzing false-negative samples**.

Takeaways and Recommendations:

- **Strengths of D-CLOSE:**
  - Focuses on the **most important positive regions**, making explanations concise and targeted.
  - Reduces noise from low-impact or irrelevant regions.
  - Ideal for understanding **what features the model relies on to make detections**.
- **Limitations:**
  - Provides **limited insight into negative contributions**, particularly for low-confidence and false-negative detections.
  - Less useful for debugging missed detections or understanding context-related failures.
- **When to Use:**
  - Best suited for **high-confidence detections** and understanding **positive driving features**.
  - Not recommended for **false negatives** or analyzing **why detections fail**.

Sodex Discussion :

The explanations provided by SODEx rely on **segmentation-based methods** as discussed in 2.1.1.3, which affects both their accuracy and faithfulness. Since the surrogate model maps object detection to a classification task and then generates explanations via segmented regions, the resulting saliency maps are coarser and less precise compared to pixel-level methods like D-RISE or D-CLOSE.

- **Lower spatial accuracy:** The segmentation boundaries may not perfectly align with the object's true shape, causing important features to be partially or incorrectly highlighted.
- **Reduced faithfulness:** Because the explanation is derived from a surrogate classifier rather than the original detector, some regions that genuinely influence the detector's confidence may be omitted or misrepresented.
- **Coarse interpretation:** While the method can indicate general areas of importance, it provides less detailed insight into exactly which object parts drive the model's predictions.

Overall, SODEx explanations are useful for obtaining a general understanding of which regions contribute to detection, but they are inherently **less precise and less faithful** than methods that analyze the detector directly.

3.4 white-box xAI implementation

3.4.1 Adapting Grad-CAM and HiResCAM for Object Detection

Traditional visual explanation methods such as Grad-CAM and HiResCAM were originally designed for image classification models. In their standard formulation, these methods compute the gradient of a class-specific probability with respect to a convolutional activation map, and use this information to identify the most influential spatial regions in the input image. However, object detectors differ fundamentally from classifiers: instead of producing a single class probability, they output thousands of bounding-box predictions, each with an objectness score, class scores, and spatial parameters. Therefore, applying Grad-CAM or HiResCAM directly to a detector without modification is not meaningful.

To convert these explanation methods into object-detector explainers, we modify the target used during backpropagation. Instead of differentiating a class probability, we use the **confidence score of the specific bounding-box prediction** we want to explain. This confidence score

is taken **before post-processing**, since operations such as Non-Maximum Suppression (NMS) and thresholding are non-differentiable, and would otherwise block gradient flow.

Let c_i denote the confidence score of the i-th predicted bounding box. For the bounding box of interest, we treat c_i as the scalar target for Grad-CAM/HiResCAM and compute:

$$\frac{\partial c_i}{\partial A_k},$$

where A_k is the activation map of the selected convolutional layer. This modification allows Grad-CAM and HiResCAM to highlight the spatial regions that most strongly contributed to that *specific detection*, rather than to a global class prediction. Consequently, the method becomes properly aligned with the internal structure and decision process of object detection networks.

Selecting the Appropriate Detection Layer Based on Object Index :

Modern object detectors, including YOLO-based architectures, generate predictions at multiple scales. Each detection head is responsible for objects of a specific size: high-resolution feature maps detect small objects, medium-resolution maps detect medium-sized objects, and low-resolution maps detect large objects. To obtain a meaningful explanation, the gradient must be computed with respect to the **correct detection layer**—the layer that produced the bounding-box prediction.

Figure 2.2.2.1 illustrates how prediction indices map to object scales. The mapping is determined by the number of anchors predicted at each scale:

- **Small objects:**
  - Spatial resolution: 80×80
  - Number of predictions: $80 \times 80 = 6400$
  - Index range:
    $$0 \leq \text{object\_index} \leq 6400$$
  - Corresponding layer: **Layer 16**
- **Medium objects:**
  - Spatial resolution: 40×40
  - Additional predictions: $40 \times 40 = 1600$
  - Index range:
    $$6400 < \text{object\_index} \leq 6400 + 1600 = 8000$$
  - Corresponding layer: **Layer 19**
- **Large objects:**
  - Spatial resolution: 20×20
  - Additional predictions: $20 \times 20 = 400$
  - Index range:
    $$8000 < \text{object\_index} \leq 8000 + 400 = 8400$$
  - Corresponding layer: **Layer 22**

Selecting the appropriate detection layer ensures that the gradients originate from the exact head responsible for the bounding-box prediction. This alignment is essential for producing explanations that accurately reflect the detector's multi-scale prediction mechanism.

3.4.2 results

The following results highlight the saliency maps generated by white-box techniques : Grad-CAM and HiresCAM

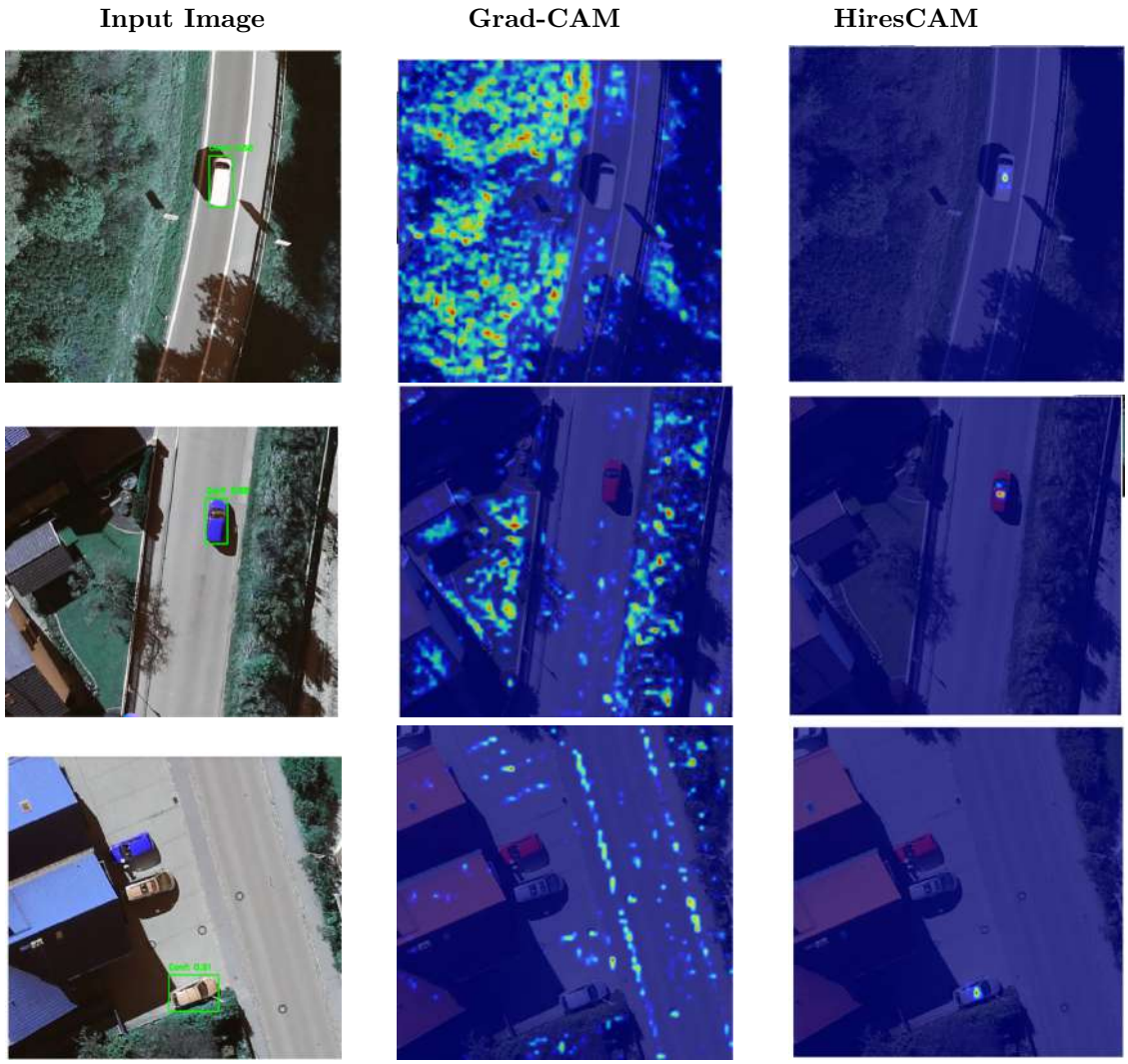


Figure 3.13: explanations with white-box X-AI Techniques

3.5 limitations

White-Box X-AI techniques

After computing the weighted sum of the activation maps in Grad-CAM-like methods, the resulting heatmap has the same spatial resolution as the final feature map. Each spatial cell in this heatmap encodes a highly *compressed representation* of the image regions that the network used to make its prediction. However, simply upsampling this heatmap back to the original image size using interpolation does not correctly recover the true spatial extent of the features represented by each grid cell. This problem becomes especially severe for detectors with complex backbone architectures, such as YOLOv11, which employ Feature Pyramid Networks (FPN) and multi-scale aggregation. In such architectures, the relationship between a feature-map cell and the input image is highly non-linear and irregular, making naive interpolation fundamentally inaccurate.

A more critical limitation arises when computing the gradient of a bounding-box confidence score with respect to the activation maps. In modern detectors, each detection head produces predictions via a 1×1 convolution applied to the final feature maps. As a result, **each bounding-box confidence score is generated from a single spatial cell in the detection head**. Only one grid cell is directly responsible for the confidence value; all other cells contribute nothing to that specific prediction.

When backpropagating the confidence score into the backbone feature maps, the gradient therefore becomes zero everywhere except in the receptive field of that single responsible grid cell. This produces a gradient map that has a very small localized region of non-zero values, while all other grid cells remain exactly zero. The non-zero region corresponds only to the subset of cells that fall inside the receptive field of the responsible detection-head cell.

HiResCAM computes its heatmap by performing an element-wise multiplication between each activation map and its corresponding gradient map, followed by summation. Since the gradient map itself is zero almost everywhere, the final HiResCAM heatmap also becomes zero everywhere except in a small spatial region. After naive upsampling, the heatmap collapses to a small blob located near the center of the bounding box, failing to highlight the true spatial cues used by the detector.

Grad-CAM has an even greater limitation: by performing global average pooling (GAP) on the gradient map, it reduces the sparse gradient signal to a set of scalar weights. These weights are then applied uniformly across the entire activation maps. For detectors, where gradients are highly localized and sparse, this produces heatmaps that do not correspond to any meaningful spatial reasoning. After upsampling, the resulting visualization becomes noisy or completely non-interpretable.

The root cause of these failures is that **the spatial structure of object detectors is not compatible with naive interpolation of activation-space heatmaps**. Because each detection is produced by a single grid cell, and because gradients propagate only through the receptive field of that cell, gradient-based explainers reveal only a very small region of influence. The only principled way to obtain a meaningful heatmap is to explicitly reconstruct the **true receptive field** of every contributing grid cell in the backbone. Without this step, Grad-CAM, HiResCAM, and related techniques cannot reliably visualize the spatial reasoning used by object detectors.

Black-Box XAI Techniques

Black-box Explainable AI (XAI) methods aim to interpret model behavior solely through input–output observations, without accessing the internal parameters or computational graph. Although this model-agnostic nature makes them widely applicable, it also introduces several notable limitations.

First, these approaches typically rely on generating a large number of perturbed inputs and repeatedly querying the model to observe how each perturbation affects the prediction. Methods such as D-RISE, D-CLOSE, and LIME often require hundreds or even thousands of masked or modified samples per explanation. Since every perturbed sample necessitates a full forward pass through the network, the computational overhead becomes substantial. This issue is amplified in high-dimensional domains such as image analysis, where deep neural networks impose high inference costs.

Moreover, the dependence on repeated sampling leads to significant latency, making black-box methods impractical for real-time or large-scale applications. Their sampling-based nature can also result in noisy or unstable explanations, as variations in perturbation strategies or random seeds may produce noticeably different outputs. Finally, because these methods operate exclusively on observable predictions, they cannot leverage internal model information such as gradients or intermediate feature activations, limiting the granularity and interpretability of the generated explanations.

In summary, while black-box XAI methods provide broad applicability and model-agnostic interpretability, they suffer from high computational cost, potential instability, and restricted explanatory depth compared to white-box approaches.

Conclusion

Black-box methods successfully identified that the detector relies on vehicle roofs, shadows, and contextual information, revealing why false positives occur (similar visual features) and false negatives happen (unfamiliar contexts). However, they require 1,000-5,000 forward passes per explanation—computationally prohibitive for real-world deployment. White-box methods failed fundamentally. The sparse gradient problem (only one grid cell contributes non-zero gradients per detection) combined with naive upsampling produced spatially inaccurate blobs. Standard HiResCAM achieved negative insertion-deletion scores, confirming architectural incompatibility. This reveals an unavoidable trade-off: black-box methods achieve faithfulness at extreme computational cost, while white-box methods offer efficiency but produce unreliable explanations. Chapter 4 presents our novel approach designed to overcome this limitation by explicitly reconstructing receptive fields rather than naively upsampling activation-space heatmaps.

Chapter 4

Proposed Method: Receptive-Field–Based HiResCAM for Object Detectors

Introduction

This chapter introduces Receptive-Field-Based HiResCAM, designed to overcome the efficiency-faithfulness trade-off identified in Chapter 3. Rather than naively upsampling activation heatmaps, we explicitly reconstruct each grid cell's receptive field through input-image gradient computation, accounting for deep convolutional backbones, Feature Pyramid Networks, and multi-scale detection heads. The method includes: (1) computing sparse grid-cell importance via modified HiResCAM, (2) reconstructing exact receptive fields, (3) Gaussian smoothing for interpretability, (4) confidence-weighted aggregation across pre-NMS bounding boxes, and (5) flexible target selection enabling explanation of confidence scores, width, height, or localization parameters.

4.1 Approach

In this subsection, we introduce a new explanation technique specifically designed to overcome the fundamental limitations of Grad-CAM and HiResCAM when applied to modern object detectors such as YOLOv11. The central idea is to replace naive upsampling of activation-space heatmaps with a principled reconstruction of the *true receptive field* of the feature-map cells that contributed to a detection. Our method extends the philosophy of HiResCAM—using element-wise activation–gradient interactions—while enabling spatially faithful saliency reconstruction in the input image domain.

The method is divided into several stages: (1) producing sparse grid-cell–level importance via modified HiResCAM, (2) computing the exact receptive field of each contributing grid cell through input-image gradients, (3) synthesizing cell-level receptive-field maps, (4) combining these maps to form an object-level explanation, and (5) aggregating explanations across all bounding boxes that represent the same physical object. Each stage is described in detail below.The figure 4.3 shows the pipeline for one bounding box.

Step 1: Producing Sparse Importance Maps via HiResCAM

As established in the previous section, the gradient of a bounding-box confidence score c_i with respect to the activation maps A_k of the detection head is non-zero only within the receptive field of the single feature-map cell responsible for the prediction. Let $A \in \mathbb{R}^{H \times W \times C}$ denote the

final activation tensor of the detection head, and let $G = \frac{\partial c_i}{\partial A}$ denote the gradient tensor of the same shape.

Traditional Grad-CAM computes weights using global average pooling:

$$\alpha_k = \frac{1}{HW} \sum_{x,y} G_{x,y,k}, \tag{4.1}$$

then forms the heatmap

$$L_{\text{Grad-CAM}} = \sum_{k=1}^{C} \alpha_k A_k. \tag{4.2}$$

However, because G is sparse (zero almost everywhere), these averages become uninformative.

HiResCAM instead uses an element-wise interaction:

$$L_{\text{HiResCAM}}(x, y) = \sum_{k=1}^{C} A_{x,y,k} \cdot G_{x,y,k}, \tag{4.3}$$

which preserves localization. Due to gradient sparsity, L_{HiResCAM} contains only a small set of non-zero grid cells:

$$\Omega = \{(x, y) \mid L_{\text{HiResCAM}}(x, y) \neq 0\},$$

where $|\Omega|$ is typically extremely small (often 1 to 5 cells).

Instead of upsampling L_{HiResCAM} naively—which would yield spatially incorrect blobs—we treat each grid cell $(x, y) \in \Omega$ as a distinct semantic contributor whose receptive field must be reconstructed.

Step 2: Computing the Receptive Field via Input-Image Gradients

For each grid cell $(x, y) \in \Omega$, we compute the gradient of its activation value with respect to the input image. Let $A_{x,y} \in \mathbb{R}^C$ denote the activation vector at grid cell (x, y), and let $I \in \mathbb{R}^{H_0 \times W_0 \times 3}$ denotes the input RGB image. We compute:

$$R^{(x,y)} = \frac{\partial A_{x,y}}{\partial I} \in \mathbb{R}^{H_0 \times W_0 \times 3}. \tag{4.4}$$

This gradient tensor describes how each input pixel in each color channel affects the activation of that grid cell. Because detectors use deep convolutional backbones (with strides, pooling, FPN, nearest/fusion connections), a grid cell's receptive field is not a clean square—this gradient-based computation naturally captures the true spatial dependencies.

To measure the total contribution of each pixel (regardless of channel sign), we collapse the channel dimension using the Euclidean norm:

$$S^{(x,y)}(i, j) = \left\| R^{(x,y)}(i, j, :) \right\|_2 = \sqrt{\sum_{c=1}^{3} \left(R^{(x,y)}(i, j, c) \right)^2}. \tag{4.5}$$

Thus, $S^{(x,y)} \in \mathbb{R}^{H_0 \times W_0}$ is a 2D saliency image representing the strength of influence of each pixel on the activation of grid cell (x, y). Negative effects are preserved inside the L_2 magnitude, ensuring that both excitatory and inhibitory contributions are captured.The figure 4.1 shows the receptive field in 2D as an image and also in 1D.

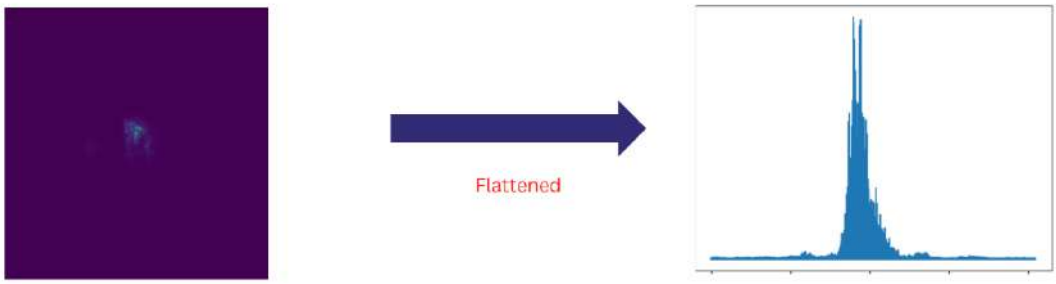


Figure 4.1: receptive-field

As shown in the figure 4.1 , the raw gradient maps $S^{(x,y)}$ are often noisy due to local pixel-level fluctuations, so we apply a Gaussian filter:

$$\tilde{S}^{(x,y)} = \text{GaussianBlur}\left(S^{(x,y)}\right),$$

yielding a smooth receptive field , providing more interpretable saliency maps.

The figure 4.2 shows the receptive field after smoothing:

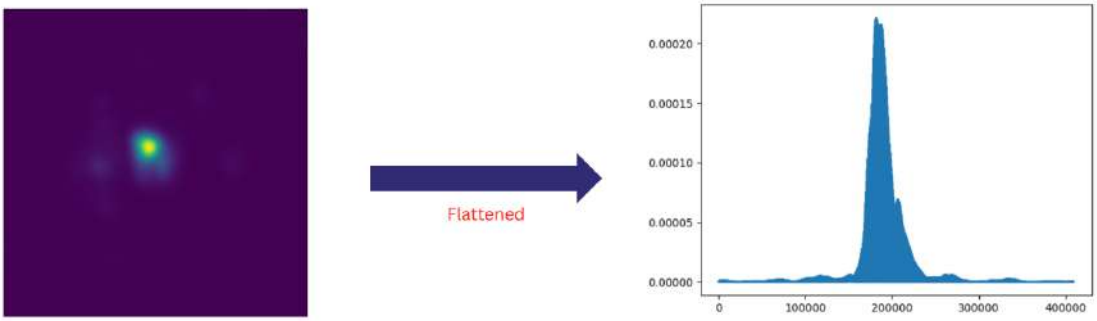


Figure 4.2: GradCAM pipeline

.

To ensure that each grid-cell contributes only its spatial receptive-field pattern—and not arbitrary magnitude differences—we normalize the smoothed receptive-field map $\tilde{S}^{(x,y)}$ to the range $[0, 1]$. This step removes scale variations caused by different gradient magnitudes across grid cells. The normalized receptive field is computed as

$$\hat{S}^{(x,y)} = \frac{\tilde{S}^{(x,y)} - \min\left(\tilde{S}^{(x,y)}\right)}{\max\left(\tilde{S}^{(x,y)}\right) - \min\left(\tilde{S}^{(x,y)}\right)}. \tag{4.6}$$

This guarantees that the receptive field represents only the *shape and location* of influence in the input image, while the importance of the grid cell is handled exclusively by the HiResCAM weight $w_{x,y}$ in the later multiplication step.

Step 3: Reconstructing Grid-Cell–Level Explanations

We now combine each grid cell's contribution weight from HiResCAM with its reconstructed receptive field. Let

$$w_{x,y} = L_{\text{HiResCAM}}(x, y) \tag{4.7}$$

denotes the importance assigned to grid cell (x, y).

The reconstructed contribution map for that cell is:

$$M^{(x,y)}(i, j) = w_{x,y} \cdot \tilde{S}^{(x,y)}(i, j). \tag{4.8}$$

Unlike naive upsampling, this operation uses the *true* receptive field of the grid cell to determine where the activation contributed in the input image. The full explanation for bounding box i is obtained by summing over all non-zero cells:

$$M_i = \sum_{(x,y)\in\Omega} M^{(x,y)}. \tag{4.9}$$

Step 4: Multi-Bounding-Box Aggregation for Faithful Object-Level Explanations

A crucial detail in YOLO-based detectors is that multiple bounding boxes (before NMS and thresholding) may correspond to the same underlying object. YOLO considers two boxes to represent the same object when their Intersection over Union (IoU) satisfies:

$$\text{IoU}(\text{box}_i, \text{box}_j) > 0.45.$$

Thus, for a target object, we find the set:

$$\mathcal{B} = \{j \mid \text{IoU}(i, j) > 0.45\},$$

where i is the index of the reference bounding box chosen for explanation.

For each bounding box $j \in \mathcal{B}$, we repeat the entire receptive-field reconstruction pipeline described above, producing object-aligned saliency maps M_j.

To aggregate these maps, we first normalize the confidences of the boxes in $\mathcal{B}$ by dividing each confidence c_j by the sum of all confidences in the set:

$$\tilde{c}_j = \frac{c_j}{\sum_{k\in\mathcal{B}} c_k}. \tag{4.10}$$

This ensures that the coefficients sum to one. We then compute the confidence-weighted fusion:

$$M_{\text{final}} = \sum_{j\in\mathcal{B}} (\tilde{c}_j \cdot M_j), \tag{4.11}$$

where $\tilde{c}_j$ is the normalized confidence coefficient of bounding box j prior to NMS , the figure 4.4 shows the pipeline of the saliency maps aggregation .

Step 5: Summary of the Full Pipeline

The proposed explanation method can be summarized as:

1. Compute modified HiResCAM on the detection head to obtain a sparse grid-cell heatmap.
2. Identify the set of grid cells with non-zero values.
3. For each grid cell:
   (a) Compute its gradient with respect to the input image.

(b) Convert gradients to a 2D receptive-field magnitude map via L_2 norm.
(c) Smooth using a Gaussian filter.
(d) Multiply by the cell's HiResCAM weight.

4. Sum all cell contributions to form the explanation for one bounding box.
5. Repeat for all bounding boxes with IoU > 0.45 relative to the reference box.
6. Fuse all explanations using a confidence-weighted sum.

This approach avoids all pitfalls of naive upsampling, reconstructs spatially meaningful receptive fields, and aggregates multi-box evidence to produce a faithful, high-resolution saliency map for each detected object.

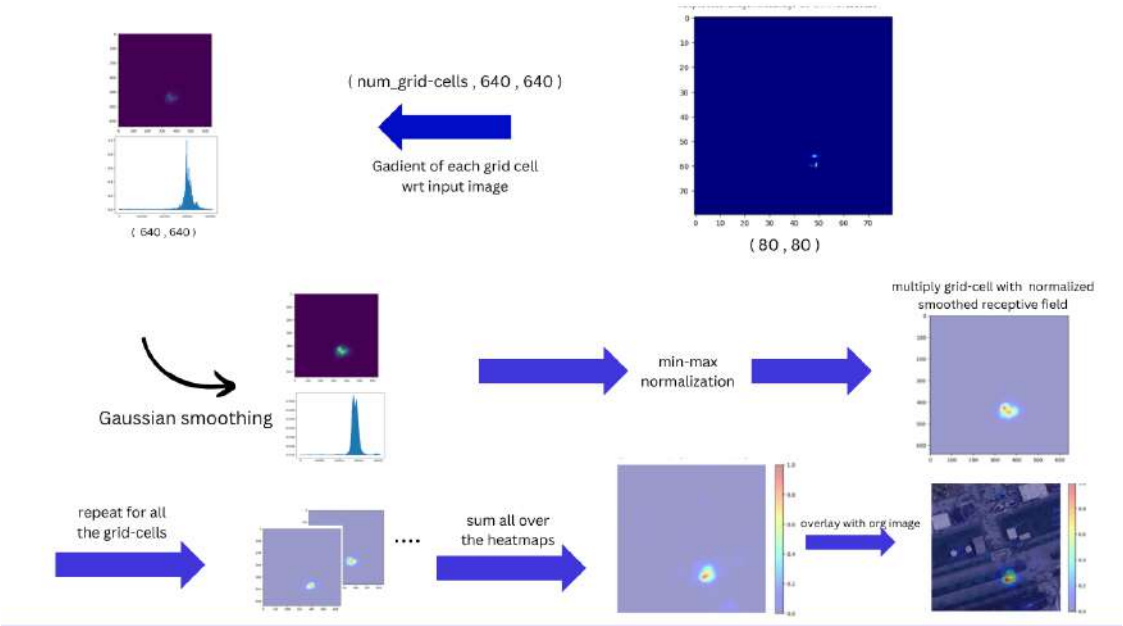


Figure 4.3: pipeline

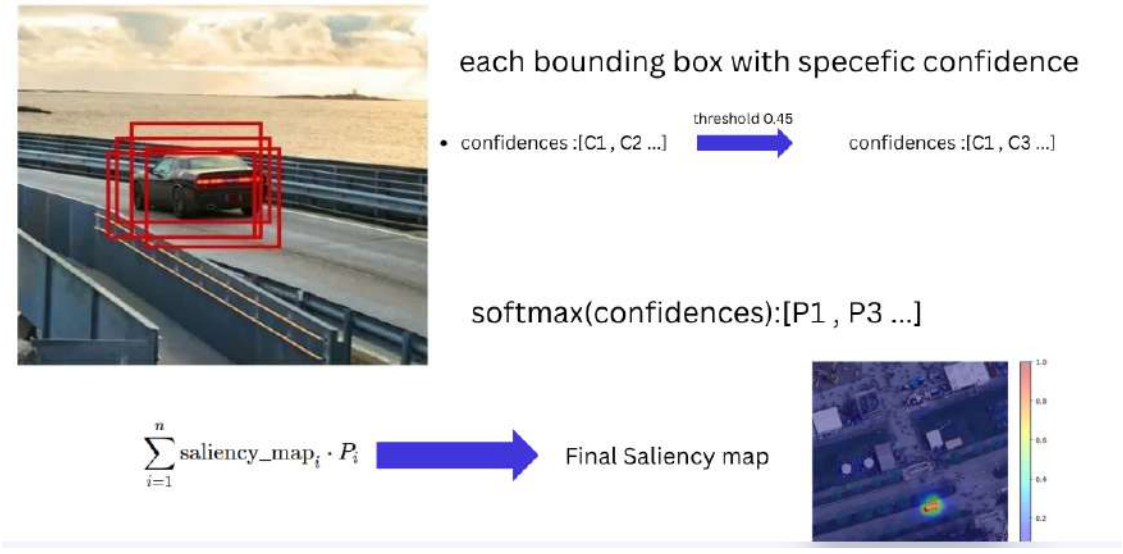


Figure 4.4: bounding-boxes aggregation

4.2 Results

A key advantage of the white-box techniques is its flexibility in selecting *any* target output of the detector to be explained. Unlike existing black-box methods, which are restricted to class probabilities * IOU white box approaches allow us to compute explanations for individual components of the bounding-box prediction vector. For example, when a detection is mislocalized, we can directly choose the *width*, *height*, or even the *center coordinates* (x, y) as the target output . This enables a much deeper analysis of failure cases: rather than merely visualizing "why the detector thinks an object exists," we can visualize *why the detector predicted a box that is too wide*, or *why the predicted center is shifted*. This fine-grained interpretability is particularly valuable for debugging object detectors and understanding spatial reasoning in modern architectures such as YOLOv11. in this subsection we will choose mainly the confidence score , width and height are the outputs to be explained.

Confidence-Explanation

The following explanations show the saliency maps generated by our proposed technique Receptive-Field–Based HiResCAM using the confidence score as the output to be explained.

True Positive

HiResCAM .

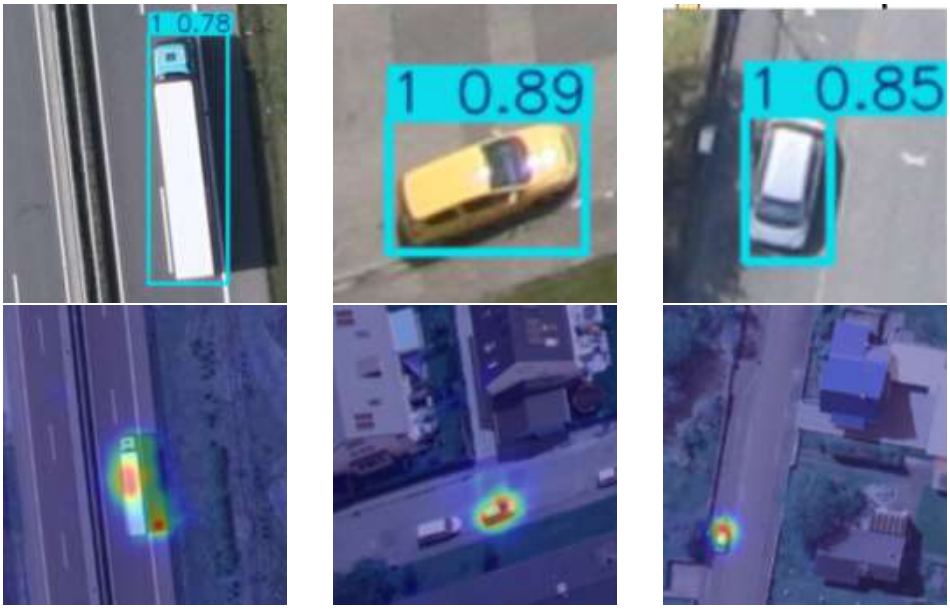


Figure 4.5: Explanations on High-Confidence objects

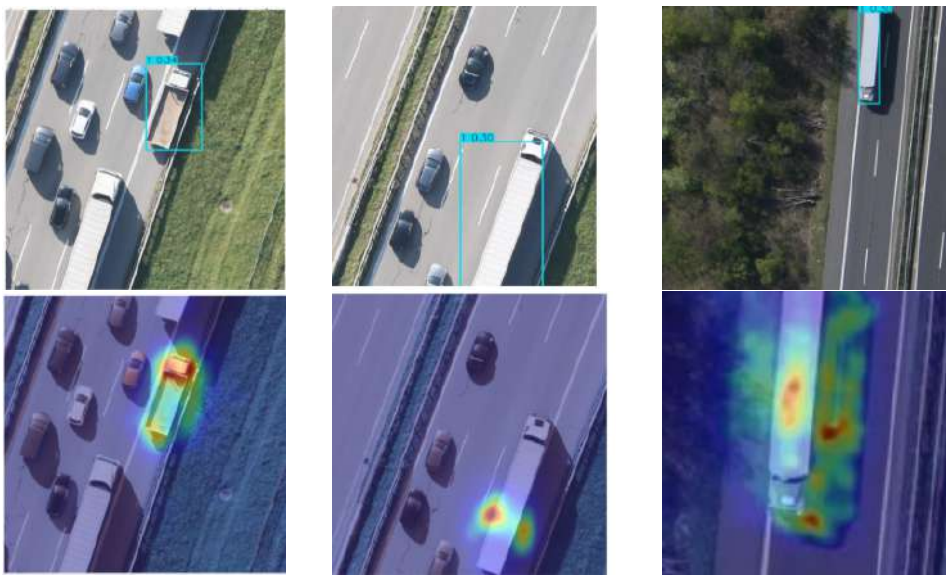


Figure 4.6: Explanations on Low-Confidence objects

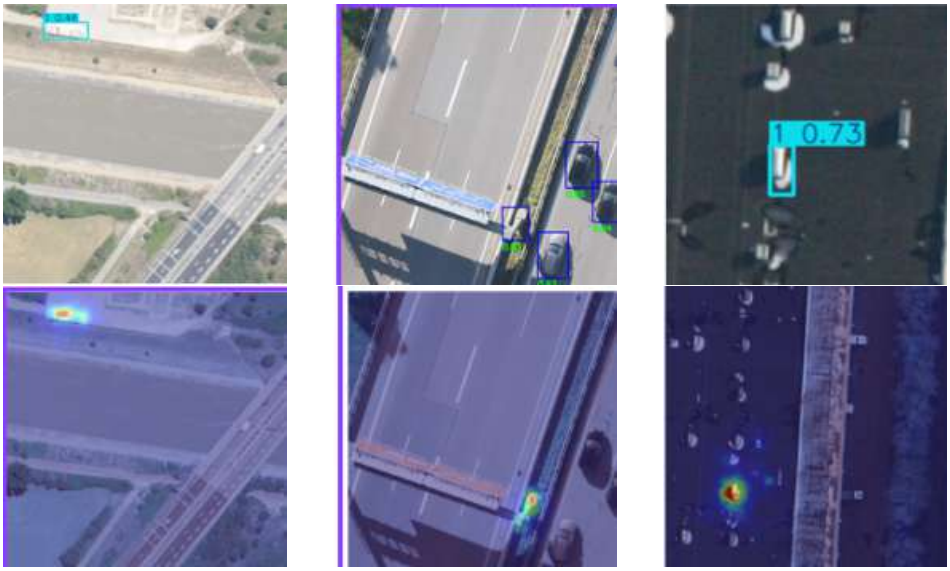


Figure 4.7: explanation on false positive samples

Width and Height Explanation

The figure 4.8 show the explanations when changing the target output from confidence to object width and height where the first row shows the original images with the target bounding box; the second row presents the width explanations and the third row presents the height explanations.

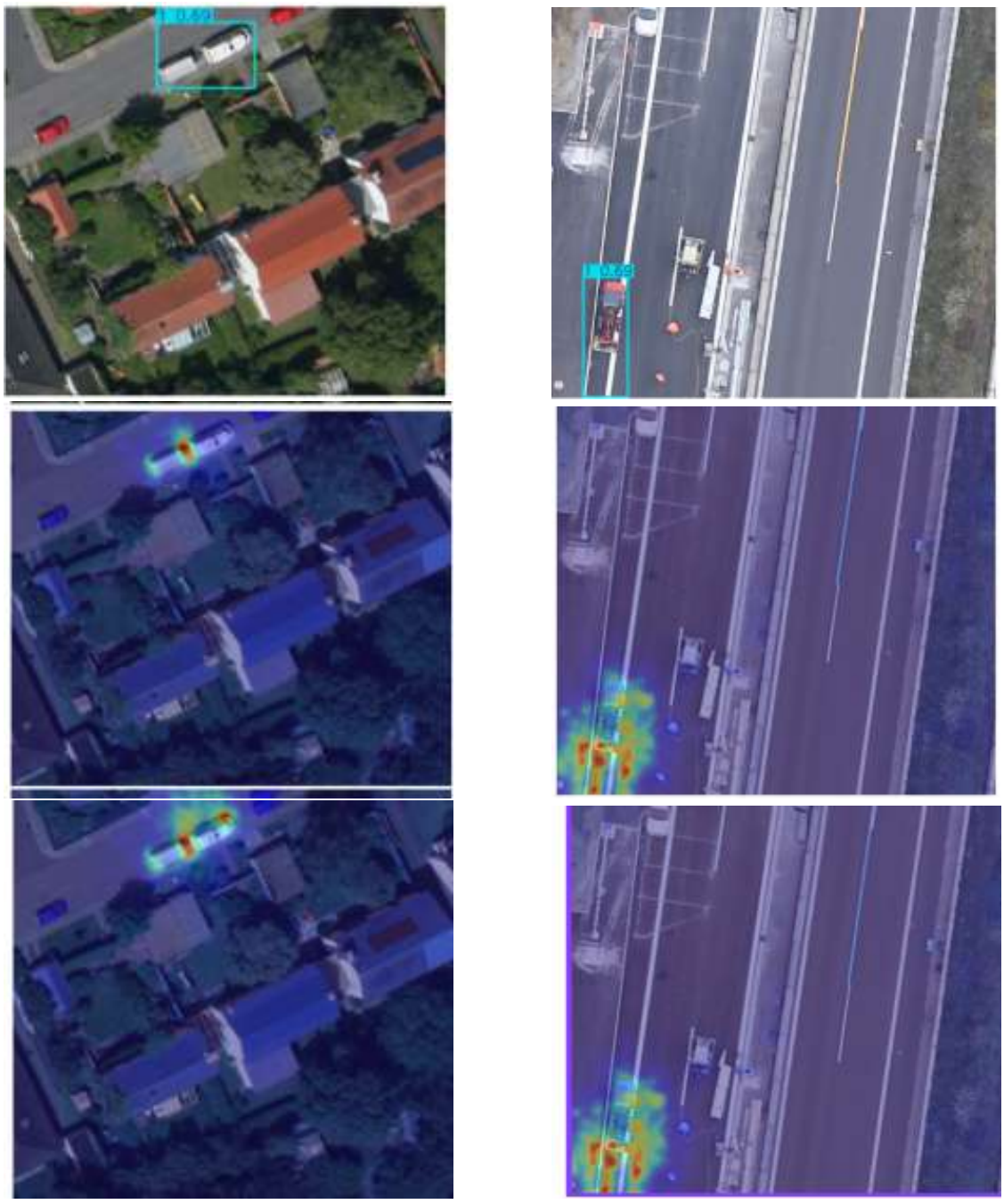


Figure 4.8: Width and Height explanation

4.3 Evaluation and comparison

4.3.1 Insertion–Deletion Metric

The insertion–deletion metric is a widely used quantitative measure for assessing the faithfulness of explainable AI (XAI) techniques, especially those that produce pixel- or feature-level importance maps. The core idea is that a faithful explanation should identify features whose addition strongly increases the model's confidence and whose removal strongly decreases it.The figure 4.9 shows the full pipeline of the insertion-deletion metric.

Insertion Metric

The insertion metric evaluates how quickly the model's confidence increases as important pixels are gradually reintroduced into a blurred baseline image.

Procedure :

1. **Generate explanation:** Obtain the importance map produced by the XAI method, flatten it, and sort the pixels in descending order of importance.
2. **Initialize baseline:** Begin with a blurred version of the original image.
3. **Progressive insertion:** Iteratively insert pixels into the baseline image, starting from the most important and ending with the least important.
4. **Confidence monitoring:** After each insertion step, record the model's confidence for the target class.
5. **AUC calculation:** Plot the confidence as a function of the proportion of inserted pixels and compute the area under this curve, denoted as AUC_{ins}.

A faithful explanation yields a steep confidence increase, resulting in a larger insertion AUC.

Deletion Metric

The deletion metric measures how strongly the model's confidence drops when important pixels are removed from the original image.

Procedure :

1. **Generate explanation:** Reuse the same sorted list of pixel importances.
2. **Initialize image:** Start from the unmodified original image.
3. **Progressive deletion:** Iteratively remove (e.g., replace with black) the most important pixels first, followed by less important ones.
4. **Confidence monitoring:** After each deletion step, record the model's confidence for the target class.
5. **AUC calculation:** Plot the confidence curve as pixels are removed and compute the area under this curve, denoted as AUC_{del}.

A faithful explanation causes a rapid confidence drop, leading to a smaller deletion AUC.

Final Faithfulness Score

The overall faithfulness score is computed as:

$$\text{Score} = \text{AUC}_{\text{ins}} - \text{AUC}_{\text{del}}.$$

A larger score indicates that the explanation strongly influences the model's confidence in the expected directions, reflecting greater faithfulness to the model's internal decision-making process.

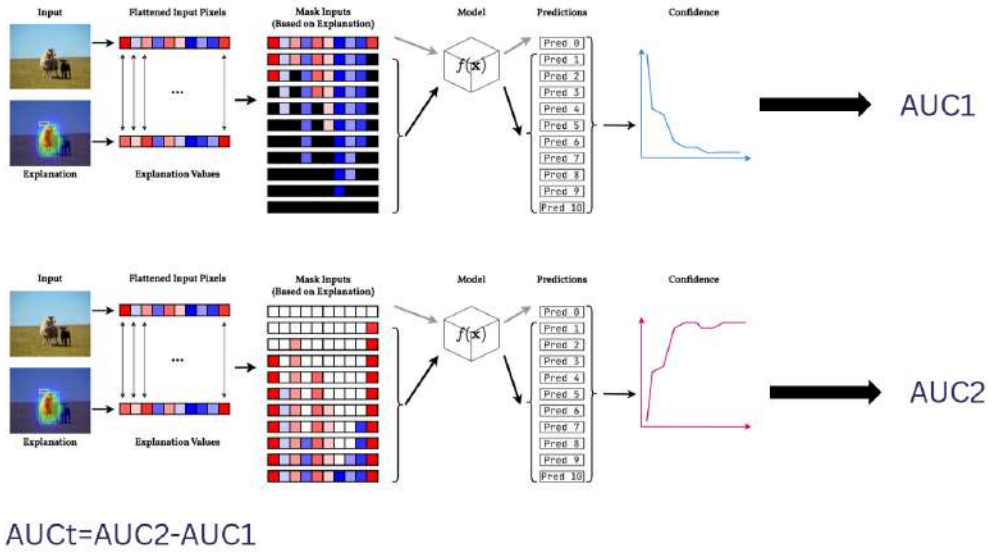
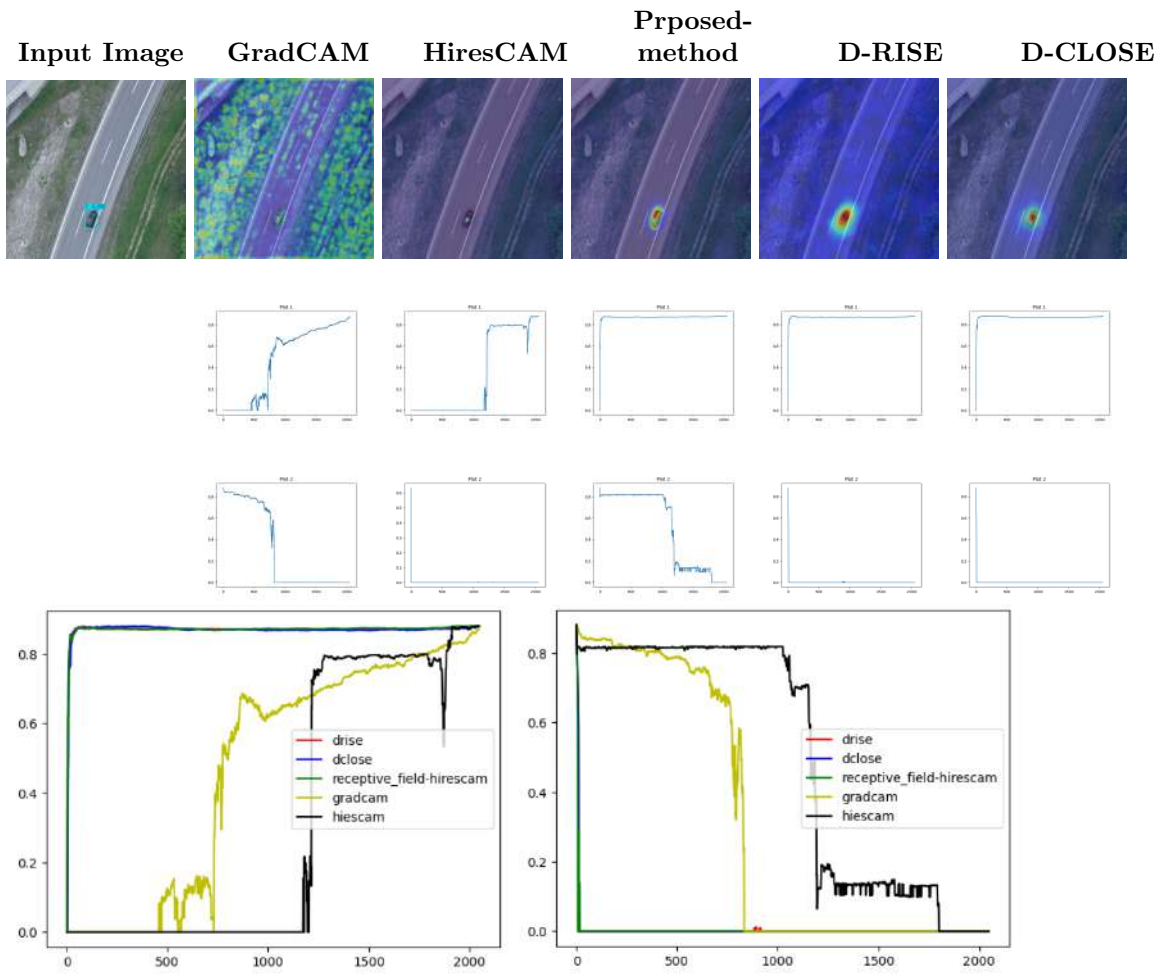


Figure 4.9: insertion-deletion metric

4.3.2 Qualitative and Quantitative evaluations

In the following results, all techniques—including both white-box and black-box methods—are evaluated using deletion-insertion metric. The second row of the figures presents the *insertion curves*, while the third row shows the corresponding *deletion curves*, illustrating how each explanation method affects the model's confidence as relevant or irrelevant regions are progressively added or removed.

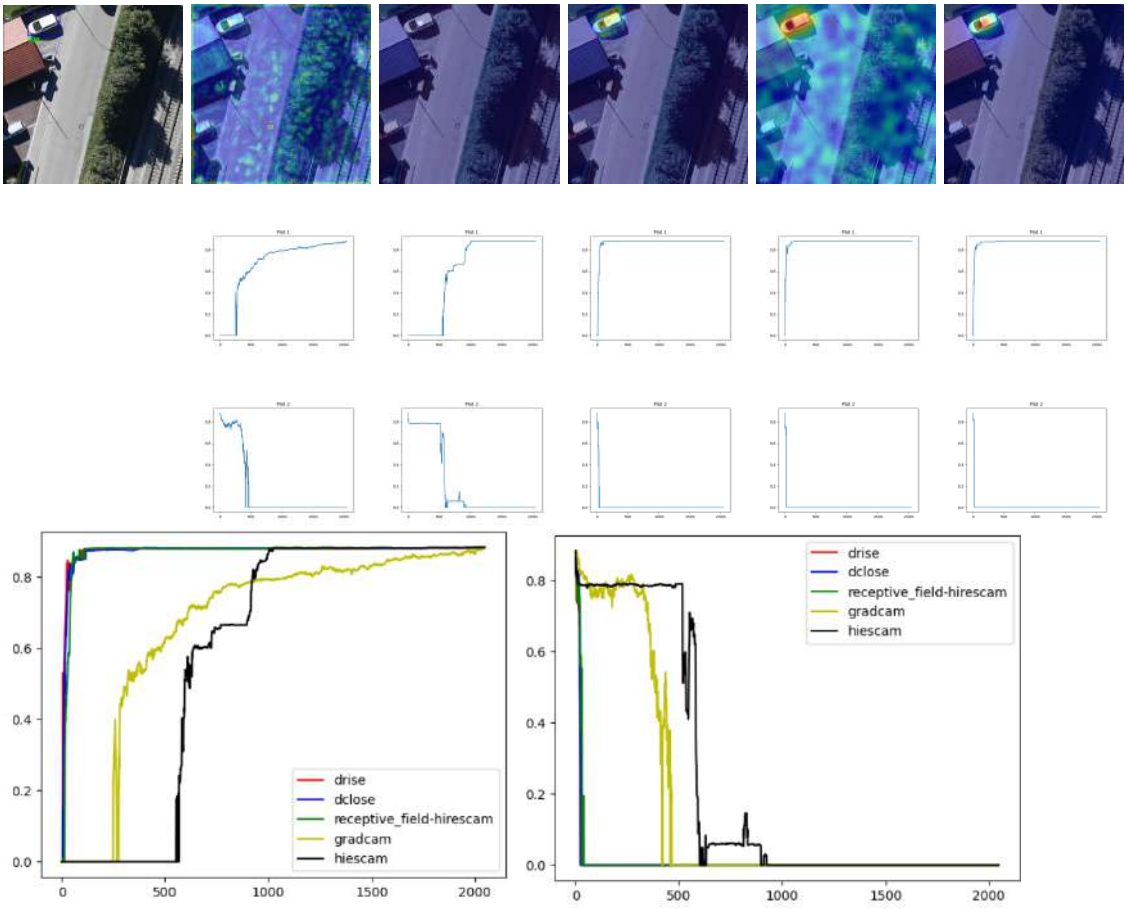
Tables 4.3 and 4.2 summarize the quantitative evaluation of each technique. Specifically, they report the *AUC(Insertion)*, *AUC(Deletion)*, and the difference *AUC(Insertion) – AUC(Deletion)*. Together, these metrics provide an indication of how effectively each method highlights the regions most influential for the detector's predictions, where higher insertion AUC and lower deletion AUC reflect stronger explanatory performance.



(a) Insertion–Deletion Comparison

	GradCAM	HiresCAM	Recptive-Field-HiresCAM	D-RISE	D-CLOSE
insertion	0.473	0.327	0.871	0.869	0.868
deletion	0.310	0.501	0.002	0.0037	0.0035
insertion-deletion	0.163	-0.17	0.868	0.865	0.864

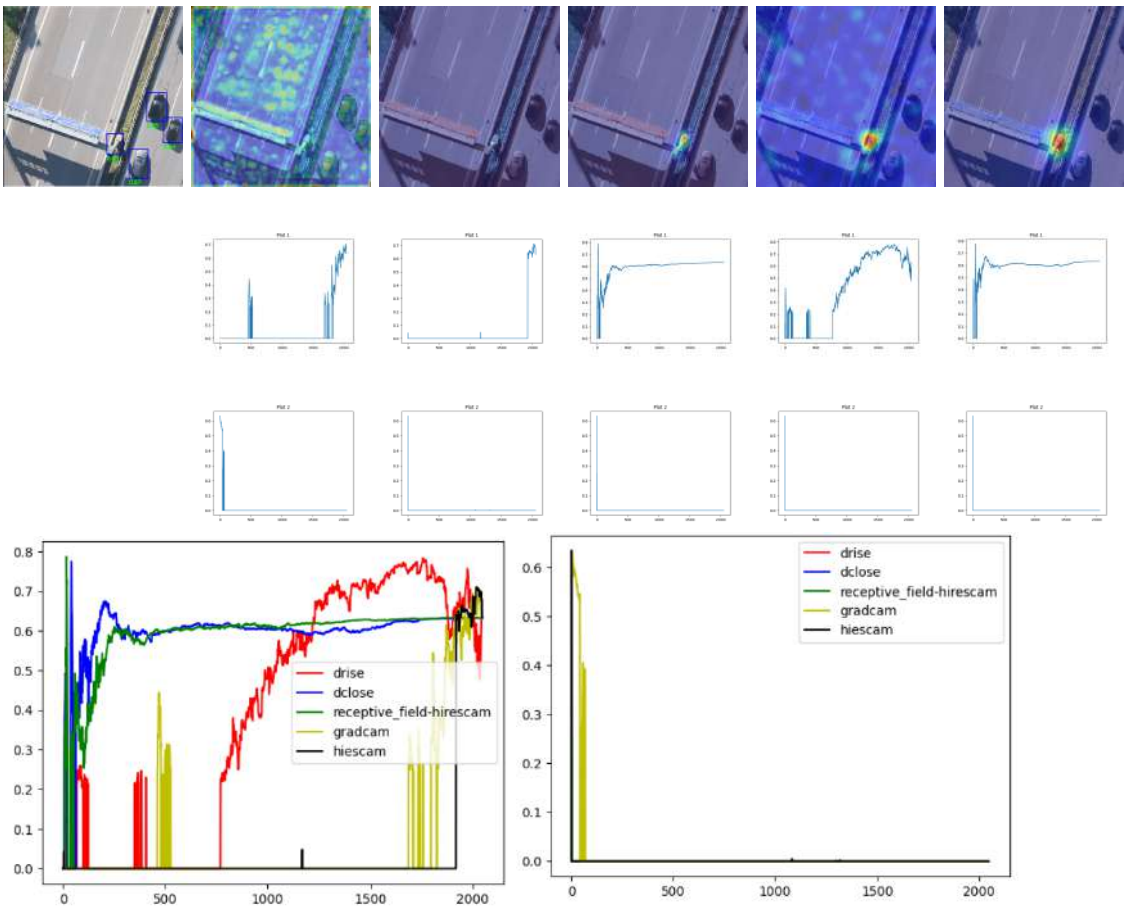
Table 4.1: Quantitative evaluation 1



(a) Insertion–Deletion Comparison

	GradCAM	HiresCAMN	Recptive-Field-HiresCAM	D-RISE	D-CLOSE
insertion	0.669	0.589	0.867	0.873	0.871
deletion	0.15	0.227	0.01	0.008	0.0084
insertion-deletion	0.362	0.857	0.857	0.865	0.863

Table 4.2: Quantitative evaluation 2



(a) Insertion–Deletion Comparison

	GradCAM	HiresCAMN	Recptive-Field-HiresCAM	D-RISE	D-CLOSE
insertion	0.076	0.04	0.589	0.393	0.595
deletion	0.01	0.00016	0.00027659	0.0001	0.00015446
insertion-deletion	0.06	0.04	0.589	0.393	0.595

Table 4.3: Quantitative evaluation 3

4.4 Advantages of the Proposed XAI Method: Overcoming Faithfulness–Efficiency Trade-offs

Explainable AI methods are traditionally constrained by a well-known trade-off between computational efficiency and explanation faithfulness. As discussed earlier, black-box approaches (D-RISE, D-CLOSE) achieve high faithfulness by densely sampling perturbed inputs and observing the model's output variations. While this perturbation-based reasoning provides strong causal evidence, it requires hundreds to thousands of forward passes for a single explanation, resulting in substantial computational and time costs. Conversely, gradient-based white-box techniques (e.g., HiresCAM, Grad-CAM) require only a single inference and a small number of backpropagations, making them far more efficient but typically less faithful: they often struggle to capture non-linear feature interactions, exhibit gradient saturation, and produce noisy or unstable heatmaps.

The proposed method (Receptive-Field-HiresCAM) overcomes this trade-off by combining the internal accessibility of white-box approaches with a more reliable estimate of feature importance that better reflects the underlying decision boundaries. Experimental evaluations demonstrate

that the proposed method achieves **significantly higher faithfulness than current white-box techniques**. This improvement is attributed to two key design choices: (i) The generation of the actual receptive field of each grid-cell rather than naively upsampling them (as the current xAI techniques do), and (ii) filtering the receptive field with gaussian-mask to make smoother and interpretable . These elements allow the method to approximate the causal effect of features more accurately without requiring external perturbations.

Furthermore,the proposed method retains the computational advantages of white-box techniques. It requires only a single forward pass and a fixed number of backward computations, regardless of input dimensionality. Unlike black-box perturbation methods, whose complexity scales linearly (and often exponentially) with the number of generated samples, the computational cost of the proposed approach remains constant. Empirical results show that it achieves explanation times that are an order of magnitude faster than black-box techniques while producing **explanations with comparable levels of faithfulness**.

In addition to its computational and faithfulness advantages, the method generates noticeably smoother and less noisy attribution maps. This reduction in noise enhances visual interpretability and produces explanations that align more closely with human perception. From a scientific standpoint, this improvement results from applying Gaussian-smoothing to the gradient information, which makes the explanation smoother and enforces local coherence.

Overall, the proposed XAI method successfully breaks the conventional trade-off between computational efficiency and explanatory faithfulness. It provides:

- white-box level computational efficiency,
- black-box level faithfulness,
- improved stability and reduced noise in the generated explanations.

This combination positions the method as a more practical and scientifically robust alternative to existing white-box and black-box XAI techniques.

Conclusion

Our method successfully breaks the efficiency-faithfulness trade-off. Quantitative evaluation using insertion-deletion metrics showed scores of 0.589–0.868, matching or exceeding black-box methods (D-RISE: 0.393–0.865, D-CLOSE: 0.595–0.864) while requiring only one forward pass plus fixed backpropagations—an order of magnitude faster. Traditional white-box methods performed poorly (Grad-CAM: 0.06–0.362; HiResCAM: -0.17 to 0.04). Qualitatively, the method produces smooth, interpretable explanations that accurately highlight vehicle features for true positives, reveal misleading similarities for false positives, and identify negatively-contributing regions for false negatives. The flexibility to explain any detection output component (confidence, width, height, position) enables deep localization error analysis. By achieving black-box-level faithfulness with white-box-level efficiency through principled receptive field reconstruction, this method enables practical deployment of explainable AI in safety-critical aerial object detection applications where understanding model decisions is essential for trust, accountability, and systematic improvement.

General Conclusion

This project successfully addressed the critical challenge of making aerial object detection models interpretable and trustworthy for safety-critical applications. Working with the EAGLE dataset of 215,986 annotated vehicles, we trained a YOLOv11l detector achieving strong performance ($\text{mAP}_{50} \approx 0.78$) and systematically investigated how to explain its predictions.

Our comprehensive evaluation of existing XAI techniques revealed a fundamental problem: black-box methods (D-RISE, D-CLOSE, SODEx) produce faithful explanations but require thousands of model runs per explanation, making them impractical for real-world use. White-box methods (Grad-CAM, HiResCAM) are computationally efficient but fail when applied to object detectors, producing spatially inaccurate visualizations that do not reflect the model's true reasoning.

To solve this, we developed **Receptive-Field-Based HiResCAM**, which explicitly reconstructs the true spatial regions each detection cell considers—rather than naively upsampling feature maps. Our method achieves the best of both worlds: explanation quality matching black-box methods (insertion–deletion scores: 0.589–0.868) with the computational efficiency of white-box approaches (single forward pass vs. thousands).

The practical impact is significant. Our explanations reveal that the detector relies on vehicle roofs, shadows, and contextual cues, explaining why false positives occur (misleading similar features) and why false negatives happen (unfamiliar contexts). These insights enable targeted dataset improvements and systematic model debugging.

This work demonstrates that faithful, efficient explainability for object detectors is achievable through principled architectural understanding. By breaking the conventional efficiency–faithfulness trade-off, we enable deployment of interpretable AI systems in mission-critical aerial surveillance applications where understanding model decisions is essential for trust, safety, and accountability.

Future directions include extending this approach to transformer-based detectors, multi-class scenarios, video object tracking, and integration with active learning frameworks for data-driven model improvement.

Bibliography

[1] Achanta, R., Shaji, A., Smith, K., Lucchi, A., Fua, P., & Süsstrunk, S. (2012). SLIC Superpixels Compared to State-of-the-Art Superpixel Methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(11), 2274–2282.

[2] Baumgartner, M., Glock, P., Bunk, J., Weinmann, M., & Hinz, S. (2021). EAGLE: Large-Scale Vehicle Detection Dataset in Aerial Imagery. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, V-3-2021, 31–38.

[3] Chattopadhay, A., Sarkar, A., Howlader, P., & Balasubramanian, V. N. (2018). Grad-CAM++: Generalized Gradient-Based Visual Explanations for Deep Convolutional Networks. *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, 839–847.

[4] Jocher, G., Chaurasia, A., & Qiu, J. (2024). Ultralytics YOLOv11. Retrieved from https://github.com/ultralytics/ultralytics

[5] Liu, K., Gao, X., & Liu, Q. (2021). HiResCAM: Faithful Location Representation in Visual Explanations of Deep Neural Networks. *arXiv preprint arXiv:2011.08891*.

[6] Lundberg, S. M., & Lee, S. I. (2017). A Unified Approach to Interpreting Model Predictions. *Advances in Neural Information Processing Systems 30 (NIPS 2017)*, 4765–4774.

[7] Petsiuk, V., Das, A., & Saenko, K. (2018). RISE: Randomized Input Sampling for Explanation of Black-box Models. *British Machine Vision Conference (BMVC)*.

[8] Petsiuk, V., Jain, R., Manjunatha, V., Morariu, V. I., Mehra, A., Ordonez, V., & Saenko, K. (2021). Black-box Explanation of Object Detectors via Saliency Maps. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 11438–11447.

[9] Ribeiro, M. T., Singh, S., & Guestrin, C. (2016). "Why Should I Trust You?": Explaining the Predictions of Any Classifier. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 1135–1144.

[10] Selvaraju, R. R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., & Batra, D. (2017). Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization. *2017 IEEE International Conference on Computer Vision (ICCV)*, 618–626.

[11] Shrikumar, A., Greenside, P., & Kundaje, A. (2017). Learning Important Features Through Propagating Activation Differences. *Proceedings of the 34th International Conference on Machine Learning (ICML)*, 3145–3153.

[12] Wang, H., Wang, Z., Du, M., Yang, F., Zhang, Z., Ding, S., Mardziel, P., & Hu, X. (2020). Score-CAM: Score-Weighted Visual Explanations for Convolutional Neural Networks. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 111–119.

[13] Wang, J., Zhou, C., & Li, W. (2022). D-CLOSE: Detector-Cascading Multiple Levels of Segments to Explain Object Detectors. *IEEE Transactions on Image Processing*, 31, 4491–4504.

[14] Zhang, J., Bargal, S. A., Lin, Z., Brandt, J., Shen, X., & Sclaroff, S. (2021). Top-Down Neural Attention by Excitation Backprop. *International Journal of Computer Vision*, 129, 1084–1102.

[15] Zhang, Y., Wang, C., Wang, X., Zeng, W., & Liu, W. (2023). SODEx: Surrogate Object Detector Explainer for Black-Box Object Detectors. *arXiv preprint arXiv:2303.10862*.

[16] Zeiler, M. D., & Fergus, R. (2014). Visualizing and Understanding Convolutional Networks. *European Conference on Computer Vision (ECCV)*, 818–833.

[17] Zhou, B., Khosla, A., Lapedriza, A., Oliva, A., & Torralba, A. (2016). Learning Deep Features for Discriminative Localization. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2921–2929.