UNIVERSIDADE FEDERAL DE VIÇOSA

Computer Vision Methods for Aerial and Ground-Based Farm Monitoring: Addressing the Labeled Data Scarcity Problem

Juliana Quintiliano de Oliveira Ferreira
Magister Scientiae

VIÇOSA - MINAS GERAIS
2025

JULIANA QUINTILIANO DE OLIVEIRA FERREIRA

Computer Vision Methods for Aerial and Ground-Based Farm Monitoring: Addressing the Labeled Data Scarcity Problem

Dissertation submitted to the Computer Science Graduate Program of the Universidade Federal de Viçosa in partial fulfillment of the requirements for the degree of *Magister Scientiae*.

Adviser: Michel Melo da Silva

Co-adviser: Thiago Luange Gomes

VIÇOSA - MINAS GERAIS
2025

Ficha catalográfica elaborada pela Biblioteca Central da Universidade Federal de Viçosa - Campus Viçosa

T
F383c
2025

Ferreira, Juliana Quintiliano de Oliveira, 1987-
Computer vision methods for aerial and ground-based farm monitoring: addressing the labeled data scarcity problem / Juliana Quintiliano de Oliveira Ferreira. – Viçosa, MG, 2025.
1 dissertação eletrônica (90 f.): il. (algumas color.).

Texto em inglês.
Orientador: Michel Melo da Silva.
Dissertação (mestrado) - Universidade Federal de Viçosa, Departamento de Informática, 2025.
Referências bibliográficas: f. 80-90.
DOI: https://doi.org/10.47328/ufvbbt.2025.829
Modo de acesso: World Wide Web.

1. Inteligência artificial. 2. Visão por computador. 3. Processamento de imagens. 4. Fazendas. I. Silva, Michel Melo da, 1990-. II. Universidade Federal de Viçosa. Departamento de Informática. Programa de Pós-Graduação em Ciência da Computação. III. Título.

CDD 22. ed. 006.3

Bibliotecário(a) responsável: Bruna Silva CRB-6/2552

JULIANA QUINTILIANO DE OLIVEIRA FERREIRA

Computer Vision Methods for Aerial and Ground-Based Farm Monitoring: Addressing the Labeled Data Scarcity Problem

Dissertation submitted to the Computer Science Graduate Program of the Universidade Federal de Viçosa in partial fulfillment of the requirements for the degree of *Magister Scientiae*.

APPROVED: November 24, 2025.

Assent:

Juliana Quintiliano de Oliveira Ferreira
Author

Michel Melo da Silva
Adviser

ACKNOWLEDGMENTS

I thank God for opening this door for me. He has been my refuge and strength at all times.

To my dear husband, Rogério, for his unconditional support, patience, and encouragement. To my little princess, Laís, for inspiring me every day to be stronger and better.

To my mother, Wilma, and my entire family, who have always believed in me and celebrated each step of this journey.

To my dear friends, who listened to my frustrations. Their support made the difficult days lighter and encouraged me to keep moving forward.

To my brothers and sisters in Christ for their prayers and spiritual support, which strengthened me along the way.

To my advisor, Professor Michel Melo da Silva, and my co-advisor, Professor Thiago Gomes Luange, for their guidance, support, and shared knowledge throughout this journey.

This study was partially funded by the National Council for Scientific and Technological Development (CNPq), under project number 409109/2021-5.

This work has been sponsored by the following Brazilian research agencies: Coordination for the Improvement of Higher Education Personnel (CAPES; Financing code 001), Minas Gerais State Foundation for Research Aid (FAPEMIG) and National Council of Scientific and Technological Development (CNPq).

Taste and see that the LORD is good; blessed is the one who takes refuge in him."
Psalm 34:8

ABSTRACT

FERREIRA, Juliana Quintiliano de Oliveira, M.Sc., Universidade Federal de Viçosa, November, 2025. **Computer Vision Methods for Aerial and Ground-Based Farm Monitoring: Addressing the Labeled Data Scarcity Problem**. Adviser: Michel Melo da Silva. Co-adviser: Thiago Luange Gomes.

The monitoring of agricultural areas is essential to ensure a safe environment, avoid economic losses, and prevent risks to infrastructure and human safety. Effective monitoring can detect lost animals, unauthorized human access, wild animal intrusions, among other issues. Artificial Intelligence (AI) is a powerful tool to automate this process through the processing of images captured from ground or aerial perspectives. However, supervised models require large volumes of labeled data to achieve good performance, and although data exists, most of it is private and inaccessible to the public or available in limited quantity. The available datasets are generally unlabeled or do not cover domain-specific scenarios, such as rural and farm environments. In addition, data collection in agricultural settings presents further challenges, such as the need for drones or other remote sensing technologies, making the process more expensive and complex. Finally, the annotation stage is also a bottleneck, as beyond collecting images, intensive manual work is required to label them, increasing both cost and time. In this context, we propose to address the problem of data scarcity and annotation complexity from two perspectives, aiming to mitigate the bottleneck of training AI models when only a small amount of labeled data is available. The first study focuses on aerial monitoring using images collected by Unmanned Aerial Vehicles (UAVs) on farms for the task of semantic segmentation. Semantic segmentation brings significant benefits to agricultural monitoring by automatically identifying and differentiating important elements of the rural environment, such as vegetation areas, bodies of water, and buildings. By precisely mapping these elements, the technique enables the identification of risk situations for livestock and infrastructure, contributing to safer and more efficient farm management. Thus, we investigated pre-training strategies using synthetic data from the same domain and real data from slightly different domains. We then fine-tuned on the target dataset, and the quantitative and qualitative results demonstrated that pre-training with the synthetic dataset achieved better final performance, leading to an increase of 3.1 p.p. in IoU, 6.4 in F1-Score, and 7.5 in Recall compared to the cross-domain real-image pre-training strategy. In the second study, we focus on object detection using ground-level images similar to security camera footage. This task is important in

agricultural monitoring because it allows automatic identification and localization of animals and people, supporting security, livestock management, and the tracking of activities on the farm. To address the data scarcity challenge in this scenario, we proposed a method to effectively use multiple datasets even when they do not share the same classes, ensuring comprehensive coverage of all required categories. The proposed SmartClass methodology achieved more robust and adaptable detection approaches suitable for agricultural environments, with significant increases in Recall, mAP50, and mAP50-95 metrics compared to models trained without the methodology, thus demonstrating improved efficiency and reliability of the model.

Keywords: artificial intelligence; computer vision; farm monitoring.

RESUMO

FERREIRA, Juliana Quintiliano de Oliveira, M.Sc., Universidade Federal de Viçosa, novembro de 2025. **Métodos de Visão Computacional para Monitoramento de Fazendas Aéreo e Terrestre: Abordando o Problema da Escassez de Dados Rotulados**. Orientador: Michel Melo da Silva. Coorientador: Thiago Luange Gomes.

O monitoramento de áreas agrícolas é essencial para manter o ambiente seguro, evitar perdas econômicas e prevenir problemas com a integridade física dos locais e das pessoas. Um bom monitoramento pode detectar animais perdidos, entradas humanas não autorizadas, invasões de animais selvagens, dentre outros problemas. A Inteligência Artificial (IA) é uma grande aliada para automatizar esse processo por meio de processamento de imagens capturadas, sejam imagens terrestres ou aéreas. No entanto, modelos supervisionados requerem grandes volumes de dados rotulados para alcançar bom desempenho, e, apesar de existirem dados, a maior parte deles são privados e inacessível ao público ou em quantidade reduzida. Os conjuntos de dados disponíveis geralmente não são anotados ou não contemplam domínios específicos, como ambientes rurais e fazendas. Além disso, a coleta de dados em cenários agrícolas apresenta desafios adicionais, como a necessidade de uso de drones ou outras tecnologias de sensoriamento, o que torna o processo mais caro e complexo. Por fim, a etapa de rotulagem também representa um gargalo, pois além da coleta das imagens, é necessário um trabalho manual intensivo para anotá-las, elevando ainda mais o custo e o tempo necessários. Pensando nisso, propomos abordar o problema da escassez de dados e da dificuldade do processo de anotação com duas vertentes a fim de lidar com o gargalo de treinar modelos de IA quando existe um número baixo de dados rotulados. O primeiro estudo foca no monitoramento aéreo com imagens coletadas por Veículos Aéreos não Tripulados (VANT's) em fazendas na tarefa de segmentação semântica. A tarefa de segmentação semântica traz grandes benefícios ao monitoramento agrícola, pois permite identificar e diferenciar automaticamente elementos importantes do ambiente rural, como áreas de vegetação, corpos d'água e construções. Ao mapear com precisão esses elementos, a técnica possibilita a identificação de situações de risco ao gado e à infraestrutura, contribuindo para uma gestão mais segura e eficiente da fazenda. Assim, investigamos estratégias de pré-treinamento usando dados sintéticos no mesmo domínio e também dados reais em domínios ligeiramente diferentes. Em seguida realizamos ajuste fino no conjunto de dados alvo e os resultados quantitativos e qualitativos demonstraram que o pré-treinamento usando o conjunto de dados

sintéticos teve melhor desempenho no treinamento final, levando a um aumento de 3,1 p.p. em IoU, 6,4 em F1-Score e 7,5 em Recall quando comparado à estratégia de pré-treinamento com imagens reais em domínio cruzado. Para o segundo estudo, focamos na detecção de objetos com imagens terrestres, semelhantes a imagens de câmeras de segurança. Essa tarefa é importante no monitoramento agrícola porque permite identificar e localizar automaticamente animais e pessoas, auxiliando na segurança, no manejo do gado e no acompanhamento das atividades na fazenda. Neste caso, para tratar o gargalo da falta de dados, propusemos um método para utilizar efetivamente vários conjuntos de dados, mesmo quando eles não têm as mesmas classes, garantindo uma cobertura abrangente de todas as categorias necessárias. A metodologia SmartClass proposta alcançou abordagens de detecção mais robustas e adaptáveis, adequadas para ambientes agrícolas, com aumentos consideráveis nas métricas de Recall, mAP50 e mAP50-95 em comparação com modelos treinados sem a metodologia, demonstrando assim um ganho na eficiencia e confiabilidade no modelo.

Palavras-chave: inteligência artificial; visão computacional; monitoramento de fazendas.

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS AND ACRONYMS

ACF Aggregate Channel Features. 45

AI Artificial Intelligence. 15, 18–21, 23, 25, 26, 38, 41, 43–46, 48, 49, 56, 58, 64, 77, 78

AP50 Average Precision at an IoU threshold of 0.50. 23, 70, 71, 73–75

AP50–95 Average Precision across IoU thresholds from 0.50 to 0.95. 23, 70, 71, 73–75

BoxP Bounding Box Precision. 70, 73, 74

CDNTS Corner Detection and Nearest Three-Point Selection. 41

CNN Convolutional Neural Network. 9, 25, 26, 30, 45

COCO Common Objects in Context. 18, 19, 22, 28–31, 33, 36, 68–71, 75, 77, 78

DA Domain Adaptation. 10, 39, 40, 47

DCED Dual-Canny Edge Detection. 42

DeiT Data-efficient Image Transformers. 26

DETR Detection Transformer. 26, 30

DPM Deformable Parts Model. 45

FCN Fully Convolutional Network. 34

FN False Negative. 50

FP False Positive. 50

HRNet High-Resolution Network. 9, 36, 37, 42, 56

HRNet.OCR High-Resolution Network - Object-Contextual Representations. 36–38, 49, 56, 58

IMOWAD Induced Minkowski Ordered Weighted Averaging Distance. 42

ISPRS International Society for Photogrammetry and Remote Sensing. 54

MLDG Meta-learning for domain generalization. 40

MS Microsoft. 28

NLP Natural Language Processing. 26

NREC National Robotics Engineering Center. 18, 19, 45, 68, 69, 71, 77, 78

OCR Object-Contextual Representations. 9, 37, 38, 56

OPEDD Off-Road Pedestrian Detection Dataset. 45

PEFT Parameter-efficient fine-tuning. 40

R-CNN Region-based Convolutional Neural Network. 30, 45

RPN Region Proposal Network. 30

SAM Segment Anything Model. 42

SDS Simultaneous Detection and Segmentation. 34

SS SyntheticSwitzerland Dataset. 9, 53, 55, 59, 61

SSD Single Shot MultiBox Detector. 30

TN True Negative. 50

TP True Positive. 50

UAV Unmanned Aerial Vehicle. 15, 17–23, 28, 35, 36, 41, 42, 46–49, 51–54, 56, 59, 77

UDA Unsupervised Domain Adaptation. 40

ViT Vision Transformer. 9, 26, 27

YOLO You Only Look Once. 9, 28–34, 44, 67–71, 75, 76

CONTENTS

Chapter 1

Introduction

Farm and agricultural monitoring play an important role in ensuring the safety of people, animals, and crops. This monitoring can be implemented through physical barriers, such as electric fences, or through image-based surveillance. Images can be collected in several ways, including ground-based cameras installed at fixed locations or aerial cameras mounted on Unmanned Aerial Vehicle (UAV) that fly over areas of interest. Both approaches generate a large volume of images and therefore require Artificial Intelligence (AI) to extract relevant information and support decision-making for farm owners. Manual human analysis of such a large number of images is impractical and may even fail to detect important events.

In the analysis of images using AI, object detection and semantic segmentation tasks play a key role in monitoring activities. These techniques contribute directly to farm security by enabling the precise mapping of vegetation, rural structures, and potential risk areas through aerial imagery, while ground-based detection systems can automatically identify animals, people, and vehicles near sensitive zones. Such capabilities help prevent livestock losses, detect unauthorized access, and strengthen overall farm surveillance. Additionally, these technologies, particularly object detection, support the development of digital fences, which are virtual barriers that allow environmental monitoring through image analysis and alert-generation algorithms, complementing or even replacing physical fences.

However, for AI to be effectively implemented in this area, we need a large amount of labeled data to train supervised models, whether for object detection or semantic segmentation task. Publicly available datasets with multiple labeled images in these environments are scarce due to the difficulties and human cost of data creation. To address this bottleneck of annotated data, our methodology addresses two studies: the first, addressing aerial UAV monitoring for semantic segmentation, and the second, addressing ground-based surveillance for object detection.

Below we will discuss in more detail the motivations, challenges, proposal, research questions, and contributions of this work.

Figure 1.1: Examples of security failures on farms. Images adapted from sources [1][2][3][4].

1.1 Motivation

Monitoring agricultural areas and rural properties is crucial for maintaining sound and sustainable economic management. This action allows for the early detection of critical events, enabling timely corrective actions. Furthermore, monitoring contributes to the safety of people, animals, and crops (Naveenkumar et al., 2022). A well-conducted monitoring task can alert people to unauthorized human entry, animal strays to inappropriate locations, and detect infrastructure failures (*e.g.*, broken fences) (Silva et al., 2024). It can also prevent wildlife intrusion, which is important for preserving not only human life but also wildlife conservation (Shetty and Ashwath, 2023; Mamat et al., 2022). Examples of problems faced in rural environments can be seen in Figure 1.1. The first row shows images of animals entering roads, posing risks to people traveling on these routes. The second row illustrates animals attacking farm property: the first image shows an animal attacking chickens, and the second shows wild boars destroying crops.

[1] https://www.tjmg.jus.br/portal-tjmg
[2] https://www.to.gov.br/
[3] https://www.bahiaja.com.br
[4] https://www.agronovas.com.br/destruidores-de-plantacao/

Figure 1.2: Examples of equipment used for farm and agricultural security. The first image shows a drone used for monitoring agricultural areas, and the second image shows a security camera for protecting facilities. Images adapted from sources [1][2].

In this scenario, UAV emerge as valuable tools for digital agricultural monitoring due to their maneuverability, battery efficiency, broad coverage, and ability to capture high-resolution images. Husain et al. (2022) note that UAVs, such as drones, can collect real-time images and sensor data, assisting farmers in decision-making, supporting crop protection, and contributing to early warning systems. Another equipment that can be used as a ground-based technique for agricultural and farm monitoring are security cameras. They can be used strategically, installed at points where people and vehicles frequently pass through, as well as in specific areas of corrals and plantations. Figure 1.2 illustrates typical examples of such devices in rural environments. The first image shows a drone used for agricultural monitoring, while the second depicts a security camera installed to safeguard farm facilities and perimeter areas.

1.2 Challenges

Agricultural environments have unique characteristics that differentiate them from other contexts, such as urban areas (Pezzementi et al., 2017). Farm images often pose greater difficulty for algorithms due to the potential for people and animals to be camouflaged among vegetation or along farm roads. The distinct visual characteristics of farm areas, dominated by vegetation or open land, contrast sharply with the urban backdrop of buildings, houses, and asphalt, further complicating detection efforts (Neigel et al., 2020). Another characteristic is that the vastness and open nature of rural environments stand out, posing substantial challenges for monitoring. In such case, effective human surveillance may even be impractical due to physical barriers such as fences, dense forest areas, water channels, and uneven or steep terrain. This reinforces the need to seek

[1] https://www.mondomacchina.it/en/drone-bee-event-flying-agriculture-c2431
[2] https://opresenterural.com.br/

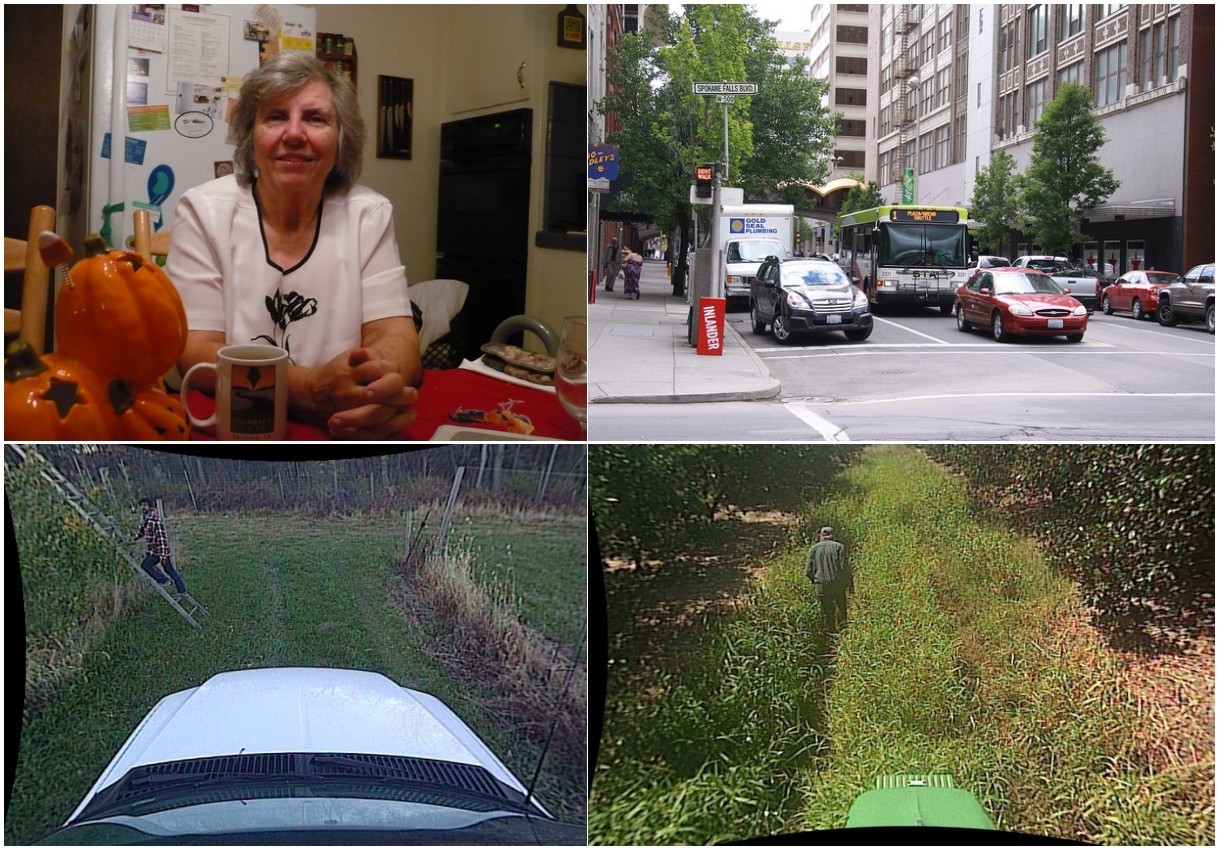

Figure 1.3: This figure demonstrates the difference in images between generic datasets and specific datasets in rural environments. Images adapted from Common Objects in Context (COCO) (Lin et al., 2014a) and National Robotics Engineering Center (NREC) (Pezzementi et al., 2017).

solutions focused on this context. G et al. (2025) note that electric fences cannot solve the problem of protecting these areas and also put the lives of animals at risk.

The deployment of drones and security cameras for agricultural monitoring can be excellent tools to help address this problem; however, their use alone does not solve the issue, as these devices only collect images with a large volume of data. Performing analysis manually is time-consuming and labor-intensive. With this large volume of images, data analysis becomes a bottleneck. To address this problem, there is growing interest in applying AI techniques to automate the interpretation of UAV and security camera images. AI has demonstrated great potential in several agricultural tasks, including crop monitoring, irrigation management, fire detection, pest control, and increased farm security (Santos et al., 2024; Nazeer et al., 2024).

Although previous studies have used AI for agricultural and rural environments, there is still a significant gap related to data annotation and model generalization for agricultural security applications. Supervised AI models require large amounts of high-quality labeled data (Liu et al., 2024). However, data annotation remains a bottleneck for supervised models because there is insufficient data in the context of

farms and agriculture, and many datasets are small or not in the public domain. This task is inherently expensive due to the need for many hours of human labor to perform it, and often the data collection locations are also difficult to access (Whitney and Norman, 2024). Furthermore, some AI tasks, such as semantic segmentation, require pixel-level annotation (Ge et al., 2024). Other tasks, such as object detection, require labeling with bounding boxes, indicating the object and the class to which it belongs.

Agricultural environments present unique characteristics, such as a greater presence of animals and predominantly green backgrounds with crops, as well as camouflaged individuals. Consequently, models trained on broad and generic datasets, such as COCO (Lin et al., 2014a), often fail to generalize well in these scenarios. Figure 1.3 illustrates differences between environments. The first row shows images from the COCO dataset (Lin et al., 2014a), featuring predominantly gray backgrounds in urban areas and indoor scenes. The second row presents images from an agricultural dataset (NREC (Pezzementi et al., 2017)), characterized by predominantly green or dry foliage backgrounds.

1.3 Proposal

When dealing with a task with a limited amount of labeled data, labeled data from other domains can be used for model pre-training. In this case, the ideal is to find data in domains that are as similar as possible. Another approach to address the limited availability of labeled data is the use of synthetic datasets. Virtual environments allow for the large-scale generation of labeled images without the need for real-world drone flights and manual labeling tasks, reducing costs (Silva et al., 2024). However, models trained exclusively with synthetic data or with data from different domains often fail to generalize effectively when applied directly to real-world data (Liu et al., 2024). Synthetic data often fails to accurately represent subtle but important details of the real world (Miletic and Sariyar, 2024), such as lighting and even real sensor noise. Another way to address this bottleneck is to aggregate multiple existing public datasets that contain the desired similarities and classes. This way, small datasets, when combined, can become a single dataset with a sufficient number of images. However, integrating different datasets is a complex task (Zhou et al., 2022), as most datasets use different annotation formats and define different classes.

In this work, we conduct two complementary studies on AI models to address the bottleneck caused by limited labeled data: one focusing on the semantic segmentation of aerial UAV imagery, and the other on object detection using ground-based images. These tasks are directly related to farm security because semantic segmentation enables the precise identification and mapping of rural structures, vegetation, and risk areas from aerial perspectives. Meanwhile, object detection in ground-based imagery allows

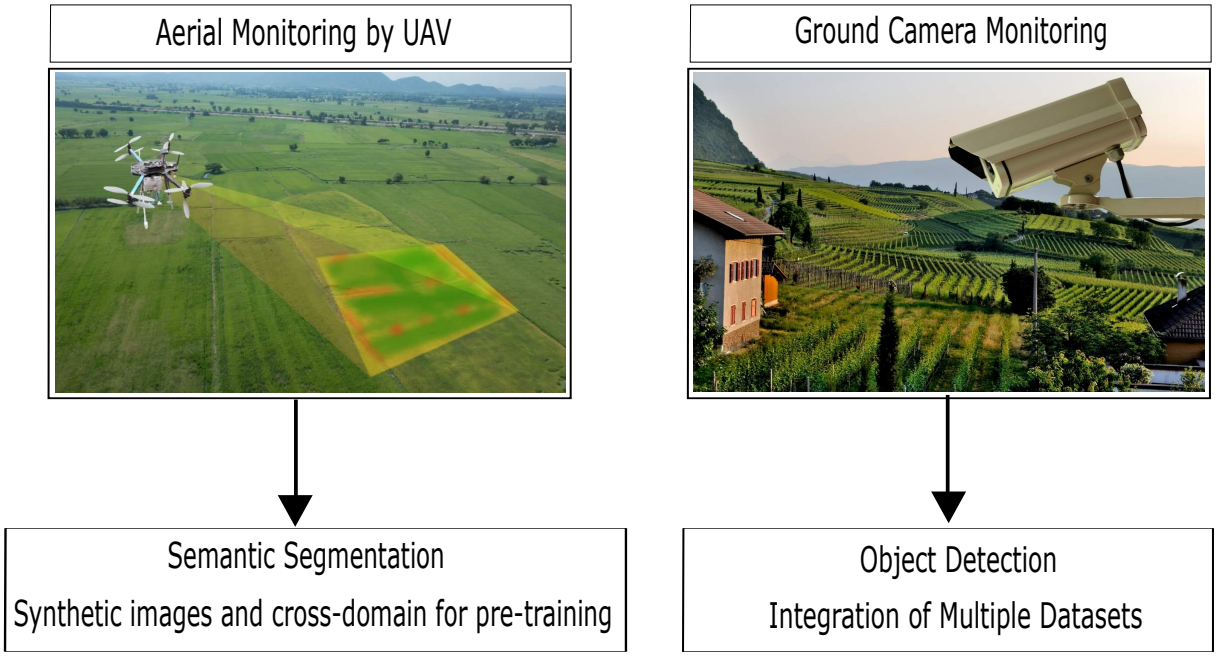

Figure 1.4: An overview of the two research fronts: the first, Aerial Monitoring by UAV, and the second, Ground Camera Monitoring. Images adapted from Radojcic and Cvetković (2023) and Rota Vigilância Patrimonial[1].

the system to automatically recognize animals, people, and vehicles near critical farm areas, helping prevent livestock loss, detect unauthorized access, and improve overall farm surveillance. Furthermore, these tasks, especially object detection, can assist in the creation of digital fences. A digital fence is a possibility of monitoring the environment without physical fences, using only images and alert-generating algorithms.

For aerial UAV monitoring, we compare real data from slightly different domains with synthetic data from the same domain as pre-training strategies. We then validate the effectiveness of these pre-training approaches when applied to a target dataset consisting of a limited number of real-world labeled images, representing UAV flight scenarios over rural areas.

The second study focuses on the challenges of aggregating multiple labeled object detection datasets to support the creation of digital fences using security camera imagery. To this end, we propose a methodology for integrating heterogeneous datasets that handles annotation inconsistencies across sources, establishing a broadly applicable approach. Additionally, we address the issue of missing annotations for certain classes within a dataset to prevent bias or degradation in model training.

The general objective of this work is to investigate and propose strategies to mitigate the scarcity of labeled data for AI models applied to the monitoring and security of rural and agricultural environments. The two approaches can be seen in the Figure 1.4.

To achieve this goal, we define the following specific objectives:

[1] https://www.rotavigilanciapatrimonial.com.br

- Evaluate the effectiveness of synthetic data and cross-domain real data as pre-training strategies for semantic segmentation of UAV imagery.
- Assess how these pre-training strategies impact model generalization on a target dataset containing a limited number of real UAV images.
- Develop a methodology to integrate multiple heterogeneous labeled datasets for object detection in farm and rural contexts.
- Propose mechanisms to mitigate the impact of missing annotations across datasets, enabling more consistent learning of relevant classes.
- Demonstrate the practical applicability of the proposed methods to support digital fence systems for farm security.

1.4 Research Questions

Given the challenges and proposal previously described and the scarcity of labeled data in agricultural security scenarios, this work is guided by the following research questions:

- How does pretraining with real-world data from slightly different domains compare to training with synthetic data from the same domain when the target dataset contains only a small number of labeled images?
- Is it possible to integrate heterogeneous object-detection datasets from rural-like environments in a way that preserves class learnability and avoids degrading model performance?

1.5 Contributions

Capturing images from UAVs or ground-based cameras alone does not solve the monitoring problem in agricultural areas and farms. Data analysis is essential, but given the large volumes of information and the need for real-time processing, AI capabilities become essential allies.

Training supervised AI models, particularly in domain-specific contexts, demands substantial amounts of annotated imagery. Although large volumes of data are being generated every day, most of it remains private, restricted, or insufficient for open research. Public datasets are often limited, unlabeled, or fail to represent real-world scenarios found in rural and agricultural environments. Additionally, acquiring data in farm settings introduces practical challenges, such as the need for drones or other remote-sensing equipment, which increases operational cost and complexity. Beyond

data collection, annotation represents another major bottleneck, requiring extensive manual effort and consequently driving up time and financial cost. To mitigate the bottleneck of manual labeling, this work proposes two fronts of action:

1. **Synthetic and cross-domain real-world pre-training study for aerial surveillance with semantic segmentation:** In this front, we discuss the use of synthetic data compared to real-world data from a slightly different domain as pre-training strategies, followed by training on the target dataset consisting of a limited set of labeled real-world data from scenarios involving UAVs flying over rural areas.

   This study investigates how to improve segmentation performance on real-world data while mitigating the bottleneck associated with the need for large datasets. To enable a fair comparison, we searched the literature for an available labeled dataset consisting of UAV images in farm-like areas, which is our target dataset for the semantic segmentation task.

   To define the target dataset, it is crucial to find a synthetic dataset in the same domain, specifically UAV images of farm-like areas, and the images from both datasets must be similar. After identifying the target and synthetic datasets, we searched for a visually similar domain. In this case, we chose Remote Sensing due to its superior vision (top-down perspective) and the vast availability of labeled data.

   Parts of this study were published and presented at the 38th SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI 2025). The code and data were made available through the link https://github.com/MaVILab-UFV/A_Study_of_Synthetic_and_Cross_Domain_SIBGRAPI_2025.

2. **Integration of ground-based monitoring datasets with similar security cameras to support the creation of digital fences in the object detection task:** In this work, we propose the development of a model capable of assisting in the monitoring of farms through the detection of important objects such as animals and people. This feature will assist in the creation of digital fences. Digital fences aim to automatically detect the presence of people, potentially dangerous wildlife that may pose a threat to humans and livestock, as well as other animals that may stray into inappropriate areas. Unlike most existing object detection models trained on the COCO dataset for example, our approach focuses on improving object detection specifically for rural environments, where objects may be camouflaged in the landscape.

   Therefore, given the scarcity of labeled data in this environment, we propose integrating multiple smaller datasets to achieve broader data coverage. However,

simply merging these datasets is not sufficient, as each one uses different labeling schemes and class definitions. To address this challenge, we introduce an approach that harmonizes the data and enables effective model training and performance.

To this end, we developed a set of methods called SmartClass, designed to train models to minimize the impact of missing annotations for certain classes in some datasets, preserving the learnability of these classes.

Parts of this study were presented at the XIX Computer Vision Workshop (WVC 2024) and published in the RITA Journal. The code and data were made available through the link https://github.com/MaVILab-UFV/Digital-Fencing-WVC-2024.

Both studies demonstrated good results in addressing the scarcity of labeled data for training AI models. In the first approach, which explored pre-training strategies using synthetic data, cross-domain data, and their combination, the results of the performed comparison demonstrate, both qualitatively and quantitatively, that using synthetic data in the pre-training stage outperforms the pre-training strategy using real data from a different domain by **3.1 p.p.** in IoU, **6.4 p.p.** in F1-Score, and **7.5 p.p.** in Recall. The effectiveness of the pre-training stage, followed by training on the target dataset with a small number of real images, improves the results of the semantic segmentation task in UAV-based farm monitoring images, thus contributing to advances in AI for aerial monitoring of farms and agricultural fields.

In the second study, the strategy of combining different labeled datasets, called SmartClass, for the object detection task demonstrated improved metric values and superior qualitative outcomes. The proposed SmartClass methodology achieved more robust and adaptable detection approaches, suitable for agricultural environments, with considerable increases in Recall, Average Precision at an IoU threshold of 0.50 (AP50), and Average Precision across IoU thresholds from 0.50 to 0.95 (AP50–95) metrics compared to models trained without the methodology.

The two studies complement each other in supporting farm monitoring, providing coverage for both large-scale aerial observation and specific ground-level points through the use of terrestrial cameras. The results demonstrate effective strategies to address the scarcity of labeled data in both semantic segmentation and object detection tasks.

1.6 Document Organization

The document is organized as follows. Chapter 2 presents the theoretical framework that underpins this study, while Chapter 3 discusses related work. As mentioned earlier, this research comprises two complementary studies, which are presented in separate chapters. Chapter 4 focuses on Semantic Segmentation – Synthetic Images and

Cross-Domain for Pre-training, describing the methodology, experimental setup, and results. Chapter 5 addresses Object Detection – Integration of Multiple Datasets, following a similar structure by detailing the methodology, experiments, and results. Finally, Chapter 6 presents the general conclusions of this study, summarizing the main findings of each approach.

Chapter 2

Theoretical Background

Monitoring agricultural areas is essential to ensure safety, animal welfare, and efficient resource management. Automating this process through image analysis allows for real-time detection of events such as the presence of animals in risk areas, human intrusion, and environmental changes. In this context, AI plays a fundamental role, and tasks such as object detection and semantic segmentation become highly relevant, as they allow for the identification and location of animals, people, vehicles, and structures, as well as the precise mapping of the rural landscape. Therefore, understanding the techniques, methods, and problem formulations, such as object detection and semantic segmentation, as well as the models employed, is essential to contextualize the problem addressed in this work and to justify the adopted approach.

2.1 Convolutional Neural Networks (CNNs)

Convolutional Neural Network (CNN) emerged from the work of Lecun et al. (1998), who introduced LeNet-5 to solve problems of handwritten digit recognition, especially in the context of automatic reading of bank checks. This work marked a turning point in AI applied to image recognition, establishing the basis for modern architectures used to this day. From this milestone, CNNs have been widely used in various computer vision tasks, such as image classification (Krizhevsky et al., 2017), object detection (Girshick et al., 2014), semantic segmentation (Long et al., 2015), facial recognition (Taigman et al., 2014), tracking (Nam and Han, 2016), and other applications in areas such as medicine, agriculture, security, and autonomous vehicles.

The main flow of a CNN architecture involves the successive application of convolutional layers, responsible for extracting visual patterns; pooling layers, which reduce the spatial dimension and increase robustness to variations; and fully connected layers, used to consolidate the learned features and produce the final prediction of the model. This sequence can be observed in the LeNet architecture shown in Figure 2.1, where convolutional layers extract hierarchical features, subsampling (average pooling) layers progressively downsample the feature maps, and the final fully connected layers transform the extracted representations into class scores.

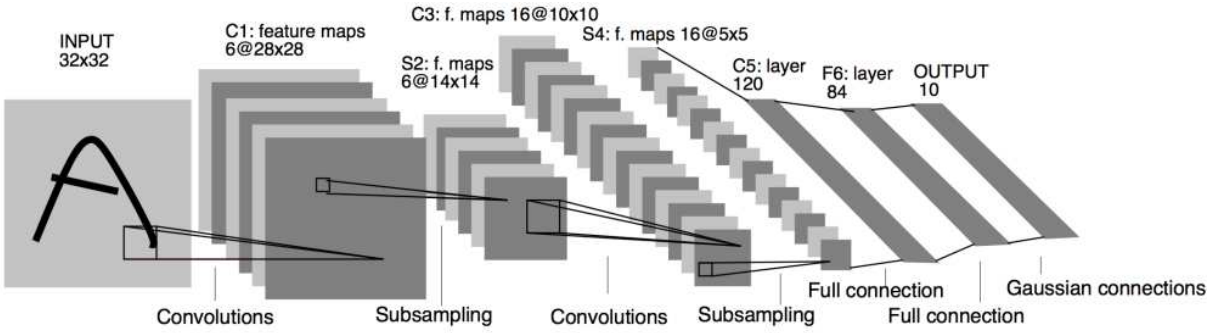

Figure 2.1: Example of a CNN architecture. LeNet architecture with convolutional layers for feature extraction, subsampling (average pooling) layers for spatial reduction, and fully connected layers for classification. Source: Adapted from Lecun et al. (1998).

2.2 Transformers

The Transformer is an architecture originally proposed to solve problems related to Natural Language Processing (NLP) (Vaswani et al., 2017), but it has since been extended to the field of computer vision due to its approach based exclusively on attention mechanisms. These mechanisms are capable of capturing long-range relationships between elements and modeling global dependencies more effectively than traditional convolutional architectures.

Currently, there are hybrid architectures that merge convolutional layers with attention modules, such as the Detection Transformer (DETR) (Carion et al., 2020), which employs convolutions to extract local features prior to the self-attention process. On the other hand, there are models based exclusively on Transformers, such as the Vision Transformer (ViT) (Dosovitskiy et al., 2021) and the Data-efficient Image Transformers (DeiT) (Touvron et al., 2021), which process images entirely as sequences of patches. Furthermore, there are distinct approaches such as the Swin Transformer (Liu et al., 2021a), which, despite being a pure Transformer, introduces a hierarchical structure and shifted windows to achieve processing efficiency and scalability similar to those of convolutions. Figure 2.2 demonstrates this concept by showing an image entering the ViT architecture (Dosovitskiy et al., 2021) and being divided into patches. These patches are then transformed into vectors (linear projection), and Position Embeddings are added to encode the spatial arrangement of the patches, as the Transformer's self-attention mechanism is inherently permutation-invariant. Subsequently, the flow continues through the standard Transformer encoder.

2.3 Object Detection

The use of AI for object detection consists of identifying the presence of elements in an image, indicating their class and location. These elements are marked by bounding

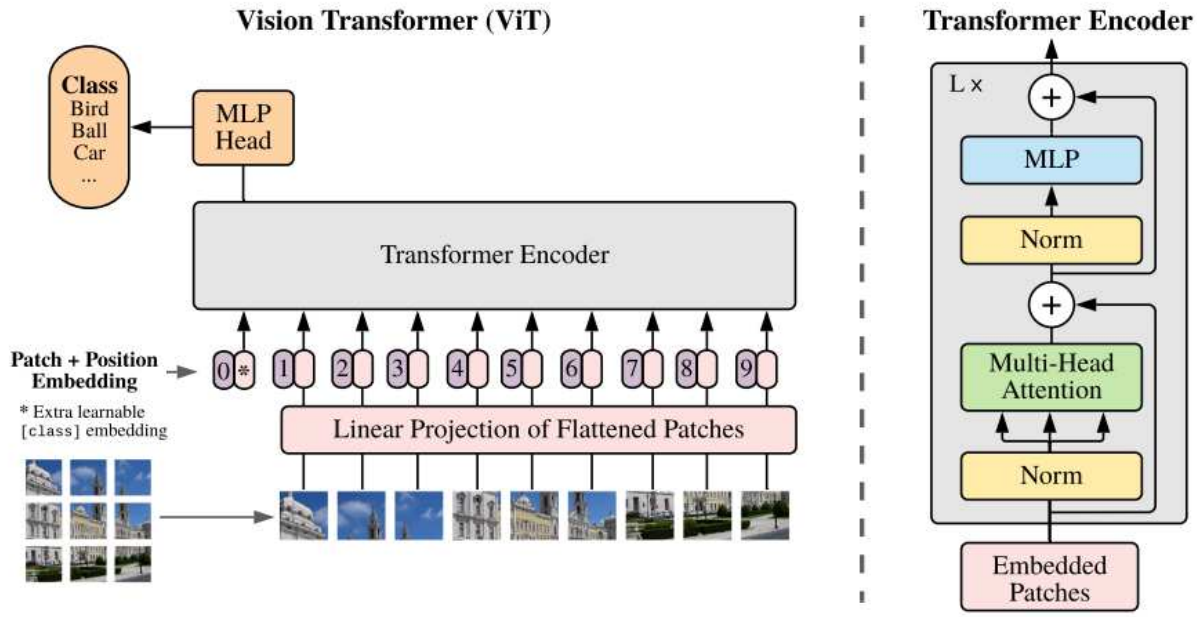

Figure 2.2: Overview of the ViT model. The architecture highlights the transformation of spatial image data into a sequence of patch embeddings, which are integrated with positional information before entering the Transformer encoder. Source: Adapted from Dosovitskiy et al. (2021).

boxes. This task has broad applicability and can be used to detect people (Elaoua et al., 2023; Gkioxari et al., 2018), vehicles (Wang et al., 2023), animals (Kumar et al., 2024), crops (Elhammamy et al., 2024), traffic signs (Changzhen et al., 2016), as well as everyday objects such as tv and cell phones (Sun et al., 2024).

Among its applications, we highlight the significant advances that improve the capability of autonomous vehicles to detect obstacles throughout their trajectory (Segu et al., 2024; Islam et al., 2024), supporting security systems in different environments, counting the flow of people (Elaoua et al., 2023) and vehicles (Anil et al., 2023) in establishments, monitoring animals (Gandhi et al., 2022; S et al., 2025; Rao et al., 2023; Delwar et al., 2025), and the growth of robotics in general (Bordado et al., 2023).

Accurate object detection is essential to ensure the successful usability of intelligent systems in real-world scenarios. However, achieving high accuracy remains a challenge, as models trained for a specific environment or application generally do not generalize well to others. Therefore, several studies focus on improving object detection performance across different domains and conditions. For instance, Sultanov et al. (2023) highlights the importance of precise detection and tracking of the ball and robots in robotic soccer to enable a successful match. Similarly, Islam et al. (2024) emphasize the relevance of object detection in autonomous driving systems, particularly stressing the need for real-time and reliable detection to ensure safe navigation.

Furthermore, effectively dealing with occlusion remains an open research problem and motivates the development of more robust architectures and training strategies

capable of reasoning under incomplete visual information (Gao et al., 2011). For instance, Kortylewski et al. (2020) address occlusion issues in vehicle detection by proposing compositional representations that explicitly model object parts, while Li et al. (2023) investigate occlusion-aware object detection in UAV-generated images, where overlapping objects and complex viewpoints are common.

Successful object detection requires a large dataset containing images properly labeled by their classes. Among the most widely used are PASCAL VOC (Everingham et al., 2010), which has about 20 classes and approximately 11,000 images in its latest 2012 update, and Microsoft (MS) COCO (Lin et al., 2014b), with 80 classes and approximately 123,000 images in its 2017 update. Both datasets have played a fundamental role in advancing computer vision research, especially in object detection.

The most common annotation formats and architectures will be explained below.

2.3.1 Annotation Format

Object detection dataset annotations provide information about the objects present in a given image. Additionally, each object has coordinates that define its bounding box and the class to which it belongs. Bounding boxes can vary in format, and some of the most well-known formats are PASCAL VOC, You Only Look Once (YOLO), and COCO.

In addition to the bounding box, some formats include additional information such as the object name, pose, image size (width and height), among others. More information on the main annotation formats is provided below.

In PASCAL VOC (Everingham et al., 2010) format, coordinates are represented as $(x_{\min}, y_{\min}, x_{\max}, y_{\max})$. Here, x indicates the position in the image width (from left to right) and y indicates the position in the image height (from top to bottom). The values of $x_{\min}$ and $y_{\min}$ correspond to the upper-left corner of the box, while $x_{\max}$ and $y_{\max}$ correspond to the lower-right corner. All values are in absolute pixels. Annotations are typically stored in `.xml` files.

Within each `object` block, `name` specifies the class label (*e.g.*, `cow`), `pose` optionally describes the object pose, `truncated` indicates whether the object is partially visible at the image boundary (1 = truncated, 0 = fully visible), and `difficult` identifies objects that are challenging to detect (1 = difficult, 0 = normal). The `bndbox` section contains the bounding-box coordinates, with the fields `xmin`, `ymin`, `xmax`, and `ymax` defining the object location in the image. For example, in the XML segment of Figure 2.3, three cow instances are annotated, each with its respective pixel coordinates.

In the YOLO (Redmon et al., 2016) format, annotations follow the pattern $(\text{class\_id}, x_{\text{center}}, y_{\text{center}}, \text{width}, \text{height})$, where all values (except `class_id`) are normalized between 0 and 1 with respect to the image dimensions. The `class_id` represents the object's class in numerical form (*e.g.*, `0` = cow). The values x_{center} and

y_{center} denote the normalized coordinates of the bounding-box center, while `width` and `height` indicate its normalized dimensions. Each line in the annotation file corresponds to one object, and the reference point is the center of the bounding box. Annotations are stored in plain `.txt` files. The normalization is computed as:

$$x_{\text{center\_norm}} = \frac{x_{\text{center}}}{\text{image\_width}}, \qquad y_{\text{center\_norm}} = \frac{y_{\text{center}}}{\text{image\_height}}$$

$$\text{width}_{\text{norm}} = \frac{\text{box\_width}}{\text{image\_width}}, \qquad \text{height}_{\text{norm}} = \frac{\text{box\_height}}{\text{image\_height}}$$

This normalization makes the annotations resolution-independent, allowing the same labels to be used with images of different sizes. For example, the YOLO annotation in Figure 2.3 contains three instances of class `0` (cow), each followed by the normalized bounding-box center and size.

In the COCO (Lin et al., 2014b) format, the coordinates are defined as $(x_{\min}, y_{\min}, \text{width}, \text{height})$, in absolute pixels. Here, $x_{\min}$ and $y_{\min}$ correspond to the upper-left corner of the bounding box, while width and height indicate the size of the box horizontally and vertically, respectively. The reference point is the upper-left corner of the box, and all dimensions are measured in pixels of the original image. The annotations are stored in a `.json` file that organizes the information into structured fields: the `"images"` section contains the image metadata, including resolution and file name; the `"categories"` section lists the object classes and their corresponding identifiers; and the `"annotations"` section stores the bounding boxes and related object information. For example, the JSON snippet in Figure 2.3 indicates an image of size 429×500 pixels with three annotated cows, each represented by its own bounding box and associated metadata.

The table 2.1 summarizes the formatting information mentioned above. The figure 2.3 shows a section of each format mentioned, illustrating how the bounding boxes are drawn on the image along with the class, which in this case is cow.

Table 2.1: Comparison of the most commonly used bounding box annotation formats.

Format	**Coordinates**	**Unit**	**Reference Point**	**File**
PASCAL VOC	$(x_{\min}, y_{\min}, x_{\max}, y_{\max})$	Pixels	Top left corner and bottom right corner	`.xml`
YOLO	$(\text{class\_id}, x_{\text{centro}}, y_{\text{centro}}, \text{largura}, \text{altura})$	Normalized (0–1)	Center of the box	`.txt`
COCO	$(x_{\min}, y_{\min}, \text{largura}, \text{altura})$	Pixels	Top left corner	`.json`

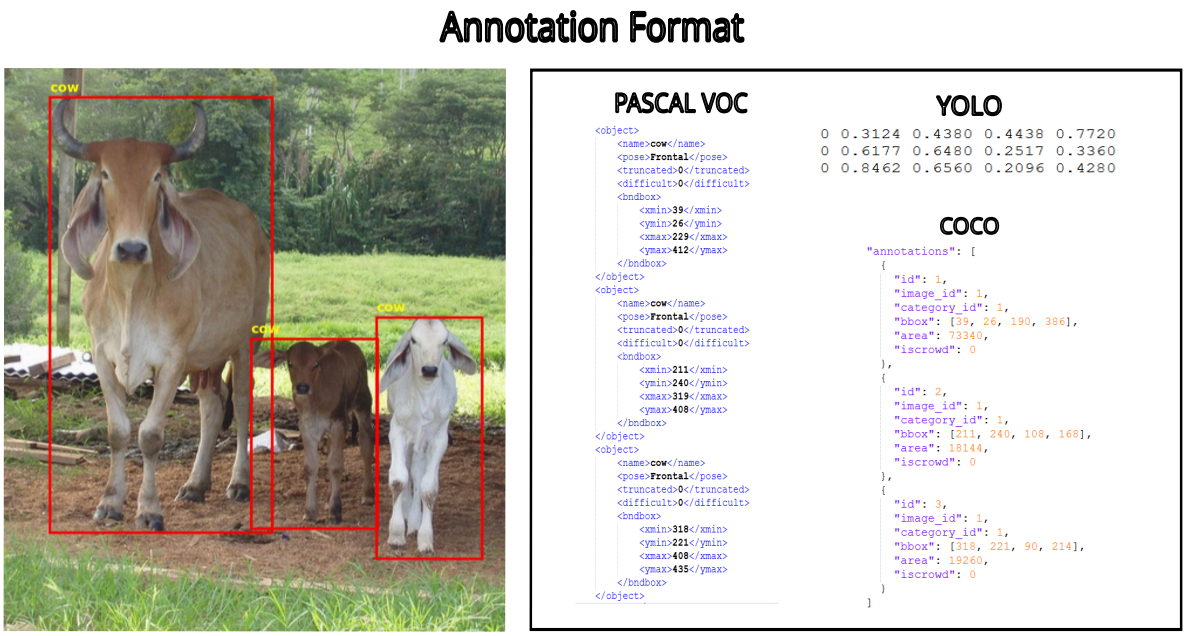

Figure 2.3: Example of annotation of bounding boxes drawn and excerpts of the three formats that represent the bounding box, PASCAL VOC, YOLO, COCO, in an image from the PASCAL VOC 2017 dataset. Source: Adapted from PASCAL VOC 2017 dataset.

2.3.2 Architectures

Several architectures are available for object detection. The best-known are Region-based Convolutional Neural Network (R-CNN) architectures, which were the first approach using CNN and were a milestone for this task (Girshick et al., 2014). Islam et al. (2024) comment that R-CNN ushered in a new era for object detection, bringing improved accuracy even with limited labeled training data, thanks to the use of pre-training combined with domain-specific refinement. This architecture selects approximately 2,000 candidate regions per image and passes each region through a CNN for feature extraction and subsequent classification.

Other well-known architectures are Fast R-CNN (Girshick, 2015) and Faster R-CNN (Ren et al., 2017), which evolved the R-CNN architecture and brought greater agility to the detection process. Fast R-CNN reduced training time by extracting features from the entire image in a single pass through the CNN, and Faster R-CNN introduced Region Proposal Network (RPN) to replace region selection methods, making the process even more efficient.

Other widely used architectures are Single Shot MultiBox Detector (SSD) (Liu et al., 2016), which performs detection in a single step using multiple feature maps at different scales, allowing better performance in detecting objects of varying sizes; RetinaNet (Lin et al., 2020), which introduced the Focal Loss loss function to deal with the problem of imbalance between background examples and objects, achieving higher accuracy in challenging scenarios; and DETR (Carion et al., 2020), which combines convolutional

networks with transformer layers to eliminate the need for region and anchor proposals, formulating detection as a direct prediction problem of bounding box-class pairs.

In this work we utilize the YOLO architecture, which is a widely known and established architecture for object detection tasks. More details about this architecture are provided below.

2.3.3 You Only Look Once (YOLO)

YOLO (Redmon et al., 2016) is widely recognized for its fast, high-accuracy, and real-time execution. This architecture simplified the detection task to a single stage, in contrast to previous methods that performed it in separate steps—region proposal generation, feature extraction, and classification. The first line in the Figure 2.4 illustrates the image passed once through a convolutional network, and then the generated results are filtered by a confidence threshold.

YOLO formulates object detection as a regression problem, directly predicting the coordinates of bounding boxes and the probabilities associated with each class, allowing the entire image to be processed in a single pass through the architecture. In this process, the input image is divided into grids $S \times S$, and each cell is responsible for predicting a fixed number of bounding boxes along with the class probabilities. This unified representation allows all detection to be performed in a single pass through the network. This is illustrated in the second line of Figure 2.4, where an image containing a dog, a bicycle, and a vehicle is divided into a regular grid, and each grid cell becomes responsible for predicting the bounding boxes and class probabilities of objects whose spatial centers lie within its boundaries, intuitively illustrating the operation of the unified prediction mechanism proposed by YOLO.

The Figure 2.5 presents the architecture used, composed of 24 convolutional layers followed by 2 fully connected layers. 1x1 layers are used to reduce the dimensionality of intermediate features, and the entire convolutional portion is initially pre-trained on ImageNet using reduced resolution (224×224) before being tuned for detection with double resolution (448×448).

The next version of YOLO, known as YOLOv2/YOLO9000 (Redmon and Farhadi, 2016), introduced significant improvements in accuracy, stability, and generalization while maintaining real-time detection. This version expanded the model's capability by combining training on detection datasets, such as VOC, with classification datasets like ImageNet, enabling it to detect thousands of classes at high speed. Among its key innovations is the use of k-means clustering to define anchor boxes based on the real dimensions of bounding boxes in the VOC and COCO datasets, resulting in more suitable priors (with k = 5 as the best trade-off between IoU and model complexity). YOLOv2 also began predicting box width and height as offsets relative to the anchor

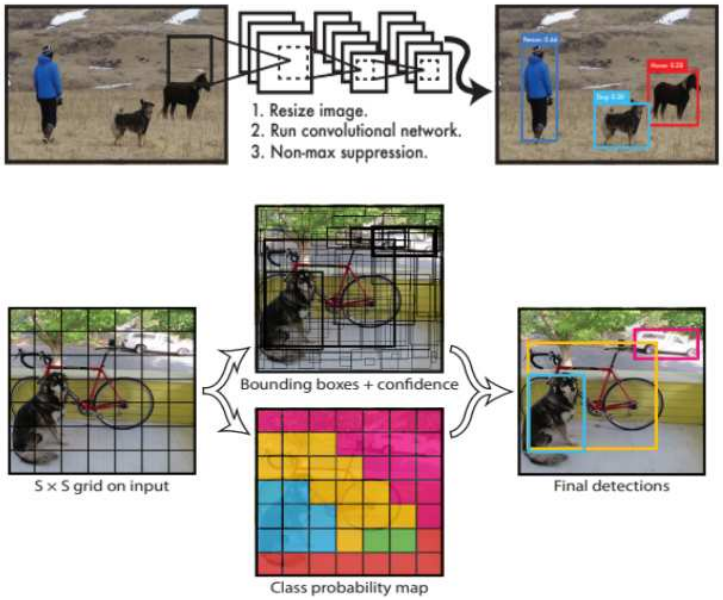

Figure 2.4: Overview of the YOLO detection system. The first part of this figure demonstrates an input image being resized and processed in a single pass through a convolutional network, producing detections filtered by confidence scores. The second part demonstrates the model dividing an image into an SxS grid, predicting bounding boxes, confidence values, and class probabilities for each cell. Adapted from Redmon et al. (2016).

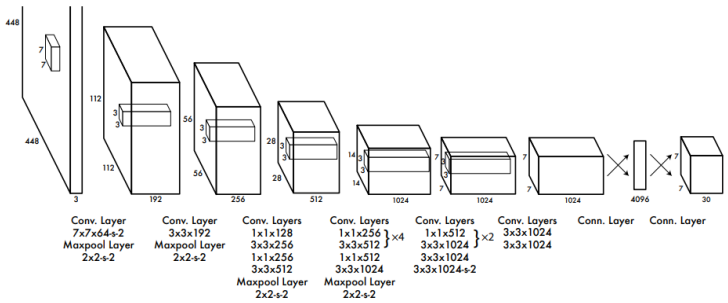

Figure 2.5: YOLO Architecture. The architecture comprises 24 convolutional layers followed by 2 fully connected layers, using 1×1 convolutions for dimensionality reduction and pre-trained ImageNet features. Source: Adapted from Redmon et al. (2016).

centroids, and estimating box center coordinates using a sigmoid function, which stabilized the spatial regression and contributed to improved accuracy.

YOLOv3 (Redmon and Farhadi, 2018) introduced a series of architectural refinements that improved accuracy while maintaining the high inference speed characteristic of the YOLO family. This version detects objects at three different scales, inspired by Feature Pyramid Networks, allowing the model to better handle objects of varying sizes.

YOLOv4 (Bochkovskiy et al., 2020) was developed with the goal of achieving an optimal balance between speed and accuracy. Its results place it on the Pareto curve, surpassing other state-of-the-art detectors in both metrics. A key contribution of YOLOv4 is its practicality, as it was designed to be trained and deployed on widely available

GPUs with 8 to 16 GB of VRAM. Furthermore, YOLOv4 follows the single-stage anchor-based detection paradigm and integrates a carefully selected set of enhancements ("bag of freebies" and "bag of specials") aimed at improving both classification and localization accuracy.

Currently, YOLO is at version 11, and after version 4 it continued to be developed and expanded by the Ultralytics team[1]. Since then, the framework has received improvements in speed, accuracy, and training efficiency, along with architectural refinements that increased flexibility and ease of deployment. It has also evolved to support additional computer vision tasks, such as semantic segmentation, pose estimation, object tracking, and classification, making it a comprehensive solution for real-time visual understanding.

2.4 Semantic Segmentation

The semantic segmentation task aims to identify and delineate objects within an image by following their precise shapes, unlike object detection, which identifies objects using bounding boxes. In this task, each pixel in the image is typically assigned a class label in the annotation. Semantic segmentation is widely used in various domains, such as detecting lane markings, pedestrian crossings, and obstacles for self-driving cars (Elhassan et al., 2024); agricultural monitoring (Heschl et al., 2024; Milioto et al., 2018); road maintenance (Mahmud et al., 2021); and medical imaging, where it helps segment regions for disease analysis (Parola et al., 2025; Albarracin et al., 2023), among other applications.

Several open datasets are widely used in the development of semantic segmentation tasks in more general contexts. Among the most notable are COCO-Stuff (Caesar et al., 2018), PASCAL VOC (Everingham et al., 2010), Cityscapes (Cordts et al., 2016), BDD100K (Yu et al., 2020), and ADE20K (Zhou et al., 2018).

COCO-Stuff contains 171 classes in total, 80 of which are things (*e.g.*, people, cars, trains) and 91 are stuff (*e.g.*, ground, sky, grass). PASCAL VOC includes 21 semantic segmentation classes, such as animals, vehicles, and people. The Cityscapes dataset provides 19 classes designed to support the understanding of urban scenes, particularly for autonomous and smart vehicles. The classes are grouped into the following categories: flat, construction, nature, vehicle, sky, object, human, and void. The BDD100K dataset also focuses on urban scene understanding, similarly to Cityscapes. It contains 100,000 driving videos captured in streets and highways, split into 70,000 for training, 20,000 for testing, and 10,000 for validation, with various classes such as car, person, bus, and bicycle, among others. This dataset also supports additional tasks beyond semantic segmentation, such as object detection. Finally, the ADE20K dataset

[1] https://www.ultralytics.com/

includes more than 20,000 images annotated for semantic segmentation tasks, featuring diverse classes such as cars, tables, doors, houses, and lights.

The most common annotation formats and architecture for this task will be explained below.

2.4.1 Annotation Format

The annotation format for this task usually consists of mask images, where each pixel encodes an integer class label corresponding to the object at that location. These class IDs can later be mapped to specific colors for visualization purposes, facilitating the interpretation of segmentation outputs (Ge et al., 2024; Lin et al., 2014b; Everingham et al., 2010). Figure 2.6 shows an example of a semantic segmentation annotation, where the color green represents the class cow in the PASCAL VOC 2017 dataset.

Some architectures, such as the YOLO architecture, also use polygon-based annotations. In this case, objects are delimited by coordinates, and each object instance is represented by a list of points forming a polygon that follows the object's shape. This type of annotation is usually stored in `.txt` files and has the advantage of reducing storage requirements. The coordinates are organized in a sequential array of points in the format

$$[x_1, y_1, x_2, y_2, \ldots, x_n, y_n],$$

where each pair (x_i, y_i) represents the position of a vertex of the polygon enclosing the object. This format is typically used as input to the architecture but is later converted into a mask representation for training or evaluation.

Figure 2.7 shows an example of polygons used to delineate regions of interest in an image. Two polygons are drawn: the first (main polygon) outlines the racket head, where the ball strikes, and the second outlines the handle, where the player holds it. Four points are highlighted in red on the main polygon to illustrate an example subset of vertices selected for analysis.

2.4.2 Architectures

Several architectures can be used for this task, among which the Fully Convolutional Network (FCN) stands out, considered the first fully convolutional architecture for segmentation (Long et al., 2015). This architecture adapted pre-trained classification networks, such as AlexNet, VGG, and GoogleNet, to the segmentation task, using transfer learning and fine-tuning. Furthermore, it dispensed with more complex methods, such as the use of region proposals employed in the Simultaneous Detection and Segmentation (SDS) architecture. This architecture has been used in studies such as hemorrhage segmentation in retinal images (Nurul Qomariah et al., 2021), brain tumor

Figure 2.6: Example of semantic segmentation annotation on an image from the PASCAL VOC 2017 dataset. Source: Adapted from PASCAL VOC 2017 dataset.

segmentation (Yang et al., 2019), and scene analysis (Ali et al., 2023), among others.

Another well-known architecture is U-Net (Ronneberger et al., 2015), a semantic segmentation network created in 2015 with a focus on biomedical images. Currently, it is widely used not only for problems in the medical field (Koudia et al., 2022), but also for various types of problems that can be solved with semantic segmentation (Gruszczyński et al., 2022; Giang et al., 2020; Kent et al., 2023). Nugraha et al. (2023) used U-Net to aid in the identification of dolphins through images captured by UAVs. Shang et al. (2023) also used the U-Net architecture to aid in the creation of a georeferencing algorithm for UAV images. The U-Net architecture contains several convolutional layers that are essential for feature extraction from images and is composed of two main parts. The first is the encoder responsible for reducing resolution and capturing the image context, and the second is the decoder, which increases resolution and generates image segmentation. For this reason, the architecture is U-shaped. Furthermore, another characteristic of the U-Net are the skip connections. These connections link corresponding layers of the encoder and decoder, allowing the network to retain detailed spatial information.

Like U-Net, Swin-Unet (Cao et al., 2021) is also a U-shaped encoder-decoder architecture. It also incorporates features of the Swin Transformer (Liu et al., 2021b). It was created for semantic segmentation of biomedical images. This architecture, released in 2021, does not use convolutional layers, but rather layers based on the Transformer architecture, with attention layers. Swin-Unet, although inspired by medical problems

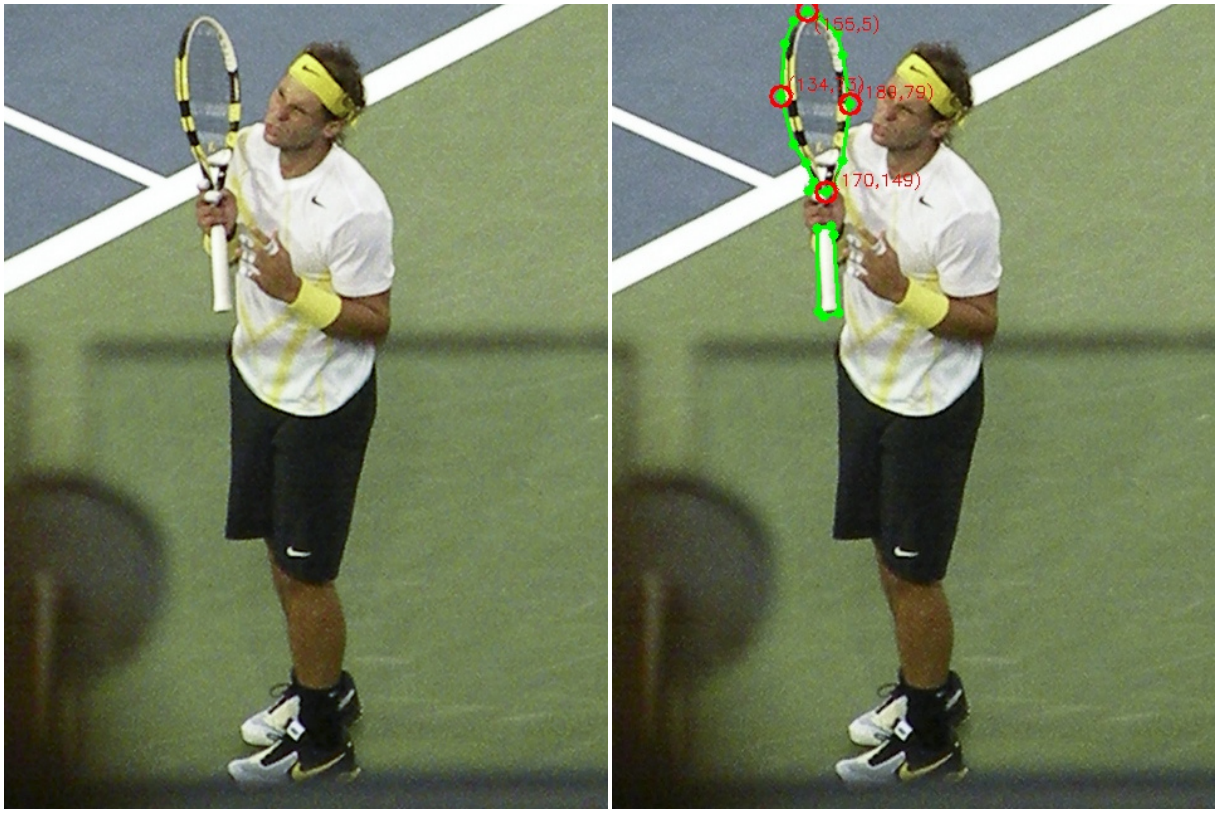

Figure 2.7: Example of polygons used to delimit the regions to be segmented. The image shows two polygons: one outlining the head of the racket, where the ball hits, and another outlining the handle held by the player. Four vertices are highlighted in red in the main polygon as an example of selected points. Source: Adapted from the COCO dataset.

and well-used in this context (Liao et al., 2023; Xue and Du, 2024), has also been used to propose solutions to semantic segmentation problems in other contexts, such as in ore segmentation (Tang et al., 2022).

The DeepLab (Chen et al., 2014) architecture, proposed in 2014, introduced the use of dilated convolutions, which increase the spacing between the kernel elements, allowing for a broader image coverage area without increasing the number of trainable parameters. This architecture uses different backbones in the image feature extraction phase and has evolved over the years, gaining new versions: DeepLabv2 (Chen et al., 2017a), DeepLabv3 (Chen et al., 2017b), and DeepLabv3+ (Chen et al., 2018). DeepLab has been widely used to solve various problems, including underwater image segmentation (Liu and Fang, 2020), road segmentation in UAV images to aid road maintenance, intelligent transportation systems, and urban planning (Mahmud et al., 2021), environment recognition for autonomous wheelchairs (Nishida et al., 2021), and medical assistance (Saranya et al., 2024), among others.

We now detail the High-Resolution Network (HRNet) and High-Resolution Network - Object-Contextual Representations (HRNet.OCR) architectures, which is used in this work.

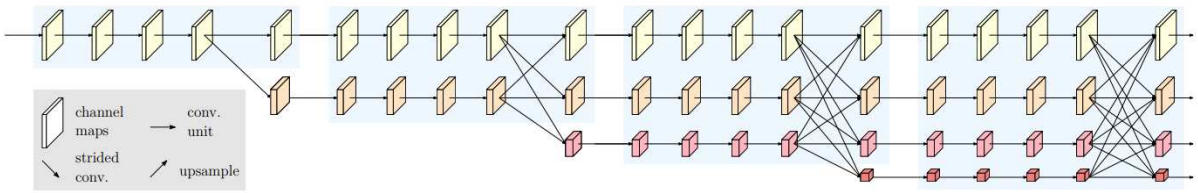

Figure 2.8: Overview of the HRNet. The architecture maintains multiple parallel branches operating at different resolutions and progressively adds new lower-resolution streams in each stage. Throughout the network, feature information is continuously exchanged across resolutions, enabling rich and spatially detailed representations. Source: Adapted from Wang et al. (2020).

2.4.3 HRNet and HRNet.OCR

The HRNet (Wang et al., 2020) was introduced to address computer vision tasks such as object detection and semantic segmentation. Unlike conventional convolutional neural networks that progressively downsample feature maps to extract semantic information, HRNet maintains high-resolution representations throughout the entire learning process. To achieve this, the architecture preserves multiple parallel convolutional streams operating at different resolutions and repeatedly exchanges information among them. This multi-scale interaction produces feature representations that are both semantically rich and spatially precise. The network is organized into several stages, with each new stage introducing an additional lower-resolution branch, as illustrated in Figure 2.8.

The main innovation of HRNet lies in its parallel multi-resolution design and the continuous information exchange between high- and low-resolution streams. By maintaining high-resolution feature maps instead of restoring them at the end of the network, HRNet provides a robust and versatile backbone for a wide range of computer vision tasks.

The HRNet.OCR architecture (Yuan et al., 2021), proposed shortly after the introduction of HRNet, incorporates attention-based mechanisms through the Object-Contextual Representations (OCR) module. The purpose of this module is to enhance pixel representations by modeling their relationship with object-level context in the image. In practice, OCR first identifies soft object regions from the coarse segmentation logits produced by the backbone. Then, it estimates object-level region representations and uses them to compute object–pixel affinities. These affinities serve as attention weights that refine each pixel representation with aggregated object-level contextual information.

The OCR module strengthens semantic segmentation by explicitly capturing how each pixel relates to all object regions in the scene. This results in a more discriminative and coherent representation, ultimately producing a refined segmentation map. Figure 2.9 illustrates the OCR workflow after the feature extraction stage: (i) forming the soft

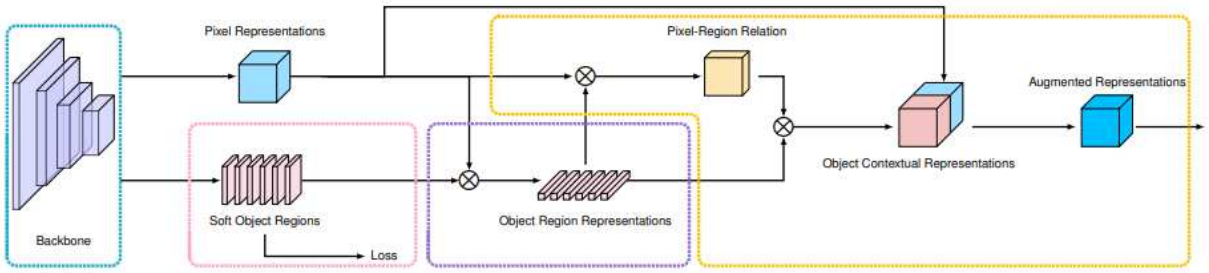

Figure 2.9: Illustration of the OCR module. First, soft object regions are generated, then object-level region representations are computed, and finally pixel features are refined using object–pixel contextual relationships. The module enhances segmentation by incorporating object-aware contextual information. Source: Adapted from Yuan et al. (2021).

object regions (pink dashed box); (ii) estimating the object region representations (purple dashed box); (iii) computing the object–contextual representations and the augmented pixel features (orange dashed box).

HRNet.OCR has been successfully applied to various domains, including power pole segmentation (Singh et al., 2023) and panoptic segmentation (Lu and Zhu, 2022), demonstrating its effectiveness in scenarios that require both fine-grained detail and robust contextual reasoning.

2.5 Domain Adaptation

In AI, training with supervised models requires a labeled dataset for efficient model learning. However, there is often insufficient data available in the target domain, or the target domain may lack labels. This problem is known as domain differences or discrepancies, where the model is trained on a specific dataset but needs to converge well in a different domain, yet this good convergence does not occur (Hidayaturrahman et al., 2024). Zhao et al. (2022) demonstrated in Figure 2.10 that training a model in one domain, for example, a simulation domain, and applying it to another domain, such as the real world, does not work well. In contrast, direct training on the target set has much superior performance.

The difference between domains is known as domain shift or distribution shift and refers to changes in the data-generating process or in the representation of the data between the source and target domains. In this context, the statistical properties of the observed data may vary due to changes in the measurement system or in the conditions under which the data are acquired (Quionero-Candela et al., 2009). This shift may affect low-level appearance aspects (such as color, texture, or illumination), mid-level structural features (such as shapes or spatial configuration), or even high-level semantic distributions (for example, the frequency of classes). As a consequence, the features learned in the

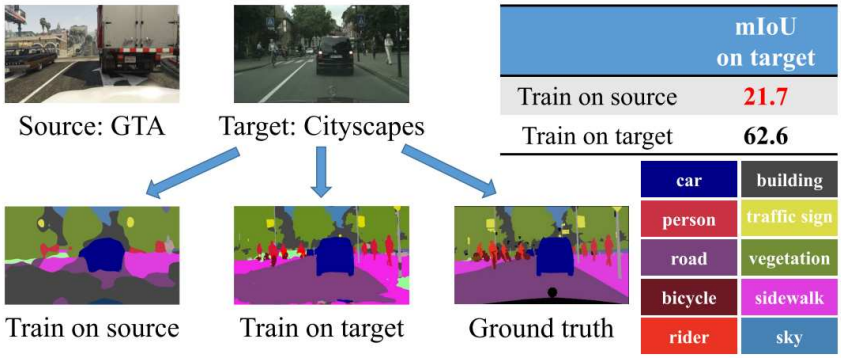

Figure 2.10: Illustration of the domain shift problem in semantic segmentation. A model trained on a labeled source domain (GTA, synthetic images) exhibits a significant performance drop when directly applied to an unlabeled target domain (Cityscapes, real images). In contrast, training directly on the target domain yields substantially better results, highlighting the impact of distribution differences between domains and the need for Domain Adaptation techniques. Source: Adapted from Zhao et al. (2022).

source domain may not transfer effectively to the target domain, leading to a significant drop in model performance. The most common cases include:

1. models trained with synthetic images and applied to real images;
2. images captured in different seasons (*e.g.*, summer for winter);
3. different sensors or cameras (*e.g.*, drone for satellite);
4. distinct visual styles (*e.g.*, urban datasets for rural datasets).

As illustrated in the figure, pixel-by-pixel semantic segmentation suffers a substantial drop in accuracy when a model trained on the labeled source domain (in this case, the GTA domain) is transferred without adaptation to an unlabeled target domain (in this case, the Cityscapes domain). In contrast, models trained directly on the target data (in this case, the Cityscapes domain) achieve substantially better performance.

The process of a model learning in the presence of changes between the distributions in the source and target domain is referred to as Domain Adaptation (DA) (Ganin et al., 2016a). In many cases, DA is relevant when images differ in lighting, resolution, viewpoint, sensor characteristics, geographic regions, or style (Schenkel and Middelmann, 2020). The DA scenario varies according to the availability of labeled data in the target domain, which leads to supervised, semi-supervised, or fully unsupervised settings. In supervised domain adaptation, both source and target domains provide labels; in semi-supervised scenarios, only a small portion of target samples is labeled; and in unsupervised domain adaptation (UDA), no target labels are available (Çallı et al., 2021).

In supervised domain adaptation, common solutions focus on statistical alignment techniques such as CORAL (Sun and Saenko, 2016), which aligns second-order statistics

Table 2.2: Common solutions and state-of-the-art approaches for different types of DA.

Type of Domain Adaptation	Common Solutions	State of the Art
Supervised	Fine-tuning, CORAL, MMD	MLDG, MetaReg, PEFT
Semi-supervised	Pseudo-labeling, consistency regularization	Mean Teacher, FixMatch
Unsupervised (UDA)	DANN	CDAN, CyCADA, self-training

between domains, and MMD (Gretton et al., 2012), a kernel-based distance that measures the discrepancy between distributions. These methods, combined with straightforward fine-tuning using the available target labels, which are typically fewer than those from the source domain, provide effective baseline strategies. More advanced state-of-the-art approaches rely on meta-learning strategies: Meta-learning for domain generalization (MLDG) (Li et al., 2018), which simulates virtual domain shifts during training to improve generalization, and MetaReg (Balaji et al., 2018), which learns a regularization function that enables better adaptation to new domains. Parameter-efficient fine-tuning (PEFT) (Houlsby et al., 2019) further improves adaptation by updating only a small portion of the model parameters to capture domain-specific variations.

In semi-supervised domain adaptation, the target domain provides only a few labeled samples alongside many unlabeled ones, and methods typically rely on pseudo-labeling and consistency regularization, as in MT (Mean Teacher) (Tarvainen and Valpola, 2018) and FixMatch (Sohn et al., 2020). In Unsupervised Domain Adaptation (UDA), no target labels are available, making distribution alignment more challenging; adversarial approaches such as DANN (Ganin et al., 2016b) and CDAN (Long et al., 2018), as well as image-translation methods like CyCADA (Hoffman et al., 2017) and self-training strategies, refine target predictions without supervision and represent some of the strongest solutions in fully unsupervised scenarios. Table 2.2 summarizes the information.

Chapter 3

Related Work

To monitor farms and agricultural areas, training AI models is a great ally, allowing for the effective use of images captured by various equipment, whether aerial or ground-based. However, for supervised AI models to achieve this capability and converge well during training, a large number of labeled images are required. Image labeling is a bottleneck, as it is a task that requires many hours of human labor and is also costly.

In this context, we present several works that address the following topics:

1. Use of synthetic images for training AI models;
2. Integration of different datasets; and
3. Safety on farms and/or agricultural environments.

3.1 Use of synthetic images for training AI models

Many studies have explored the use of synthetic images to address the scarcity of labeled real images. The goal of these studies was to improve the training of AI models for various tasks.

Arezoomandan et al. (2024) used synthetic images to evaluate the impact of using these images on object detection models. They used AirSim as a UAV simulator and Unreal Engine as a graphics engine and simulated flights in four environments with diverse plant life and climates. After generating the synthetic images, the authors combined these images with real images and evaluated the model training results, demonstrating the effectiveness of using these images to train object detection models.

Öztürk and Erçelebi (2021) generated synthetic images of UAVs and birds for classification tasks, training models solely with synthetic data and evaluating them on real images, achieving satisfactory results. The dataset was created using the Unity game engine, where 3D models of UAVs and birds were inserted into a virtual studio, resulting in approximately 69,120 images (34,560 of UAVs and 34,560 of birds). The AI models were trained with deep networks such as AlexNet, SqueezeNet, and VGG16, with the inclusion of a Corner Detection and Nearest Three-Point Selection (CDNTS) layer that significantly improved performance in classifying both synthetic and real images.

Silva et al. (2024) developed a highly realistic virtual environment based on orthophotos of Switzerland and simulated UAV flights to generate photorealistic images of the region. They created a pipeline capable of producing a large volume of images annotated with semantic segmentation labels and depth information. The authors used ROS and Gazebo tools, with advanced rendering techniques and textured terrain models, ensuring a realistic visual representation of the scene. The results demonstrated that the pipeline could generate large and highly realistic datasets, useful for training computer vision algorithms; however, no training experiments were conducted.

Klein et al. (2024) addressed early disease detection in greenhouse-grown tomato plants by proposing an iterative framework for generating synthetic training data. Procedural models were used to create realistic plant geometries and textures, rendering photorealistic images labeled for healthy and diseased plants. The dataset was iteratively refined based on model performance, with all classifiers trained exclusively on synthetic images.

Blaga and Nedevschi (2022) created the Forest Inspection Dataset, which contains both real and synthetic images focused on forest inspection and monitoring, particularly for deforestation assessment. The authors employed advanced image segmentation techniques, three-dimensional reconstruction of the environment, and the development of quantitative metrics to measure deforestation over time. They evaluated the performance of HRNet and PointFlow networks for semantic segmentation under various transfer learning scenarios from synthetic to real data.

Motoi et al. (2025) proposed a method for generating synthetic data that segments and seamlessly matches anomalous fruit parts with healthy parts using image processing techniques. The method was applied to table grapes, leveraging the Segment Anything Model (SAM) for automatic segmentation and a Dual-Canny Edge Detection (DCED) filter to enhance the detection of texture variations indicative of diseases or pest infestations. Their experiments demonstrated that incorporating these synthetic samples significantly improved the performance of anomaly classifiers when combined with real data.

Sun et al. (2025) used UAV images and interpolated synthetic samples to monitor potato late blight, aiming to reduce noise and class overlap. To address crop variability caused by factors such as variety, genetics, and climate, they applied transfer learning across datasets. Their approach included an improved SMOTE method (SMOTE-CS) that incorporates feature selection based on the FS-CS principle and the Induced Minkowski Ordered Weighted Averaging Distance (IMOWAD) distance to better define synthetic sample boundaries and suppress noise. Additionally, they compared shallow 1D-CNN and deep DRSN models, showing that while the 1D-CNN achieved very high accuracy, the DRSN model with nonlinear soft-thresholding demonstrated superior noise robustness and generalization across datasets.

The cited studies offer important contributions to the use of synthetic data as a means

of expanding training datasets for AI models in various domains. However, they differ from our work by focusing on specific problems, such as bird detection, drone identification, deforestation assessment, or fruit anomalies. In contrast, our approach specifically targets agricultural monitoring applications with the goal of improving the safety of agricultural environments.

3.2 Integration of different datasets

For supervised AI models to achieve high efficiency in their results, it is crucial to have a large number of images for training and validation, especially in the specific context of this work, which is rural, focusing on agriculture and farms. However, when available data is limited, searching for smaller datasets and combining them can be a viable alternative. For this purpose, we reviewed some studies related to the topic.

Zhou et al. (2022) explored training a model for multiple domains by integrating multiple datasets, highlighting challenges such as semantic discrepancies and annotation inconsistencies. They proposed a unified object detection model with an automatic taxonomy to group similar classes. This approach is most effective when there are direct correspondences between classes across the different datasets. However, it presents limitations when there are unannotated classes in certain datasets or when there are no semantically similar annotated counterparts.

Zhao et al. (2020) worked on integrating multiple datasets and proposed a learning approach that combines dataset-specific detectors trained on each dataset with a pseudo-labeling technique to generate synthetic annotations for missing or unlabeled categories. The results demonstrated that this strategy significantly improved category coverage and reduced false positives, leading to overall better performance in unified detection tasks, similar to the results presented in this study.

Kim et al. (2022) investigated the problem of jointly training semantic segmentation models using multiple datasets with different and potentially incompatible label spaces. Instead of proposing a new network architecture, the authors introduced UniSeg, a training framework that modifies the loss function to handle semantic conflicts across datasets. By replacing the standard cross-entropy loss with a class-independent binary cross-entropy formulation, the method avoids penalizing the model for classes that are absent or defined differently in each dataset. In addition, UniSeg accounts for semantic overlap between classes from different datasets, allowing related categories to be learned jointly during training. This strategy enables effective multi-dataset training without requiring manual relabeling, improving generalization performance, especially on unseen datasets.

The referenced works present techniques for integrating multiple datasets; however, their primary goal is to address missing annotations or semantic inconsistencies by

expanding label coverage or leveraging implicit class relationships across datasets. In contrast, our approach focuses exclusively on integrating the target classes of interest, without introducing additional categories, while still achieving competitive model performance.

3.3 Safety on farms and/or agricultural environments

Working on farm and agricultural security is essential for the preservation of people, fields, and animals. Therefore, we reviewed studies related to improving security focused on rural environments and the use of AI.

Several studies have focused on agricultural security, especially on the detecting animal incursions that could harm farm animals or damage crops. For example, Mamat et al. (2022) proposed a model for detecting four categories of animals—elephant, monkey, tiger, and wild boar—that frequently invade agricultural areas. Their work demonstrated promising results using the YOLOv5 architecture, achieving high accuracy in detecting these animals. They also compared their results with the YOLOv4 architecture, which performed equally well. The study used a dataset with only 280 images, with 70 images per category, which limits its applicability in different contexts.

Naveenkumar et al. (2022) also worked to prevent attacks on farms and people using DenseNet201. The work proposed the detection of nine animal classes through video analysis, adopting the classes bear, bison, elephant, fox, leopard, lion, tiger, pig, and horse. In addition to detection, the system was designed to trigger preventive actions when the presence of potentially dangerous animals was identified, such as activating alarms or sending real-time alerts to farmers.

Akhil et al. (2023) employed several techniques to develop a virtual fence to prevent human-animal conflicts on farms with cultivated crops. Their approach involved using motion detection to identify the presence of animals. Once movement was detected, the system processed the image to confirm the presence of the animals. If confirmed, the system used loud sounds and high-intensity strobe lights to scare them away. For image detection, a convolutional neural network implemented in TensorFlow was used. The dataset contained 100 images, including the categories cat, dog, elephant, and wild boar. Although the study addressed the entire virtual fence process, the accuracy of animal detection was unsatisfactory. The authors noted the need for more images and additional model training to improve accuracy.

G et al. (2025) proposed the creation of a smart and sustainable digital fence to mitigate human-animal conflicts, especially in agricultural and forestry areas, with the combined use of Internet of Things (IoT) and AI. They used a hybrid model integrating YOLOv8 and MobileNetv3 for real-time intrusion detection and classification. Furthermore, the proposed digital fence integrates advanced hardware components such

as Raspberry Pi, ultrasonic, and infrared sensors.

Several studies have used the Internet of Things (IoT) and sensors to detect human intrusions. Hsu et al. (2019) implemented a system to detect unauthorized persons using image recognition combined with sensor fusion and 5G technology. The system integrated cameras and beacons for real-time monitoring and achieved an average identification accuracy of approximately 90% during field tests, demonstrating its effectiveness in preventing crop theft and increasing farm security. Oyelade et al. (2024) implemented an intrusion detection system that combines IoT and computer vision to identify both humans and animals. The system performs facial recognition of people and alerts owners in case of intrusion. The study focused on the detection of cows, goats, and people, using Fast-R-CNN for object detection. The dataset used contained 3,752 images.

Research efforts have been made to improve person detection in rural environments to increase the safety of autonomous vehicles in these areas. Tabor et al. (2015) reinforces the challenges of autonomous driving in rural settings, where people are often occluded by weeds and branches, and their posture is not always upright. The study evaluated three image-based pedestrian detection algorithms — Aggregate Channel Features (ACF), Deformable Parts Model (DPM), and a CNN — using data collected from an autonomous tractor operating in off-road conditions. Neigel et al. (2020) developed the Off-Road Pedestrian Detection Dataset (OPEDD) dataset for this purpose. Similarly, the NREC dataset (Pezzementi et al., 2017) was introduced to advance autonomous driving research with an emphasis on person detection in agricultural settings.

The referenced works provide valuable contributions to field monitoring. However, several of them focus on detecting only specific classes that do not encompass the full range of categories relevant to terrestrial monitoring in our study. Others concentrate on improving AI techniques for this domain, comparing model architectures or proposing new monitoring approaches, yet they do not address the challenge of labeled data scarcity as we do. Additionally, some works introduce new datasets, but they do not include all the necessary classes for effectively monitoring both people and animals.

Chapter 4

Semantic Segmentation - Synthetic Images and Cross-Domain for Pre-training

Monitoring farms and agricultural areas can be performed aerially using images captured by UAVs. In this context, analyzing these images with AI through the task of semantic segmentation is a powerful tool for identifying different regions, especially the most important ones. Since labeled images are scarce for training supervised AI models, in this section we investigate the use of synthetic images generated in the same domain as the target dataset, as well as real images from different domains used as pre-training strategies.

4.1 Methodology

To address the need for a large number of labeled images to train AI supervised models for monitoring farms and agricultural areas using UAVs, this topic proposes a study of different pre-training strategies to improve the effectiveness of semantic segmentation models on real-world data. Pre-training is useful for leveraging the learning of a model trained on a given dataset and replicating them to the desired dataset. This approach is commonly used when an adequate number of images are not available in the target dataset, thus reducing the need for extensive manual annotation.

The strategies we employ utilize a large synthetic dataset composed of labeled photorealistic synthetic images, as well as a large dataset of real-world images from a different domain (Remote Sensing), to perform various pre-training configurations. This study is subdivided into the following steps: Dataset Definition, Pre-training Strategies, and Pre-training Effect Evaluation. Figure 4.1 demonstrates this part of the methodology.

Domain Adaptation has been widely investigated as a strategy to mitigate performance degradation caused by distribution shifts between source and target domains, especially in scenarios involving synthetic-to-real transfer, different sensors, or

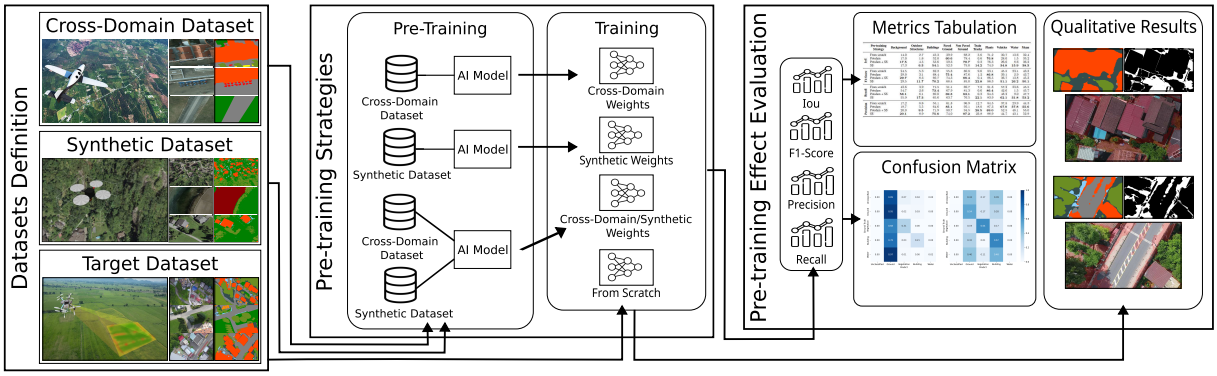

Figure 4.1: Overview of the semantic segmentation study - synthetic images from the same domain and real images from different domains for pre-training. The study is divided into three stages. In the first, we collect the target, synthetic, and cross-domain datasets. In the second stage, we execute the pre-training strategies and training on the target dataset, using the pre-trained weights and also a from-scratch training. In the third stage, we evaluate the pre-training quantitatively and qualitatively.

distinct acquisition conditions. While several works propose explicit DA techniques, such as adversarial learning, statistical alignment, or image-to-image translation, our work does not employ a dedicated domain adaptation algorithm in the strict sense. Instead, we explore domain-related challenges through alternative and complementary strategies.

In this study, we leverage synthetic images and cross-domain real datasets as a pre-training mechanism, aiming to improve feature representation before fine-tuning on the target agricultural domain. This approach is closely related to DA in that it addresses domain shift indirectly, by exposing the model to diverse visual distributions during training, rather than explicitly aligning source and target domains.

4.1.1 Datasets Definition

To validate different pre-training strategies, the selection of datasets is a fundamental step in this study. First, it is necessary to identify the main dataset that represents the focus of our problem — the target dataset consisting of aerial images in an agricultural environment. Since a large amount of data is not expected, two additional datasets are required to support the pre-training process. The two additional datasets, plus the target dataset used in this work, are described in detail below.

1. **Target Dataset**: Since our focus in this part of the study is on aerial monitoring of farms and agricultural areas, the Target Dataset should consist of real images captured by UAVs under controlled flight conditions and annotated for the semantic segmentation task. The use of real UAV imagery collected under controlled flight conditions ensures consistent coverage, stable illumination, and realistic visual patterns that reflect the actual monitoring scenario. Furthermore, they should simulate a monitoring task of an area with farm characteristics,

featuring a fixed height and speed, automatic flight control through the definition of waypoints, top-down images, and a flight that predominantly covers natural areas with a predominance of low vegetation, trees, water channels, roads or dirt paths, vehicles, and buildings. It is important to highlight that buildings are relevant structures in rural areas, representing barns, warehouses, farm headquarters, garner, and storehouses. Mapping these elements allows the system to distinguish infrastructure zones from natural areas, supporting security, resource monitoring, and operational planning on farms.

2. **Synthetic Dataset**: The synthetic dataset should be similar to the target dataset but created in a virtual environment. This similarity can be understood as both belonging to the same image domain. For this purpose, the images must be aerial, captured from a top-down perspective, and represent areas with agricultural characteristics—predominantly low vegetation, trees, water channels, roads or dirt paths, vehicles, and buildings. In addition, the images captured during the flights should have a similar altitude to the target dataset in order to avoid significant discrepancies in object scale. The dataset must also include annotations for the same task, namely, semantic segmentation. Finally, since it is a synthetic dataset, the generation of images and annotations is performed using algorithms and 3D simulation tools, which allow for the creation of a large volume of labeled data with consistent and reliable annotations.

3. **Cross-Domain Dataset**: We define this dataset as one that contains a larger amount of labeled data compared to the Target Dataset, also annotated for the semantic segmentation task but belonging to a slightly different domain. This slightly different domain, which we refer to as the cross-domain, maintains visual proximity to the target domain. In this context, the images are also aerial, captured from a top-down perspective, but may differ in flight type, altitude, and even object scale. Since both the Target and Synthetic Datasets consist of aerial images captured by UAVs over agricultural areas, we propose the use of a remote sensing dataset as the cross-domain dataset. Such datasets meet the mentioned characteristics, as they share virtually the same viewpoint (top-down). Furthermore, high-resolution remote sensing imagery can serve as a valuable source of information. Due to extensive research efforts and public initiatives in the remote sensing field, there is wide availability of high-quality labeled data, which can significantly benefit the AI model during the pre-training phase.

4.1.2 Pre-training Strategies

To increase the effectiveness of AI models in the task of semantic segmentation in images from farm monitoring with UAVs, we propose studying the effect of different pre-training strategies. As mentioned previously, there is a scarcity of labeled UAV data for rural area monitoring, and our hypothesis is that an AI model will benefit from a pre-training strategy. Generally, pre-training is performed with real data from the same domain. However, since there is no publicly labeled dataset with the necessary number of images to train the model, we propose studying the use of data from another domain, synthetic data, and combinations of this data.

To carry out the pre-training strategies, we use a well-established AI model for the semantic segmentation task. Choosing a robust and widely adopted model ensures comparability with related studies and provides a solid basis for evaluating the proposed pre-training approaches. We chose the HRNet.OCR architecture for the experiments in this work, which we discuss in the next section. Below, we present more details about the training procedures that support the study of this methodology.

1. Training from Scratch (From Scratch): As a sanity check and to establish a lower performance bound in our study, we train the AI model entirely from scratch, meaning that all network parameters are randomly initialized rather than using any pre-trained weights. This approach allows us to evaluate how well the model learns solely from the target dataset, without relying on prior knowledge. However, since the dataset contains a limited number of labeled samples, the model is expected to underperform compared to those benefiting from pre-training strategies.

2. Pre-training with **Cross-Domain Dataset**: The first pre-training strategy is based on the use of the Cross-Domain Dataset. Given the large number of labeled images and the visual similarity between this set and the Target Dataset, we expect this pre-training approach to improve the model's performance compared to training from scratch, as it provides a stronger initialization and better feature generalization.

3. Pre-training with **Synthetic Dataset**: The second pre-training strategy employs the Synthetic Dataset, which shares the same domain as the Target Dataset and contains a large number of labeled samples, differing only in the fact that the data is artificially generated. We expect this pre-training to help the model learn the key visual features needed for effective initialization. Although the data is synthetic, its realistic characteristics provide advantages compared to training the model from scratch.

4. Combined pre-training with **Cross-Domain Dataset** and **Synthetic Dataset**: We propose combining both pre-training strategies through two sequential stages.

In the first stage, the model is pre-trained using the Cross-Domain Dataset, allowing it to learn general visual representations from a large amount of labeled data. In the second stage, the model is further pre-trained on the Synthetic Dataset, using the weights obtained from the previous stage as initialization. This second phase aims to adapt the model to a domain that is more closely related to the Target Dataset while preserving the knowledge acquired in the first step.

Once we have the weights for each pre-training strategy, we must use each weight and perform the training on the target dataset using that weight. Therefore, three separate training sessions will be run on the target dataset, taking advantage of the three prepared weights: cross-domain, synthetic, and combined. The second stage (Pre-training Strategies) of Figure 4.1 demonstrates the weights from the "Pre-training" stage being used in each "Training" session.

4.1.3 Pre-training Effect Evaluation

After conducting the training using the pre-training strategies, the results are analyzed both quantitatively and qualitatively to determine which approach is most effective in supporting farm monitoring through supervised AI.

Quantitative analysis is performed to evaluate the effectiveness of each model. The Target Dataset contains a fraction of data for training and testing, and this analysis should be done on the testing fraction. The evaluation is based on standard performance metrics, which are detailed below. To calculate these metrics, four fundamental quantities are defined.

- True Positive (TP): number of positive instances correctly classified as positive;
- True Negative (TN): number of negative instances correctly classified as negative;
- False Positive (FP): number of negative instances incorrectly classified as positive;
- False Negative (FN): number of positive instances incorrectly classified as negative.

Below are the metrics that will be used:

- **Recall**: characterizes the proportion of actual positive instances that are correctly identified by the model, indicating its effectiveness in detecting truly positive samples:

$$\text{Recall} = \frac{TP}{TP + FN}$$

- **Precision**: calculates the proportion of instances predicted as positive that are actually correct, reflecting the model's ability to avoid false positives:

$$\text{Precision} = \frac{TP}{TP + FP}$$

- **F1 Score**: combines Precision and Recall into a single metric, providing a balanced measure of the model's performance on positive predictions:

$$F1 = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$$

- **Intersection over Union (IoU)**: measures the overlap between the predicted segmentation and the ground-truth segmentation:

$$IoU = \frac{|A \cap B|}{|A \cup B|}$$

  where A is the predicted mask and B is the ground-truth mask.

- **Confusion Matrix**: summarizes, in tabular form, the model's performance in pixel-wise classification by comparing predicted and true labels, indicating how many instances were correctly or incorrectly classified for each class:

$$\begin{bmatrix} TP & FP \\ FN & TN \end{bmatrix}$$

The accuracy metric was not used because, in the semantic segmentation task, it does not accurately represent the model's performance. Accuracy can be influenced by the majority classes, resulting in artificially high values even when the model fails in important classes. Therefore, more appropriate metrics for this type of problem were used, such as IoU, F1-score, and the others already mentioned, which provide a more reliable assessment of the segmentation quality.

In addition to the quantitative evaluation, a **qualitative analysis** is performed to enhance the interpretability of the results. In this analysis, we examine the segmentation maps generated by the model. Based on these maps and the corresponding ground truth, we compute binary error masks that indicate pixel-level misclassifications. Using these error masks, we generate new images by superimposing the errors in red on the original images. This visualization facilitates the identification and interpretation of model errors. Furthermore, we use a confusion matrix to evaluate the classification performance of the model across all classes.

4.2 Experiments

To address the data annotation bottleneck in the semantic segmentation task of aerial UAV images, we conducted experiments following the methodology described previously. Accordingly, this topic is divided into two parts: Data Collection and Preparation and Implementation Details.

To explicitly answer the research question introduced in Section 1.4, "How does pretraining with real-world data from slightly different domains compare to training with synthetic data from the same domain when the target dataset contains only a small number of labeled images?", the experimental design focuses on comparing different pre-training strategies under a low-data regime in the target domain. Specifically, we evaluate: (i) pre-training using synthetic data generated within the same domain as the target dataset, (ii) pre-training using real-world data from a slightly different but related domain, and (iii) a combined strategy using both data sources. All models are subsequently fine-tuned using the same limited set of labeled images from the target dataset, enabling a controlled and fair comparison of how domain similarity and data realism influence segmentation performance.

4.2.1 Dataset Collection and Preparation

Since the focus of this study is to compare pre-training strategies to increase the effectiveness of semantic segmentation in UAV imagery captured in farm-like regions, we choose three publicly available datasets as close as possible to this scenario. The datasets are described in detail below, and a summary can be found in Table 4.1.

1. **Target Dataset**: We select the Switzerland and Okutama (a town in Japan) Dataset (Speth et al., 2022) as our Target Dataset. Although it does not contain typical farmland, the dataset includes farm-like environments characterized by abundant vegetation, structured terrain, and environmentally protected areas. These characteristics make it suitable for evaluating segmentation performance in natural, non-urban environments relevant to agricultural monitoring. We choose this dataset because there is a lack of publicly available datasets specifically focused on real farm environments with detailed semantic annotations. In this context, the Switzerland and Okutama Dataset serves as a realistic and representative alternative, providing scenarios with visual complexity, vegetation density, and terrain variations similar to those found in rural and agricultural areas.

   The dataset contains a limited number of images, comprising 699 UAV images, 355 from Switzerland (resolution 2304×1728) and 344 from Okutama (resolution 1920×1080), collected at flight altitudes between 50 and 90 meters, with the following semantic segmentation labels: Background, Outdoor structures, Buildings, Paved Ground, Non-Paved Ground, Train Tracks, Plants, Wheeled Vehicles, Water, and People. As recommended by the authors, for the semantic segmentation task, we ignore the People class due to its low incidence. We follow the dataset split proposed by the authors using three-fold. Example datasets are

shown in Figure 4.2. The first and second rows contain images from the target dataset, the first from Okutama and the second from Switzerland.

2. **Synthetic Dataset**: Guided by the domain of the data that compose the Target Dataset, that is, UAV images captured at altitudes of 50 to 90 meters in regions such as Switzerland and Okutama, we use as a Synthetic Dataset the work of Silva et al. (2024), an open-source framework for creating photorealistic images in a simulated world, captured during a UAV flight. The framework uses public data, orthophotos, and labeled point clouds collected over the territory, also in Switzerland, to create a virtual world with semantic segmentation labels, where a digital UAV can fly and collect data.

   This virtual world is created using the ROS and Gazebo tools, and it offers control over the flight altitude and the region used to create the virtual world. Using this framework, we generate 17 different maps and collect $\sim 27K$ photorealistic synthetic images labeled at an altitude of 60 meters, with a 800×600-pixel spatial resolution and 10 cm pixel resolution. The generation of these annotations occurs automatically, through the integration between the textured terrain mesh and the positions and orientations of the virtual camera during flight, which are recorded by the simulation system. Specifically, the process consists of projecting the semantic labels, which were previously created from the mapping of the classified cloud point and the textured mesh, onto the camera's position and orientation at each moment of capture, thus producing pixel-by-pixel labels corresponding to the images. This is used as our Synthetic Dataset, hereinafter referred to as SyntheticSwitzerland Dataset (SS). The available semantic segmentation labels are: Unclassified, Ground, Vegetation, Building, and Water.

   We ignore the Unclassified class during training, as this category corresponds to undefined or background regions that do not represent any meaningful semantic class. These areas usually include shadows, image borders, or visually inconsistent textures, which do not exhibit a well-defined visual pattern. Therefore, they are excluded from the loss computation, as detailed later, to prevent the model from learning noise or irrelevant visual information. Examples of this class can be observed in Figure 4.2, on the third row, where the Unclassified regions are annotated in light blue in the segmentation masks.

   We perform the training and testing split at the map level, using maps 1, 2, 4, 5, 14, and 15 as test images ($\sim 9K$) and the remaining ones as training images ($\sim 18K$). The selection of maps used in the dataset division aims to minimize data imbalance and maintain the same data distribution across the divisions. As can be seen in the samples represented in Figure 4.2, the images are remarkably similar to those in the Switzerland and Okutama datasets, presenting farm-like environments characterized

by abundant vegetation, structured terrain, and environmentally protected areas.

3. **Cross-Domain Dataset**: Regarding the Cross-Domain Dataset, since we choose the remote sensing domain, we select the Potsdam dataset (provided by the International Society for Photogrammetry and Remote Sensing (ISPRS)) [1], widely used as a benchmark for semantic segmentation tasks in Remote Sensing. This dataset comprises high-resolution aerial images with a spatial resolution of 5 cm per pixel and 6000×6000-pixel images. Although Potsdam represents an urban environment, it also includes extensive natural areas such as trees, grass, and vegetation surrounding buildings and streets. This mix of natural and built environments makes it suitable for cross-domain pre-training, as it provides visual features relevant to both rural and semi-urban contexts. Although the ISPRS benchmark also provides the Vaihingen dataset, we opted not to use it because its images predominantly depict densely built-up urban areas, offering less diversity in green and semi-natural regions compared to Potsdam.

   The Potsdam dataset images are labeled for the classes: Impervious surfaces, Buildings, Low vegetation, Trees, Cars, and Background or clutter. To approximate the images to the UAV imagery, we crop them using a 512×512-pixel sliding window with 96-pixel overlap, resulting in $\sim 5K$ training images and $\sim 2K$ validation images. As illustrated in Figure 4.2, the images in this dataset share the same viewpoint as the Target dataset; however, they present some discrepancies, especially in terms of light intensity, color saturation, and spatial resolution. Notably, this dataset contains $10\times$ more images than the Target dataset and, similarly, includes farm-like environments.

We compute the mean, standard deviation, and number of pixels available in each class for each training dataset. The mean refers to the average pixel intensity value (per RGB channel) computed across all images in the dataset, and the standard deviation measures the variability of these values. The mean and standard deviation values are presented in Table 4.2 and are used to normalize the images during training. The mean and standard deviation used during training are always extracted from the dataset corresponding to the current training session. In other words, in the case of fine-tuning, we use the statistics (mean and standard deviation) from this fine-tuning dataset for normalization. Using the number of pixels per class, we compute the relative frequency (values can be seen in the table 4.3) to the weight and normalize the loss, due to the class imbalance of the datasets, following Eq. (4.1), where I represents the set of ignored classes, $\tilde{w_c}$ is the logarithmic smoothing function $w_c = (\log(f_c + 1.1))^{-1}$, where c denotes a given class and f_c its relative frequency.

[1] https://www.isprs.org/

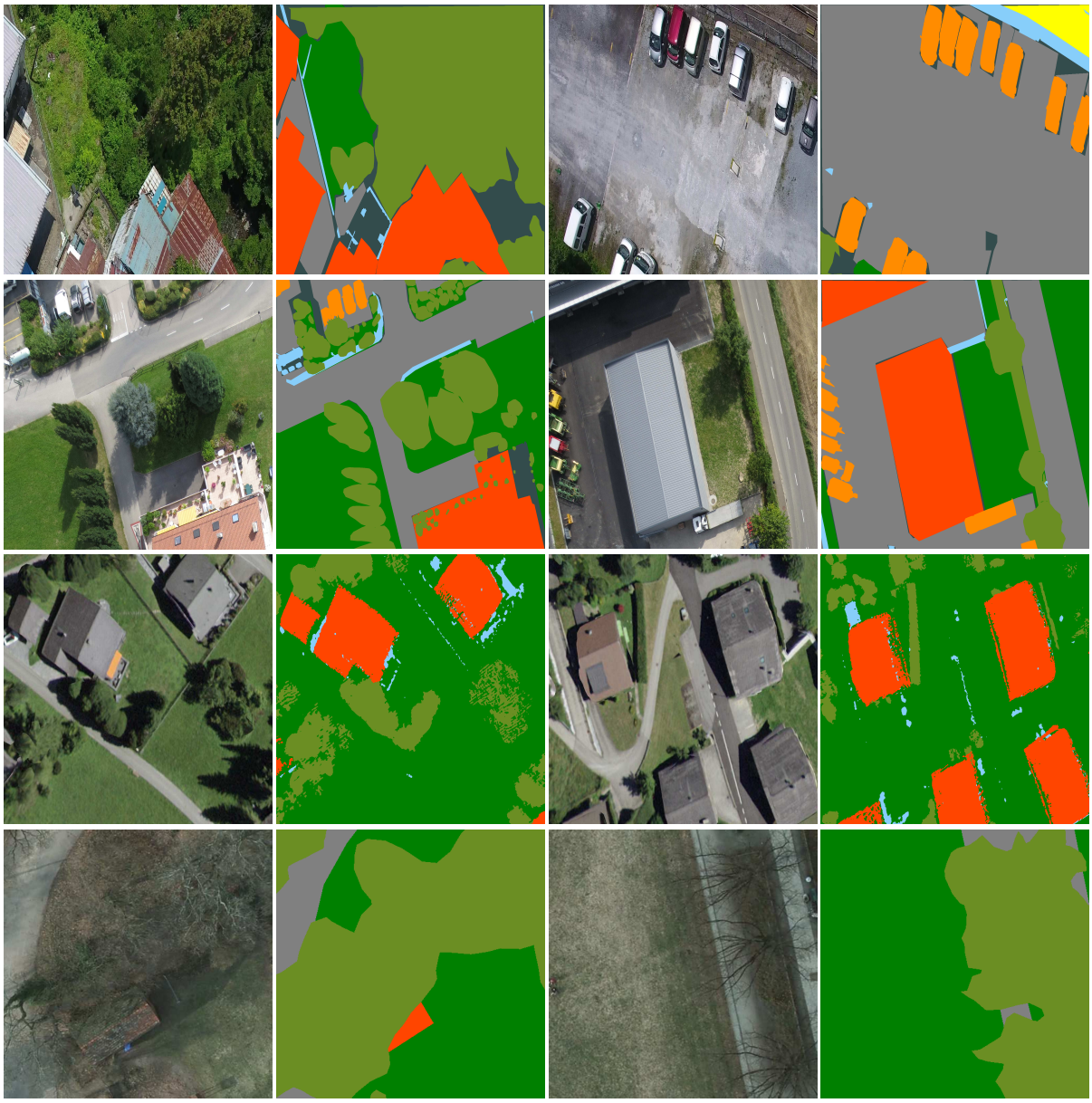

Figure 4.2: Image samples from the selected datasets, with their respective semantic segmentation labels. The datasets in each row are, respectively: Okutama, Switzerland, SS, and Potsdam.

$$\tilde{w}_c = \begin{cases} \frac{w_c}{\max\limits_{k \notin I} w_k}, & \text{if } c \notin I \\ 0, & \text{if } c \in I \end{cases} \tag{4.1}$$

Table 4.1: Comparison of the main characteristics of the Target, Synthetic, and Cross-Domain datasets.

Characteristic	**Target Dataset**	**Synthetic Dataset**	**Cross-Domain Dataset (Potsdam)**
Dataset Name	Switzerland / Okutama	SyntheticSwitzerland (SS)	Potsdam
Source / Domain	Real UAV images (Switzerland and Okutama)	Simulated UAV images (virtual world of Switzerland)	Real aerial images (remote sensing in Potsdam)
Number of Images	699 (355 Switzerland, 344 Okutama)	∼27,000 images	∼7,000 images
Image Resolution	2304×1728 and 1920×1080 px	800×600 px	6000×6000 px (cropped to 512×512)
Flight Altitude	50–90 m	60 m	N/A (5 cm/pixel aerial imagery)
Classes	Background Outdoor Structures Buildings Paved Ground Non-paved Ground Train Tracks Plants Wheeled Vehicles Water People	Unclassified Ground Vegetation Building Water	Impervious surfaces Buildings Low vegetation Trees Water Background or clutter

Table 4.2: Mean and standard deviation for each dataset, remembering that the split of the Swiss/Okutama dataset is three-fold.

Dataset	**Mean**	**Standard Deviation**
SyntheticSwitzerland	0.3552, 0.3858, 0.3131	0.1803, 0.1642, 0.1593
Swiss/Okutama Fold 1	0.4446, 0.4746, 0.3958	0.2249, 0.2101, 0.2402
Swiss/Okutama Fold 2	0.4446, 0.4746, 0.3958	0.2249, 0.2101, 0.2402
Swiss/Okutama Fold 3	0.4600, 0.4784, 0.3895	0.2165, 0.1948, 0.2328
Potsdam	0.3400, 0.3619, 0.3361	0.1380, 0.1365, 0.1419

4.2.2 Implementation Details

The AI model chosen for the experiments is HRNet.OCR (Yuan et al., 2021), a well-established and high-performing architecture for semantic segmentation tasks. HRNet.OCR is derived from HRNet (Wang et al., 2020), a high-resolution network that combines convolutional layers with attention mechanisms through the OCR module. More details can be found in Section 2.4.2.

We perform both pre-training and final training by systematically varying key hyperparameters. These hyperparameters were selected through an empirical tuning

Table 4.3: Relative frequency (%) of classes in each dataset, remembering that the split of the Swiss/Okutama dataset is three-fold.

Dataset	Class	Relative Frequency (%)	Ignored Class
Potsdam	Impervious surfaces	27.29	None
	Buildings	25.30	
	Low vegetation	24.02	
	Trees	17.16	
	Cars	1.67	
	Background	4.56	
Synthetic Switzerland	Unclassified	1.58	Unclassified
	Ground	52.29	
	Vegetation	26.14	
	Building	10.24	
	Water	9.75	
Swiss/Okutama Fold 1	Background	5.50	People
	Outdoor structures	1.58	
	Buildings	14.03	
	Paved Ground	16.52	
	Non-Paved Ground	37.91	
	Train Tracks	1.40	
	Plants	21.13	
	Wheeled Vehicles	1.23	
	Water	0.70	
	People	0.00	
Swiss/Okutama Fold 2	Background	4.72	People
	Outdoor structures	1.39	
	Buildings	14.79	
	Paved Ground	21.82	
	Non-Paved Ground	24.39	
	Train Tracks	1.00	
	Plants	28.30	
	Wheeled Vehicles	3.02	
	Water	0.57	
	People	0.00	
Swiss/Okutama Fold 3	Background	5.49	People
	Outdoor structures	1.48	
	Buildings	15.56	
	Paved Ground	19.51	
	Non-Paved Ground	32.30	
	Train Tracks	0.33	
	Plants	22.29	
	Wheeled Vehicles	2.30	
	Water	0.73	
	People	0.00	

process, in which we tested different configurations and analyzed their effects on the predictions and evaluation metrics. The hyperparameters identified for each pre-training strategy are:

- Synthetic Dataset: Loss function: CE+DICE; Optimizer: Adam; Initial learning rate: 1×10^{-4}; Batch size: 36; Number of epochs: 50.
- Cross-Domain Dataset: Loss function: CE+DICE; Optimizer: Adam; Initial learning rate: 5×10^{-4}; Batch size: 12; Number of epochs: 100.
- Combined: Loss function: CE+DICE; Optimizer: SGD; Initial learning rate: 1×10^{-2}; Batch size: 36; Number of epochs: 50.

For final training on the target dataset, using either pre-trained weights or from-scratch training, we used a CE loss function, an initial LR of 1×10^{-4}, a *ReduceLROnPlateau* scheduler (3-epoch patience, 0.1 reduction factor), and batch sizes of 36 with 50 epochs. We follow the dataset split proposed by the authors using

three-fold, and the values reported are the average over them. The input images are randomly cropped to 512×512-pixels and then resized to 128×128-pixels due to the high GPU memory demands of HRNet.OCR. The experiments are run on a DGX A100, with each training session consuming approximately $\sim$ 70GB of VRAM.

4.3 Results

The quantitative results of the experimental evaluation of the pre-training strategies are summarized in Table 4.4, where the benefit of using data from the same target domain during the pre-training stage is evident compared to cross-domain data, even when the data from the same domain is synthetic and the other domain is only slightly different and composed of real data.

As expected from our sanity check, training the AI model from scratch using only real images from the target dataset produced the lowest performance across all metrics. This result highlights the importance of transfer learning, particularly when working with limited datasets.

Compared to training from scratch, pre-training on the synthetic dataset improved IoU by 5.9 p.p., F1-Score by 6.8 p.p., Recall by 7.7 p.p., and Precision by 10.7 p.p. Pre-training on the cross-domain dataset also outperformed training from scratch, with an increase of 2.8 p.p. in IoU, 0.4 p.p. in F1-Score, 0.2 p.p. in Recall, and 11.3 p.p. in Precision. However, most metrics with this cross-domain pre-training were lower, when compared to pre-training on Synthetic Dataset, by 3,1 p.p. in IoU, 6,4 p.p. in F1-Score and 7,5 p.p. in Recall.

Table 4.4: Per-class results for each metric (%). Best results are in bold. SS stands for the SyntheticSwitzerland Dataset.

	Pre-training Strategy	Backg.	Outd. Struct.	Buildings	Paved Ground	Non P. Ground	Train Tracks	Plants	Vehicles	Water	Mean
IoU	From scratch	14.0	2.7	45.3	39.9	68.2	5.6	71.2	30.7	13.6	32.4
	Potsdam	17.0	1.6	52.0	**60.6**	78.4	0.6	**76.8**	28.0	1.5	35.2
	Potsdam + SS	**17.5**	4.5	53.6	59.3	**79.7**	0.2	76.3	26.0	8.6	36.2
	SS	17.3	**6.3**	**54.1**	52.3	75.0	**14.2**	74.9	**34.8**	**15.9**	**38.3**
F1-Score	From scratch	24.5	5.3	62.2	55.8	80.8	9.6	83.1	45.5	23.1	43.3
	Potsdam	29.0	3.1	68.4	**75.4**	87.6	1.2	**86.8**	39.1	2.9	43.7
	Potsdam + SS	**29.7**	8.3	69.7	74.3	**88.4**	0.4	86.5	36.7	13.8	45.3
	SS	29.5	**11.7**	**70.2**	68.4	85.0	**22.8**	86.3	**51.1**	**26.2**	**50.1**
Recall	From scratch	42.6	3.9	71.5	51.1	69.7	7.8	81.8	57.2	23.6	45.5
	Potsdam	54.7	2.6	**73.4**	67.9	81.3	0.6	**86.4**	42.6	1.5	45.7
	Potsdam + SS	**58.1**	8.1	69.0	**69.3**	**83.1**	0.2	84.3	48.2	9.0	47.7
	SS	55.9	**17.3**	65.6	63.7	76.5	**22.1**	83.9	**62.1**	**31.8**	**53.2**
Precision	From scratch	17.2	8.6	55.1	61.8	96.9	12.7	84.5	37.8	23.9	44.3
	Potsdam	19.7	5.5	64.6	**85.1**	95.1	18.0	87.3	**67.0**	**57.8**	**55.6**
	Potsdam + SS	20.0	**9.5**	71.9	80.7	94.5	**28.5**	**89.0**	52.5	48.1	55.0
	SS	**20.1**	8.9	**75.6**	74.0	**97.2**	23.8	88.9	44.7	43.1	52.9

It is worth noting that the average Precision of the cross-domain pre-training is 2,7 p.p. higher than that of the pre-training with the synthetic dataset; however, the Recall is 7,5 p.p. lower, indicating a considerable false negative rate for this model, especially in the Vehicles and Water classes.

Finally, the metrics of the combined pre-training using the cross-domain and synthetic datasets were higher than those using only the cross-domain dataset but still inferior to those of the pre-training with only the synthetic dataset. This indicates that incorporating synthetic data in the combined strategy helped the model learn visual features more effectively than using only real cross-domain data.

Thus, based on the results of the experimental evaluation, we conclude that using data from the same target domain, even if synthetic, during the pre-training stage is effective in improving the performance of models trained with real-world data. Using synthetic data becomes a better option than using real data from a slightly different domain, which is the case of the top-down UAV image domain related to the high-resolution spatial remote sensing image domain.

Differences between the domains in terms of scale and depth of visual information may explain the poor performance compared to synthetic data from the same domain. Although both the synthetic dataset and the target dataset have similar flight heights (50 to 90 m), high-resolution aerial remote sensing imagery presents a higher apparent altitude.Another factor that may contribute to this difference is the fact that the synthetic images are highly realistic, with quality and colors closer to the target dataset. However, the images from the Potsdam dataset, as illustrated in Figure 4.2, present lower light intensity, color saturation, and spatial resolution when compared to the synthetic dataset.

Analyzing Table 4.4 more closely, we observe that only the Paved Ground and Plants classes performed better in the cross-domain pre-training. For the Paved Ground class, we believe this behavior is due to the fact that the SS dataset only has annotations for the Ground class, being predominantly composed of green areas, while the Potsdam dataset includes the Impervious Surface class, which is more directly related to the Paved Ground class of the target dataset. Figure 4.3 shows an example image containing both green and gray ground areas in the synthetic dataset, along with its annotation where both are labeled as a single class. The second line of this same figure shows the Potsdam dataset that contains a specific class for the gray floor.

Regarding the Plants class, the lower performance observed with synthetic data can be attributed to the presence of artifacts in the representation of tall trees within the synthetic dataset, which hindered the model's performance for this class. In the context of image data, artifacts refer to visual distortions or unrealistic elements introduced during image generation or rendering, such as unnatural textures, irregular shapes, or inconsistencies in color and lighting. These imperfections can mislead the model during training, preventing it from learning accurate visual patterns. Figure 4.4 illustrates some of these artifacts.

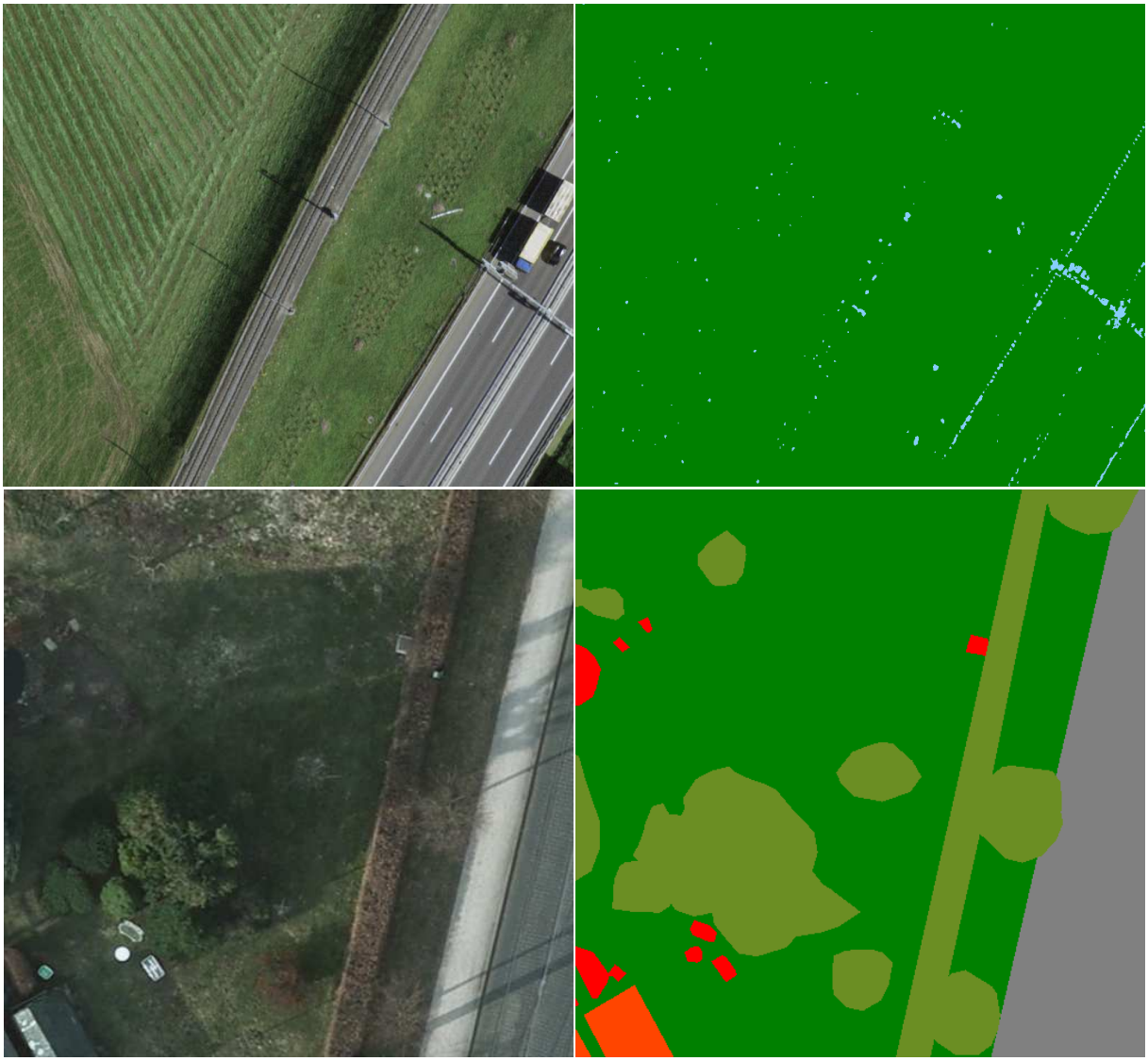

Figure 4.3: Example images demonstrating only the Ground class in the synthetic dataset in the first row. The second row shows the Impervious Surface class for gray ground in the Potsdam dataset.

The qualitative results, shown in Figure 4.5, highlight key points discussed in this study. The first column presents the original image, while the second shows the ground-truth semantic mask. The subsequent columns display the prediction errors overlaid in red on the original image. To generate these overlays, a binary error mask was first created and then applied to the original image. Specifically, the third column corresponds to training from scratch, the fourth to training with the cross-domain pre-training strategy, the fifth to the combined strategy (cross-domain and synthetic), and the sixth to pre-training using only the synthetic dataset.

In this figure, the first row illustrates the superiority of the Synthetic pre-training over the others in all classes present in the image. The second row demonstrates that only the Synthetic pre-training correctly classified all vehicles, which is a relevant class in the farm

Figure 4.4: Example images demonstrating the presence of artifacts in tall trees in the Synthetic Dataset. The first row contains the original images and the second row contains the images with a red circle highlighting some artifacts.

context due to the need to monitor tractors, trucks, buses, and agricultural implements. Cross-domain pre-training fails to capture all vehicles in this example, demonstrating the importance of the data domain. The third row also depicts a scene in which the cross-domain pre-training did not classify any portion of the river correctly, whereas the Synthetic model correctly classified it. The fourth row depicts a scene in which the cross-domain pre-training overcomes the Synthetic in the classes Paved-Ground and Plants, as discussed in the last paragraph.

We performed a zero-shot test on the 699 images of the Swiss/Okutama dataset using weights trained on the SS and Potsdam datasets, and generated the total confusion matrices (Figure 4.6). In this context, a zero-shot test refers to evaluating a model on a dataset it has never seen during training, without any fine-tuning or adaptation, in order to assess its generalization capability.

The goal of this test was to verify whether fine-tuning was necessary on the target dataset or if training solely on synthetic or cross-domain data would be sufficient for direct application to the target data. Since datasets had different class counts, we mapped the model with more classes to the one with fewer by mapping specific classes into broader

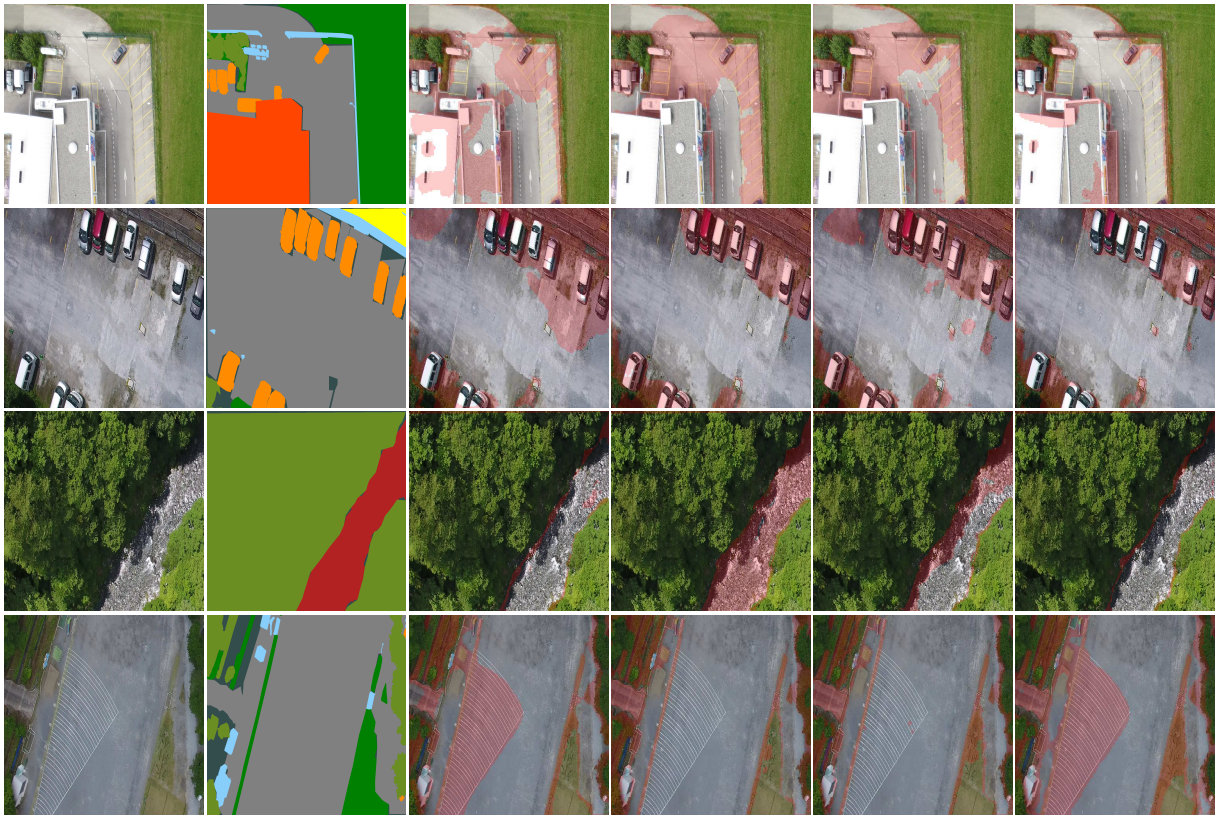

Figure 4.5: Qualitative results, using error overlay (IO) images, using a red mask to indicate the inference error, of the pre-training strategies combined with the final training applied to the target dataset. The columns represent, respectively: original image, ground truth label, IO from training from scratch, IO from pre-training on the cross-domain dataset, IO from the combined pre-training strategy, and IO from pre-training on the synthetic dataset.

ones (example Paved-Ground and Non-Paved-Ground to Ground). Both zero-shot tests demonstrated the necessity of fine-tuning, as the confusion matrices show poor results.

The results presented provide a direct answer to the research question "How does pretraining with real-world data from slightly different domains compare to training with synthetic data from the same domain when the target dataset contains only a small number of labeled images?". When the target dataset contains only a small number of labeled images, pretraining with synthetic data from the same domain consistently outperforms pretraining with real-world data from a slightly different domain.

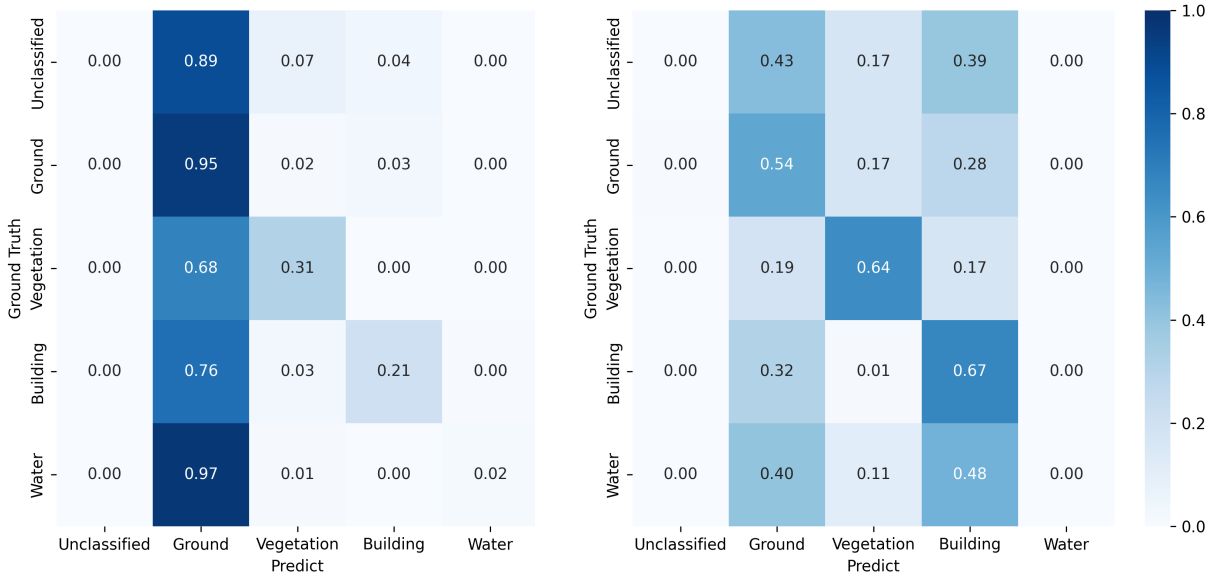

Figure 4.6: Zero-Shot confusion matrix on the target dataset using models trained on the synthetic (left) and cross-domain (right) datasets.

Chapter 5

Object Detection - Integration of Multiple Datasets

5.1 Methodology

To address the need for a large number of labeled images in AI supervised models using ground-based images, this study proposes a new methodology to integrate different datasets. Since a single dataset often does not contain all the classes required for effective farm monitoring and the number of available images is insufficient, we will identify multiple datasets and apply a novel approach to integrate them. The datasets we need to locate must contain classes that help detect different types of animals, including wild animals that may invade the farm and other animals that represent those belonging to the farm itself. In addition, all datasets should include object detection annotations.

So, to enhance object detection in farm contexts and ensure coverage of relevant classes across different datasets, we will follow a structured approach. As shown in Figure 5.1, first, we collect images and annotations from various datasets and we unify this data in a new dataset. Second, we employ a strategy to define models (we propose SmartClass) and select subsets of datasets where the chosen classes are consistently annotated across all datasets in the subset, thereby avoiding samples with unannotated present classes. Finally, we chose an architecture to train the selected models on the object detection task, leveraging transfer learning to develop the SmartClass models. More details of each step are provided below.

5.1.1 Data Preparation

The Data Preparation stage is fundamental to our method, aiming to ensure good class coverage and an adequate number of images. To this end, we divide this stage into two parts: Collecting Images and Converting Labels.

In the Collecting Images step, we focus on identifying and gathering different datasets. To perform terrestrial monitoring of farms and animals using images, we must select

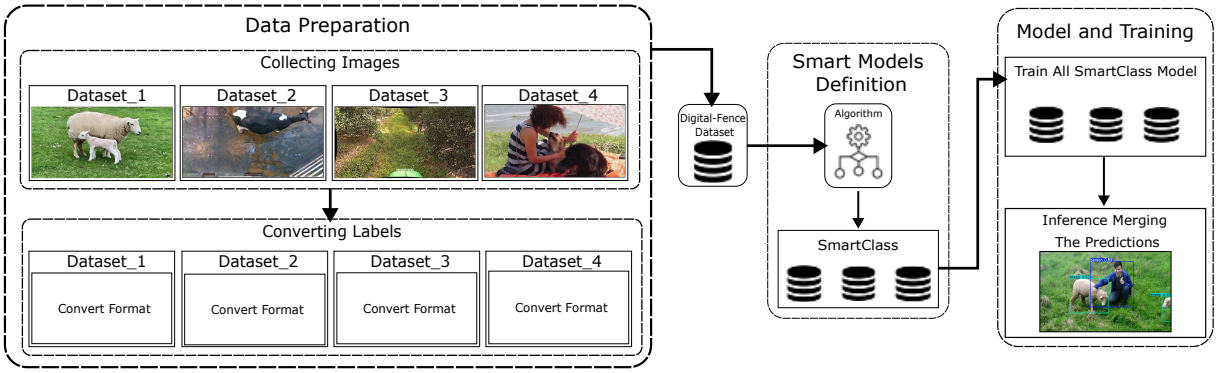

Figure 5.1: Overview of the object detection method - Integration of multiple datasets. It is divided into three stages. In the Data Preparation stage, we locate the datasets and convert the annotations to a single format. This stage outputs the Digital-Fence-Dataset, which is the input for the second stage, Smart Models Definition. This stage is responsible for generating the subsets of the Digital-Fence-Dataset that we call SmartClass models. The last stage, Model and Training, serves to train all the SmartClass models and perform image inferences in each SmartClass model.

datasets that include classes such as people, farm animals (*e.g.*, cows and sheep), as well as animals not typically found in the farm context. The selected datasets may contain only one or two of the desired classes. The animal classes are essential for detecting intruders that may attack farm animals or identifying animals that have wandered into restricted areas. The person class is relevant for recognizing individuals in unauthorized or inappropriate locations. To locate such datasets, public repositories and scientific works that have made relevant datasets available should be explored.

Once the dataset is defined, we perform the Converting Labels step, in which the annotations of each dataset are standardized to a common format. This target annotation standard is defined according to the architecture selected for model training. After establishing this standard, we implement specific conversion algorithms to transform the original annotation format of each dataset into the target format.

Now that all datasets are standardized into a common format, this step results in the Digital-Fence-Dataset, which is the combined dataset containing all images and classes. The organization of this dataset preserves the reference to each original dataset and its respective classes. This information is then used as input for the next stage, Smart Model Definition.

5.1.2 Smart Models Definition

Integrating different datasets is a complex task (Zhou et al., 2022). A significant challenge that can compromise the accuracy of models trained on combined datasets is the inconsistency in class annotations. For example, in some datasets, only animals may be labeled, but images may contain people alongside animals, and people may not be

annotated in this particular dataset. If we combine this dataset with another that contains the annotated person class, this inconsistency can decrease the model's performance for the person class. To address this issue, we proposed a suite of methods collectively termed SmartClass models. These methods are designed to train models in a way that minimizes the negative impact of missing annotations in one dataset, thereby preserving the accuracy of the trained classes.

To define the SmartClass models, we adopt a strategy that selects dataset subsets where the target classes are consistently annotated across all datasets within each subset. Consequently, each subset includes only classes with available labels in the source datasets, preventing the inclusion of samples with unlabeled classes. Our goal is to maximize the number of classes represented in each model of the SmartClass model and ensure that each selected class is well-represented across the largest possible subset of datasets.

For example, consider three datasets, A, B, and C, with the following classes:

- **Dataset A**: Person, Cow
- **Dataset B**: Cow, Dog
- **Dataset C**: Person

Based on these class distributions, the SmartClasses are organized as follows:

- **SmartClass 1 — Person**: Images originating from datasets A and C.
- **SmartClass 2 — Cow**: Images originating from datasets A and B.
- **SmartClass 3 — Dog**: Images originating from dataset B.

This grouping ensures that each SmartClass corresponds to a single semantic category, aggregating all images from different datasets that contain that specific class. If each dataset has a disjoint class, each dataset would represent a SmartClass model. Moreover, the same image may appear in multiple subgroups, but with different annotations. For example, an image containing both people and animals could be included in the people subgroup, with only the people annotation, and also in the animals subgroup, with only the animals annotation.

Algorithm 1 illustrates the procedure for creating the SmartClass models. The algorithm takes as input each dataset and its corresponding set of classes. The outputs are the Selected Models and Selected Classes, which respectively represent the generated sub-datasets and the classes associated with each of them. After initializing the sets of models and selected classes, the algorithm iterates through all possible subsets of classes, ordered from the largest to the smallest. This ordering ensures that the algorithm prioritizes the creation of models covering the greatest number of classes

whenever possible. For each subset, the algorithm checks whether any of its classes have already been selected in a previously defined model. If so, that subset is skipped to avoid redundancy and to ensure that each class belongs to only one SmartClass model. Otherwise, the algorithm identifies the largest possible combination of datasets that share all the classes in the subset. Once this combination is found, the shared classes are added to the list of selected classes, and a new model, defined by this specific set of classes and datasets, is created and added to the collection of SmartClass models. This process continues until all relevant class combinations are evaluated, resulting in a final set of SmartClass models that maximize class consistency across datasets while preventing overlap between models.

Algorithm 1 Model and Class Selection

Input: Set of Classes, Datasets
Output: Selected Models, Selected Classes
$Classes \leftarrow \{person, \ldots\}$
$Models \leftarrow \emptyset$
$Selected\_classes \leftarrow \emptyset$
for each subset S of $Classes$ ordered from largest to smallest **do**
  if there is a class $s \in S$ such that $s \in Selected\_classes$ **then**
    Skip to the next subset
  else
    Identify the subset $T \subseteq S$ such that each element $t \in T$ is present in all datasets of the subset D of $Datasets$, where D is as large as possible.
    Add classes in T to $Selected\_classes$
    Add model $M(T, D)$ to $Models$
  end if
end for

5.1.3 Model and Training

In this stage, we select a well-established architecture for the object detection task. We opted for the YOLO architecture for the experiments in this work, which we will discuss in the next section. Next, training is performed for each SmartClass model generated by the algorithm described in the previous section. Each model is adapted to a specific set of classes. Training is carried out separately for each model, as each one corresponds to a specific dataset and its associated classes. Collectively, all trained models cover the entire set of classes.

After training the SmartClass models, we proceed to the inference phase, in which predictions are made. Each model performs its predictions independently. For instance, if the definition process generates three SmartClass models, the images to be predicted must be passed through all three models. The inference is performed sequentially: the image first passes through the first SmartClass model, which produces predictions for its specific

class. The same image, along with its predicted results, is then passed to the second model, which generates predictions for its own class, and this process continues until all models have been executed. Then the individual predictions from each SmartClass model are combined to produce comprehensive object detection results across all classes.

5.2 Experiments

To address the data annotation bottleneck in the object detection task and to provide a solution for creating digital fences on farms, we conduct experiments following the methodology described in Section 5.1. We select the YOLO architecture for our model training because it is a well-established architecture in the object detection task. More information about this architecture can be found in Section 2.3.

To explicitly answer the second research question presented in Section 1.4, "Is it possible to integrate heterogeneous object detection datasets from rural environments in a way that preserves the learning capacity of the classes and avoids model performance degradation?", the experimental project focuses on evaluating the SmartClass methodology as a strategy for integrating multiple object detection datasets with heterogeneous class definitions and annotation scopes. The experiments compare models trained on two other datasets (Model1 and Model2, which are explained below in the Implementation Details section) with those trained using the SmartClass hierarchical integration, allowing us to assess whether organizing classes into coherent groups can mitigate annotation inconsistencies and preserve detection performance in diverse rural environments.

This topic is divided into two parts: Data Preparation, which describes the collection and preparation of multiple datasets, and Implementation Details, which explains the training process and the development of the SmartClass models.

5.2.1 Data Preparation

To train a model capable of creating digital fences on farms, it is essential to obtain a representative dataset of images depicting such environments. However, due to the high cost of collecting and labeling new data, we rely on existing datasets. We search public repositories and research papers that provided datasets for academic use and identify four that met our study requirements: NREC, Cows2021, APT-36k, and COCO. These datasets allow us to evaluate our dataset aggregation methodology, as they contain the target classes of interest—animals and people. It is important to note that we were unable to find a single dataset that fully represents our use case of farm monitoring from ground-level imagery, which reinforces the need to conduct experiments with our proposed methodology. More details about each dataset are presented below, and a summary is

presented in Table 5.1.

The NREC Person Detection Dataset (Pezzementi et al., 2017) consists of off-road videos captured in orange and apple orchards, primarily focused on person detection. Annotated in the Pascal VOC format, it includes 76,000 images in .png format with a resolution of 720x480 pixels. The annotated classes are Person, Part-Person, and Background. For our work, we exclude images labeled as Background, as our focus is on images containing people. We convert the annotations from VOC to YOLO txt format.

The Cows2021 dataset (Gao et al., 2021) contains aerial images of Holstein-Friesian cows. It was originally designed to identify individual cows based on their unique coat patterns. The dataset is organized into folders, each corresponding to a specific cow, and also provides object detection annotations for all images. The annotations follow the Pascal VOC format, and the images are in .jpg format with a resolution of 1280×720 pixels. We select this dataset to improve the detection of cows viewed from above, considering that this animal is very common in these locations. To be compatible with our methodology, we convert its annotations from Pascal VOC format to YOLO TXT format.

The COCO (Lin et al., 2014a) dataset is one of the largest benchmarks for object detection, segmentation, and captioning tasks, containing more than 200,000 labeled images with annotations for 80 object categories. It is widely used for training models in various detection tasks. We used COCO to ensure the maintenance of person and animal detection in different contexts, as some rural areas may present urban-like characteristics, such as regions near the farm headquarters. As with the other datasets, annotations are converted to YOLO's TXT format to maintain consistency within the Digital-Fence-Dataset.

The APT-36k dataset (Yang et al., 2022) is designed for animal pose estimation and tracking, containing 36,000 images of various animal classes annotated in the COCO format. These classes range from deer to elephants. For our purposes, we selected only the classes cat, dog, horse, sheep, cow, elephant, zebra, and giraffe, which overlap with those from the COCO dataset. The annotations are converted from the COCO format to the YOLO format. This dataset was chosen because it includes other animal classes that may be found on farms or occasionally invade such areas. However, we limit the selection to classes also present in COCO, ensuring at least two different datasets for each included class.

Since none of the available datasets fully met our requirements, we constructed the Digital-Fence-Dataset by combining images from the NREC, Cows2021, APT-36k, and COCO datasets. As mentioned earlier, each dataset originally used its own annotation format, which we converted to a unified format, YOLO, to meet the requirements of the architecture adopted in this work. The resulting dataset includes nine specific classes—person, cat, dog, horse, sheep, cow, elephant, zebra, and giraffe—totaling

Table 5.1: Comparison of the datasets used to build the Digital-Fence-Dataset.

Dataset	Classes Used	Annotation Format	Purpose in Our Work
NREC (Pezzementi et al., 2017)	Person Part-Person	Pascal VOC to YOLO TXT	Person detection in farm and off-road environments
Cows2021 (Gao et al., 2021)	Cow	Pascal VOC to YOLO TXT	Improve cow detection from top-down viewpoints
COCO (Lin et al., 2014a)	Person Cat Dog Horse Sheep Cow Elephant Zebra Giraffe	COCO to YOLO TXT	Maintain detection of people and animals across diverse contexts
APT-36k (Yang et al., 2022)	Cat Dog Horse Sheep Cow Elephant Zebra Giraffe	COCO to YOLO TXT	Complement animal classes
Digital-Fence-Dataset (ours)	Person Cow Dog Horse Sheep Cat Elephant Zebra Giraffe	Unified YOLO TXT	Combined dataset for digital fence training in rural environments

177,566 images.

5.2.2 Implementation Details

In this work, we utilize the Ultralytics framework [1], which includes the YOLOv8 object detection architecture and pre-trained weights from the COCO dataset, which has 80 classes. We apply transfer learning using these pre-trained weights, focusing on nine specific classes: person, cat, dog, horse, sheep, cow, elephant, zebra, and giraffe.

The Digital-Fence-Dataset dataset is used to perform these experiments. For all training sessions, we perform 100 epochs and resize the images to 416×416 pixels. Furthermore, we use the original YOLOv8 weights to initiate training. The optimizer is Stochastic Gradient Descent (SGD), with an initial learning rate (lr) of 0.01 and linear decay over epochs.

The metrics used are Bounding Box Precision (BoxP), Recall, AP50, and AP50–95. BoxP refers to the proportion of positive predictions relative to the total predictions made by the model, evaluating the correctness of a detection. Recall indicates whether

[1] Available: https://docs.ultralytics.com/

the model detects most real objects present in the images. AP50 returns the Average Precision (AP) with an IoU threshold of 0.5, and AP50–95 returns the Average Precision (AP) with an IoU threshold ranging from 0.5 to 0.95 in steps of 0.05.

The initial model trained was **Model** 1, which uses all classes and images from the Digital-Fence-Dataset, except those from the COCO set. We exclude the COCO images because YOLO contains pre-trained weights on this dataset. This approach helps to understand whether a new training run on this data is necessary or not. Although the COCO images are excluded from the training set, they are incorporated into the test set to evaluate model performance and obtain relevant metrics. This approach allows us to assess whether the model training effectively generalizes to more general problems or whether its performance is limited to the specific contexts of the Digital-Fence-Dataset.

The second model trained is **Model** 2, which uses the full range of images and classes in the Digital-Fence-Dataset, including people, cats, dogs, horses, sheep, cows, elephants, zebras, and giraffes. This model is designed to evaluate the effectiveness of directly integrating all available datasets, providing insights into model performance when using a comprehensive and unified data source.

The next models are the **SmartClass** models, which are the models defined by Algorithm 1, demonstrated in the methodology Section 5.1.2. Three specialized datasets are derived from the Digital-Fence-Dataset, resulting in the training of three distinct SmartClass models. The first SmartClass model focuses on human detection, incorporating labeled images from two datasets. The second model is designed for cow detection, using labeled images from three datasets. The third SmartClass model targets the detection of cats, dogs, horses, sheep, elephants, zebras, and giraffes, and is trained with aggregated labeled images from two datasets. The list below summarizes each SmartClass, along with its classes and the original datasets from which the images are obtained.

Images representing each SmartClass model are shown in Figure 5.2. The first line of the image contains only the class of people, although we can observe that it contains a cow in the first image along with a person; only the class of people is annotated in SmartClass_1. The second line of the image shows the class of cows, and the third line shows the other animals.

- **SmartClass_1:** person class — derived from the COCO and NREC datasets, comprising approximately 143,000 images.
- **SmartClass_2:** cow class — derived from the APT36k, COCO, and Cows2021 datasets, comprising approximately 13,000 images.
- **SmartClass_3:** sheep, dog, elephant, cat, horse, giraffe, and zebra classes — derived from the APT36k and COCO datasets, comprising approximately 40,000 images.

Table 5.2: Summary of each trained model: Model 1, Model 2, and the SmartClass Models.

	Model 1	**Model 2**	**SmartClass**		
			SmartClass_1	**SmartClass_2**	**SmartClass_3**
Datasets	NREC Apt36k Cows2021	NREC Apt36k Cows2021 COCO	COCO NREC	APT36k COCO Cows2021	APT36k COCO
Classes	sheep, dog, elephant, cat, horse, giraffe, cow, zebra, person	sheep, dog, elephant, cat, horse, giraffe, cow, zebra, person	person	Cow	sheep, dog, elephant, cat, horse, giraffe, zebra
Notes	In inference we added COCO dataset to validate more generic contexts		∼143k images	∼13k images	∼40k images

Table 5.3: Comparison of object detection metrics across different models and object classes. Best in bold.

Class	**Box Precision**			**Recall**			**AP50**			**AP50-95**		
	Model 1	**Model 2**	**SmartClass**	**Model 1**	**Model 2**	**SmartClass**	**Model 1**	**Model 2**	**SmartClass**	**Model 1**	**Model 2**	**SmartClass**
person	***0.977***	0.901	0.918	0.543	0.758	***0.784***	0.669	0.851	***0.875***	0.439	0.56	***0.572***
cat	***0.971***	0.837	0.953	0.568	0.86	***0.925***	0.649	0.899	***0.958***	0.543	0.735	***0.812***
dog	***0.961***	0.874	0.917	0.616	0.757	***0.871***	0.721	0.861	***0.921***	0.586	0.685	***0.762***
horse	0.926	0.911	***0.945***	0.606	0.78	***0.825***	0.7	0.86	***0.905***	0.573	0.678	***0.741***
sheep	***0.941***	0.849	0.913	0.474	0.695	***0.746***	0.582	0.802	***0.859***	0.503	0.627	***0.679***
cow	***0.971***	0.935	0.959	0.768	0.854	***0.886***	0.864	0.927	***0.949***	0.756	0.779	***0.815***
elephant	***0.93***	0.897	0.922	0.637	0.871	***0.899***	0.748	0.929	***0.954***	0.619	0.734	***0.812***
zebra	0.944	0.922	***0.976***	0.778	0.873	***0.893***	0.848	0.94	***0.961***	0.691	0.756	***0.805***
giraffe	***0.97***	0.926	0.969	0.691	0.873	***0.905***	0.805	0.927	***0.957***	0.681	0.759	***0.822***
all classes	***0.96***	0.89	0.94	0.62	0.81	***0.86***	0.76	0.88	***0.93***	0.61	0.70	***0.79***

Table 5.2 provides an overview of each trained model. We trained Model 1, Model 2, and SmartClass. Training Model 1 and Model 2 serves as a baseline for comparison with SmartClass, as these two models are trained using the grouped datasets, without applying the proposed methodology.

5.3 Results

Object detection can play an important role in enabling digital fence systems for monitoring farms. This section reports the results obtained by applying the SmartClass methodology to mitigate the lack of annotated data in this field.

The table 5.3 presents the performance metrics for each class in the three trained

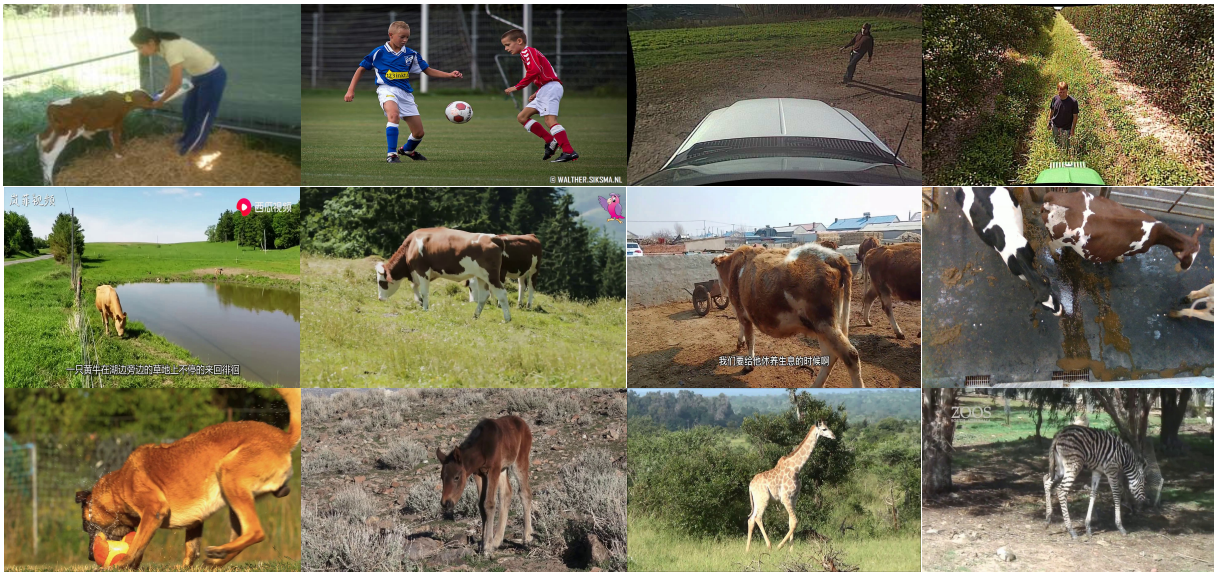

Figure 5.2: Sample images for each SmartClass model. The first row contains images from SmartClass_1, the second from SmartClass_2, and the third from SmartClass_3.

models: Model 1, Model 2, and SmartClass, explained in the Implementation Details subsection 5.2.2 of the Experiments section. This table presents the values of BoxP, Recall, AP50, and AP50–95 metrics for the three models. The values are shown per class: person, cat, dog, horse, sheep, cow, elephant, zebra, and giraffe.

Model 1 consistently underperformed the other models, with the exception of the BoxP metric, which ranged from 0.926 (horse) to 0.97 (giraffe). This high precision

Figure 5.3: Model 1 results in the first row, Model 2 in the second row and third row with SmartClass model.

indicates that Model 1 is accurate when detecting objects, but its low Recall scores reveal a significant limitation: the model fails to identify or predict all objects present in the images. This suggests that, although the model performs well in the original context in which it was trained, it has difficulty generalizing to new or varied contexts. Furthermore, the low AP50 and AP50–95 values confirm its lack of robustness across different scenarios.

Model 2 was trained on all classes of the Digital-Fence-Dataset (person, cat, dog, horse, sheep, cow, elephant, zebra, giraffe). It showed BoxP ranging from 0.837 (cat) to 0.935 (cow), and Recall ranging from 0.695 (sheep) to 0.873 (zebra, giraffe). The AP50 metrics ranged from 0.802 for sheep to 0.929 for elephant, while the AP50–95 values ranged from 0.56 for the person class to 0.779 for the cow class. Overall, Model 2 shows improvement over Model 1 in most metrics, especially in Recall and AP metrics. However, the values remain relatively low, with the exception of BoxP, indicating that while the model accurately localizes objects when it detects them, it still struggles to detect objects in different contexts. This suggests that while Model 2 performs better in object recognition and localization, there is still room for improvement in the model's robustness across multiple scenarios.

As shown in Table 5.3, the SmartClass models consistently outperformed Models 1 and 2 across all classes, especially in the Recall (R), AP50, and AP50–95 metrics. This indicates that the SmartClass approach significantly improves object detection performance across a variety of contexts. However, SmartClass models performed slightly worse on the BoxP metric. This is likely because by detecting more objects, SmartClass models are also more likely to find cases where the bounding boxes are less precise, leading to a slight decrease in accuracy.

Figure 5.3, Row 1, shows the poor performance of Model 1 on images outside its training context. In all three images shown in Row 1, the model failed to detect any people, identifying only animals. This result confirms that a model trained in a specific context struggles to generalize effectively to different contexts, as evidenced by its inability to detect people in images that differ from the original training environment.

Row 2 of Figure 5.3 illustrates the performance of Model 2 on images that did not include the person label during training. In the first image in Row 2, no detection was performed. In the second image, a detection occurred, but with a low confidence score of 0.32. The third image showed a detection with a higher confidence score of 0.69. These results suggest that unlabeled objects in the dataset can negatively impact model performance, leading to variable detection quality and reduced overall effectiveness.

For SmartClass, we observed significant improvements in person detection in a variety of contexts, including challenging environments such as rural areas and agricultural fields, where individuals may be camouflaged or in varying positions, such as crouching or lying down. Row 3 in Figure 5.3 highlights how the SmartClass model improves person detection compared to Model 1 and Model 2. In the three images shown in this row, SmartClass

achieves confidence scores above 0.85 for each detection. SmartClass also performed well in detecting cows and other animals. Figure 5.3 also shows the detection performance for sheep and cows in Models 1, 2, and SmartClass, highlighting the effectiveness of SmartClass in these categories as well.

These results directly answer the second research question, "Is it possible to integrate heterogeneous object detection datasets from rural environments in a way that preserves the learning capacity of the classes and avoids model performance degradation?", by demonstrating that it is possible to integrate heterogeneous object-detection datasets from rural-like environments without degrading model performance. The consistent improvements observed in Recall, AP50, and AP50–95 indicate that the SmartClass methodology effectively preserves class learnability while reducing ambiguity caused by heterogeneous annotations. By structuring classes hierarchically and avoiding exposure to incomplete or conflicting labels, SmartClass enables the model to benefit from multiple datasets simultaneously, resulting in more robust and generalizable object detection.

To evaluate the computational cost of the SmartClass method, we measured exclusively the inference time of each of the three specialized models generated by the hierarchical classification process. In the experiment, the same set of three test images was sequentially processed by the three models, SmartClass_1, SmartClass_2, and SmartClass_3, regardless of the classification outcome at each stage, ensuring that the full hierarchical pipeline was executed. The cumulative inference times recorded for SmartClass_1, SmartClass_2, and SmartClass_3 were 0.0440 s, 0.0434 s, and 0.0480 s, respectively. The sum of these values corresponds to a total inference time of 0.1354 s, representing the computational cost strictly associated with the forward pass of the neural networks. The experiment was conducted on a workstation equipped with an NVIDIA GeForce RTX 4090 GPU (24 GB of memory), with the GPU under low load (approximately 600 MB of memory usage at the time of measurement), ensuring that the reported inference times reflect the intrinsic performance of the SmartClass models.

Additionally, Figure 5.4 illustrates person detection in agricultural fields. The first row displays detections using the YOLO model originally trained on the COCO Dataset, while the second row presents the results from SmartClass, demonstrating consistently good performance in person detection in challenging scenarios.

Figure 5.4: Object detection in a rural environment. First line prediction made with original YOLOv8 training and second line with SmartClass.

Chapter 6

Conclusion

Monitoring farms and agricultural areas is essential for effective management and the safety of crops, people, and animals. The use of AI for analyzing images captured in these environments is a valuable ally; however, supervised models require labeled data, which is costly to produce due to the time and labor involved in image collection and annotation.

This work explored two complementary research directions to support agricultural monitoring through AI and to address the scarcity of labeled data. The tasks investigated were semantic segmentation in aerial UAV images and object detection in terrestrial images captured at specific locations. These two tasks proved essential for monitoring, as they provide the foundation for mapping areas, detecting intrusions, identifying animal deviations, and supporting other related applications. The first part of our methodology investigated the use of synthetic data and real data from slightly different domains as pre-training sources for semantic segmentation models applied to aerial imagery. The second part proposed a methodology for integrating multiple datasets in object detection tasks, addressing challenges related to differing annotation formats and class definitions across datasets, thereby providing the means to support the development of digital fences.

For the first line of research, our results demonstrated that synthetic data generated within the same domain can significantly improve model performance compared to training from scratch or pre-training with real data from slightly different domains (cross-domain transfer). These findings underscore the importance of domain alignment in transfer learning for semantic segmentation tasks and emphasize that synthetic data can be highly beneficial for model initialization, particularly when real data is limited. This approach also helps reduce data collection and annotation costs. As a limitation, it is worth noting that the experiments were conducted in specific scenarios covering regions of Japan, Switzerland, and Germany, and therefore may not be fully generalizable to other geographical or environmental conditions.

Regarding the second line of research, we introduced the Digital-Fence-Dataset, an aggregated dataset that combines elements from the NREC, Cows2021, APT-36k, and COCO datasets. Our results showed that, while combining datasets is beneficial for addressing coverage gaps, careful management is essential to avoid confusion during model

training. Unlabeled or partially labeled images within the dataset can mislead the model and compromise performance.

To mitigate this issue, we employed the SmartClass approach, which involves training models with fractional class groups to reduce confusion and improve detection accuracy. Experiments using the COCO, NREC, Cows2021, and APT-36k datasets demonstrated that integrating these datasets with the SmartClass methodology led to improvements in quantitative metrics and superior qualitative results. Consequently, the SmartClass-based approach proved effective in supporting the development of digital security fences for rural environments. As a limitation, it is worth noting that this approach may require multiple models to cover all classes instead of relying on a single comprehensive model, which increases processing demand and computational cost. Furthermore, since each model maintains its own accuracy level, interpretability challenges may arise regarding the confidence of detections.

These two approaches complement each other in enhancing the safety of rural environments: while aerial monitoring enables the mapping of large areas and the identification of critical points, object detection from strategically placed ground cameras assists in creating digital fences and issuing alerts about intrusions by people, animals, or other risks. Both studies showed promising results in overcoming the data annotation bottleneck and improving the performance of supervised AI models—both in semantic segmentation applied to aerial images and in object detection applied to ground images—thus contributing to the advancement of intelligent agricultural monitoring.

In the introduction to this work, we presented two research questions that can now be answered. The first one is: how does pretraining with real-world data from slightly different domains compare to training with synthetic data from the same domain when the target dataset contains only a small number of labeled images? The results obtained in this work allow us to answer the first research question: pretraining models with synthetic data from the same domain proved to be more effective than using real data from slightly different domains, especially when the target dataset contains only a small number of labeled images. Models pretrained with synthetic data achieved higher performance and showed less class confusion, demonstrating that domain alignment, even when achieved through simulation, has a stronger impact on knowledge transfer than real images. This reinforces that visual and semantic proximity between source and target domains is a key factor for successful fine-tuning under limited annotation conditions.

The second question is: is it possible to integrate heterogeneous object-detection datasets from rural-like environments in a way that preserves class learnability and avoids degrading model performance? Regarding the second research question, this work showed that it is indeed possible to integrate heterogeneous object-detection datasets from rural-like environments without degrading performance, as long as the integration

process preserves class consistency and avoids exposing the model to incomplete or conflicting annotations. The SmartClass methodology proved to be an effective solution for this scenario, as it organizes classes into coherent groups, reduces ambiguity, and enables the model to learn from multiple datasets without performance loss. Thus, we confirm that dataset integration is both feasible and beneficial, provided that it is accompanied by specific strategies to control data heterogeneity.

As future work, we intend to extend the SmartClass methodology by training a single shared backbone while adapting only the classification head for each hierarchical level. This strategy is expected to significantly reduce computational overhead, memory consumption, and inference latency, since feature extraction would be performed only once per input image, overcoming the current limitation of sequential inference with multiple specialized models. Furthermore, a unified model can facilitate deployment in resource-constrained environments and improve scalability when new classes or hierarchical levels are introduced.

In addition, we plan to address the current frame-by-frame video processing limitation by incorporating object tracking and re-identification techniques. By associating detections across consecutive frames, the system will be able to preserve object identities over time, resulting in more stable predictions, reduced temporal inconsistencies, and improved robustness in dynamic scenes. This extension is particularly relevant for real-world monitoring scenarios, where temporal continuity is essential for reliable event analysis and decision-making.

Bibliography

Akhil, V., Hareesh, R., Rahul, P., Yesudas, Y., Menon, H., and Sebin, P. (2023). Virtual fencing to prevent human wild conflict. In *Inter. Conf. on Innovations in Engineering and Technology (ICIET)*.

Albarracin, J., Cano, F., Romero, E., and Cruz-Roa, A. (2023). A comparative analysis between two convolutional networks architectures for semantic segmentation of histopathology breast cancer images. In *2023 19th International Symposium on Medical Information Processing and Analysis (SIPAIM)*, pages 1–5.

Ali, N., Ijaz, A. Z., Ali, R. H., Ul Abideen, Z., and Bais, A. (2023). Scene parsing using fully convolutional network for semantic segmentation. In *2023 IEEE Canadian Conference on Electrical and Computer Engineering (CCECE)*, pages 180–185.

Anil, J. M., Mathews, L., Renji, R., Jose, R. M., and Thomas, S. (2023). Vehicle counting based on convolution neural network. In *2023 7th International Conference on Intelligent Computing and Control Systems (ICICCS)*, pages 695–699.

Arezoomandan, S., Klohoker, J., and Han, D. K. (2024). Analyzing the efficacy of synthetic images in unmanned aerial vehicle detection. In *IEEE International Conference on Consumer Electronics (ICCE)*.

Balaji, Y., Sankaranarayanan, S., and Chellappa, R. (2018). Metareg: Towards domain generalization using meta-regularization. In *Neural Information Processing Systems*.

Blaga, B. C. Z. and Nedevschi, S. (2022). Forest inspection dataset for aerial semantic segmentation and depth estimation. In *arXiv Pre-Print*.

Bochkovskiy, A., Wang, C.-Y., and Liao, H.-Y. M. (2020). Yolov4: Optimal speed and accuracy of object detection.

Bordado, M., Silveria, D., and Laureano, G. (2023). Ball detection and tracking with different embedded systems in the robocup soccer context. In *2023 Latin American Robotics Symposium (LARS), 2023 Brazilian Symposium on Robotics (SBR), and 2023 Workshop on Robotics in Education (WRE)*, pages 514–519.

Caesar, H., Uijlings, J., and Ferrari, V. (2018). Coco-stuff: Thing and stuff classes in context. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1209–1218.

Cao, H., Wang, Y., Chen, J., Jiang, D., Zhang, X., Tian, Q., and Wang, M. (2021). Swin-unet: Unet-like pure transformer for medical image segmentation.

Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., and Zagoruyko, S. (2020). *End-to-End Object Detection with Transformers*, pages 213–229.

Changzhen, X., Cong, W., Weixin, M., and Yanmei, S. (2016). A traffic sign detection algorithm based on deep convolutional neural network. In *2016 IEEE International Conference on Signal and Image Processing (ICSIP)*, pages 676–679.

Chen, L.-C., Papandreou, G., Kokkinos, I., Murphy, K., and Yuille, A. L. (2014). Semantic image segmentation with deep convolutional nets and fully connected crfs. *arXiv preprint arXiv:1412.7062*.

Chen, L.-C., Papandreou, G., Kokkinos, I., Murphy, K., and Yuille, A. L. (2017a). Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, volume 40, pages 834–848.

Chen, L.-C., Papandreou, G., Schroff, F., and Adam, H. (2017b). Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv:1706.05587*.

Chen, L.-C., Zhu, Y., Papandreou, G., Schroff, F., and Adam, H. (2018). Encoder-decoder with atrous separable convolution for semantic image segmentation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 801–818.

Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., and Schiele, B. (2016). The cityscapes dataset for semantic urban scene understanding.

Delwar, T. S., Mukhopadhyay, S., Kumar, A., Singh, M., Lee, Y.-w., Ryu, J.-Y., and Hosen, A. S. M. S. (2025). Real-time farm surveillance using iot and yolov8 for animal intrusion detection. *Future Internet*, 17(2).

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., and Houlsby, N. (2021). An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations (ICLR)*.

Elaoua, A., Nadour, M., Cherroun, L., and Elasri, A. (2023). Real-time people counting system using yolov8 object detection. In *2023 2nd International Conference on Electronics, Energy and Measurement (IC2EM)*, volume 1, pages 1–5.

Elhammamy, Y. Y. S., Chung, G. C., Gan, M. T., Tiang, J. J., and Teong, K. V. (2024). Classification and detection of rice crop using deep learning for smart agriculture. In *2024 Multimedia University Engineering Conference (MECON)*, pages 1–6.

Elhassan, M. A., Zhou, C., Khan, A., Benabid, A., Adam, A. B., Mehmood, A., and Wambugu, N. (2024). Real-time semantic segmentation for autonomous driving: A review of cnns, transformers, and beyond. *Journal of King Saud University - Computer and Information Sciences*, 36(10):102226.

Everingham, M., Van Gool, L., Williams, C. K., Winn, J., and Zisserman, A. (2010). The pascal visual object classes (voc) challenge. *International Journal of Computer Vision*, 88(2):303–338.

G, G., S, B., N, M., S, D., M, P. K., and T, A. (2025). Harm-aware iot-powered smart digital fencing framework for real-time intrusion detection and mitigation using solar energy. In *2025 2nd International Conference on Research Methodologies in Knowledge Management, Artificial Intelligence and Telecommunication Engineering (RMKMATE)*, pages 1–7.

Gandhi, R., Gupta, A., Yadav, A. K., and Rathee, S. (2022). A novel approach of object detection using deep learning for animal safety. In *2022 12th International Conference on Cloud Computing, Data Science Engineering (Confluence)*, pages 573–577.

Ganin, Y., Ustinova, E., Ajakan, H., Germain, P., Larochelle, H., Laviolette, F., Marchand, M., and Lempitsky, V. (2016a). Domain-adversarial training of neural networks.

Ganin, Y., Ustinova, E., Ajakan, H., Germain, P., Larochelle, H., Laviolette, F., Marchand, M., and Lempitsky, V. (2016b). Domain-adversarial training of neural networks.

Gao, J., Burghardt, T., Andrew, W., Dowsey, A. W., and Campbell, N. W. (2021). Towards self-supervision for video identification of individual holstein-friesian cattle: The cows2021 dataset. *arXiv preprint arXiv:2105.01938*.

Gao, T., Packer, B., and Koller, D. (2011). A segmentation-aware object detection model with occlusion handling. In *CVPR 2011*, pages 1361–1368.

Ge, J., Zhang, Z., Phan, M. H., Zhang, B., Liu, A., and Zhao, Y. (2024). Esa: Annotation-efficient active learning for semantic segmentation. In *arXiv Pre-Print*.

Giang, T. L., Dang, K. B., Toan Le, Q., Nguyen, V. G., Tong, S. S., and Pham, V.-M. (2020). U-net convolutional networks for mining land cover classification based on high-resolution uav imagery. *IEEE Access*.

Girshick, R. (2015). Fast r-cnn.

Girshick, R., Donahue, J., Darrell, T., and Malik, J. (2014). Rich feature hierarchies for accurate object detection and semantic segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 580–587.

Gkioxari, G., Girshick, R., Dollár, P., and He, K. (2018). Detecting and recognizing human-object interactions. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8359–8367.

Gretton, A., Borgwardt, K. M., Rasch, M. J., Schölkopf, B., and Smola, A. (2012). A kernel two-sample test. In *Journal of Machine Learning Research*, volume 13, pages 723–773.

Gruszczyński, W., Puniach, E., Ćwiąkała, P., and Matwij, W. (2022). Correction of low vegetation impact on uav-derived point cloud heights with u-net networks. *IEEE Transactions on Geoscience and Remote Sensing*.

Heschl, A., Murillo, M., Najafian, K., and Maleki, F. (2024). Synthset: Generative diffusion model for semantic segmentation in precision agriculture.

Hidayaturrahman, Trisetyarso, A., Herwidiana Kartowisastro, I., and Budiharto, W. (2024). Adversarial multitask learning for domain adaptation through domain adapter. *IEEE Access*, 12:184989–184999.

Hoffman, J., Tzeng, E., Park, T., Zhu, J.-Y., Isola, P., Saenko, K., Efros, A. A., and Darrell, T. (2017). Cycada: Cycle-consistent adversarial domain adaptation.

Houlsby, N., Giurgiu, A., Jastrzebski, S., Morrone, B., de Laroussilhe, Q., Gesmundo, A., Attariyan, M., and Gelly, S. (2019). Parameter-efficient transfer learning for nlp.

Hsu, C.-K., Chiu, Y.-H., Wu, K.-R., Liang, J.-M., Chen, J.-J., and Tseng, Y.-C. (2019). Design and implementation of image electronic fence with 5g technology for smart farms. In *VTS Asia Pacific Wireless Communications Symposium (APWCS)*.

Husain, S. M. A., Ahmad, S. Y., Aziz, A., and Sohail, S. S. (2022). Drone for agriculture: A way forward. In *International Conference on Data Analytics for Business and Industry (ICDABI)*.

Islam, M. M., Chowdhury, I. J., Mahboob, T. Z., Mazumder, M. S. J., Hossain, M. J., Biswas, M. S., and Rone, P. D. (2024). A comprehensive review on object detection in the context of autonomous driving. In *2024 4th International Conference on Ubiquitous Computing and Intelligent Information Systems (ICUIS)*, pages 1860–1864.

Kent, O. W., Chun, T. W., and Choo, T. L. (2023). Deep learning approach for detection of oil palm tree on uav images. In *IEEE International Conference on Automatic Control and Intelligent Systems (I2CACIS)*.

Kim, D., Tsai, Y.-H., Suh, Y., Faraki, M., Garg, S., Chandraker, M., and Han, B. (2022). Learning semantic segmentation from multiple datasets with label shifts.

Klein, J., Waller, R., Pirk, S., Pałubicki, W., Tester, M., and Michels, D. L. (2024). Synthetic data at scale: a development model to efficiently leverage machine learning in agriculture. *Frontiers in Plant Science*.

Kortylewski, A., Liu, Q., Wang, A., Sun, Y., and Yuille, A. (2020). Compositional convolutional neural networks: A robust and interpretable model for object recognition under occlusion.

Koudia, A., Chouaba, S. E., and Chouaib Belkhiat, D. E. (2022). Implementation and comparison of u-net networks for automatic covid-19 lung infection segmentation. In *International Multi-Conference on Systems, Signals Devices (SSD)*.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2017). Imagenet classification with deep convolutional neural networks. *Commun. ACM*, 60(6):84–90.

Kumar, S., Zhang, B., Gudavalli, C., Levenson, C., Hughey, L., Stabach, J. A., Amoke, I., Ojwang, G., Mukeka, J., Mwiu, S., Ogutu, J., Frederick, H., and Manjunath, B. (2024). Wildlifemapper: Aerial image analysis for multi-species detection and identification. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12594–12604.

Lecun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324.

Li, D., Yang, Y., Song, Y.-Z., and Hospedales, T. M. (2018). Learning to generalize: Meta-learning for domain generalization. In *AAAI Conference on Artificial Intelligence*.

Li, X., Diao, W., Mao, Y., Gao, P., Mao, X., Li, X., and Sun, X. (2023). Ogmn: Occlusion-guided multi-task network for object detection in uav images. *ISPRS Journal of Photogrammetry and Remote Sensing*, 199:242–257.

Liao, Z., Peng, H., and Liu, T. (2023). Brain tumor segmentation based on improved swin-unet. In *International Conference on Artificial Intelligence and Intelligent Information Processing (AIIIP)*.

Lin, T.-Y., Goyal, P., Girshick, R., He, K., and Dollár, P. (2020). Focal loss for dense object detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42(2):318–327.

Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., and Zitnick, C. L. (2014a). Microsoft coco: Common objects in context. In *ECCV*.

Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., and Zitnick, C. L. (2014b). Microsoft coco: Common objects in context. In *European Conference on Computer Vision (ECCV)*, pages 740–755. Springer.

Liu, F. and Fang, M. (2020). Semantic segmentation of underwater images based on improved deeplab. *Journal of Marine Science and Engineering*, 8(3).

Liu, R., Wei, J., Liu, F., Si, C., Zhang, Y., Rao, J., Zheng, S., Peng, D., Yang, D., Zhou, D., and Dai, A. M. (2024). Best practices and lessons learned on synthetic data. In *arXiv Pre-Print*.

Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y., and Berg, A. C. (2016). *SSD: Single Shot MultiBox Detector*, page 21–37. Springer International Publishing.

Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., and Guo, B. (2021a). Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 10012–10022.

Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., and Guo, B. (2021b). Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*.

Long, J., Shelhamer, E., and Darrell, T. (2015). Fully convolutional networks for semantic segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3431–3440.

Long, M., Cao, Z., Wang, J., and Jordan, M. I. (2018). Conditional adversarial domain adaptation.

Lu, T. and Zhu, C. (2022). Reshaping the semantic logits for proposal-free panoptic segmentation. In *IEEE International Conference on Tools with Artificial Intelligence (ICTAI)*.

Mahmud, M. N., Osman, M. K., Ismail, A. P., Ahmad, F., Ahmad, K. A., and Ibrahim, A. (2021). Road image segmentation using unmanned aerial vehicle images and deeplab v3+ semantic segmentation model. In *2021 11th IEEE International Conference on Control System, Computing and Engineering (ICCSCE)*, pages 176–181.

Mamat, N., Othman, M. F., and Yakub, F. (2022). Animal intrusion detection in farming area using yolov5 approach. In *Inter. Conf. on Control, Automation and Systems (ICCAS)*.

Miletic, M. and Sariyar, M. (2024). Challenges of using synthetic data generation methods for tabular microdata. *Applied Sciences*, 14(14).

Milioto, A., Lottes, P., and Stachniss, C. (2018). Real-time semantic segmentation of crop and weed for precision agriculture robots leveraging background knowledge in cnns. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2229–2235.

Motoi, I. M., Belli, V., Carpineto, A., Nardi, D., and Ciarfuglia, T. A. (2025). Synthetic data generation for anomaly detection on table grapes. *Smart Agricultural Technology*.

Nam, H. and Han, B. (2016). Learning multi-domain convolutional neural networks for visual tracking.

Naveenkumar, M., Manoj, R., Nandhakumar, B., and Rahul, R. (2022). Densenet201 for animal detection and repellent system. In *Inter. Conf. on Electronic Systems and Intelligent Computing (ICESIC)*.

Nazeer, I., Umer, S., Rout, R. K., and Tanveer, M. (2024). Artificial intelligence-based smart agricultural systems for saffron cultivation with integration of unmanned aerial vehicle imagery and deep learning approaches. *Computers and Electrical Engineering*, 119:109542.

Neigel, P., Ameli, M., Katrolia, J., Feld, H., Wasenmüller, O., and Stricker, D. (2020). Opedd: Offroad pedestrian detection dataset. *Journal of WSCG*.

Nishida, Y., Li, Y., and Kamiya, T. (2021). Environment recognition from a spherical camera image based on deeplab v3+. In *2021 21st International Conference on Control, Automation and Systems (ICCAS)*, pages 2043–2046.

Nugraha, P. Z. E. S., Sunarya, I. M. G., and Maysanjaya, I. M. D. (2023). Binary semantic segmentation of dolphin on uav image using u-net. In *International Seminar on Intelligent Technology and Its Applications (ISITIA)*.

Nurul Qomariah, D. U., Tjandrasa, H., and Alam, B. R. (2021). Hemorrhage segmentation in retinal images using modified fcn-8. In *2021 Fourth International Conference on Vocational Education and Electrical Engineering (ICVEE)*, pages 1–6.

Oyelade, I., Boyinbode, O., Adewale, O., and Ibam, E. O. (2024). Farmland intrusion detection using internet of things and computer vision techniques. *Inter. Journal of Information Technology and Computer Science*.

Parola, M., Cimino, M. G., Cantini, I., Mantia, G. L., Campisi, G., and Di Fede, O. (2025). Oral cancer recognition on photographic images via deep learning semantic

segmentation. In *2025 IEEE Symposium on Computational Intelligence in Health and Medicine Companion (CIHM Companion)*, pages 1–5.

Pezzementi, Z., Tabor, T., Hu, P., Chang, J. K., Ramanan, D., Wellington, C., Wisely Babu, B. P., and Herman, H. (2017). Comparing apples and oranges: Off-road pedestrian detection on the nrec agricultural person-detection dataset. *Journal of Field Robotics*.

Quionero-Candela, J., Sugiyama, M., Schwaighofer, A., and Lawrence, N. D. (2009). *Dataset Shift in Machine Learning*. MIT Press, Cambridge, MA.

Radojcic, V. and Cvetković, A. S. (2023). Development of new sensors and technologies for precision agriculture. *STED JOURNAL*, 5:44–49.

Rao, M. V. K., Challawar, A., Adithi, B., Adithi, B., and Bussa, P. (2023). Enhancing farm security: Animal intrusion detection using yolo. In *2023 2nd International Conference on Automation, Computing and Renewable Systems (ICACRS)*, pages 1476–1481.

Redmon, J., Divvala, S., Girshick, R., and Farhadi, A. (2016). You only look once: Unified, real-time object detection. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 779–788.

Redmon, J. and Farhadi, A. (2016). Yolo9000: Better, faster, stronger.

Redmon, J. and Farhadi, A. (2018). Yolov3: An incremental improvement. *ArXiv*, abs/1804.02767.

Ren, S., He, K., Girshick, R., and Sun, J. (2017). Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks . *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 39(06):1137–1149.

Ronneberger, O., Fischer, P., and Brox, T. (2015). U-net: Convolutional networks for biomedical image segmentation. arXiv:1505.04597. https://arxiv.org/abs/1505.04597.

S, D., Bruntha, P. M., G, R., R, J. S. A., and A, K. (2025). Determining the efficacy of senet integrated yolo models for animal detection. In *2025 5th International Conference on Expert Clouds and Applications (ICOECA)*, pages 1119–1123.

Santos, W. M. d., Martins, L. D. C. d. S., Bezerra, A. C., Souza, L. S. B. d., Jardim, A. M. d. R. F., Silva, M. V. d., Souza, C. A. A. d., and Silva, T. G. F. d. (2024). Use of unmanned aerial vehicles for monitoring pastures and forages in agricultural sciences: A systematic review. *Drones*, 8(10).

Saranya, S. M., V, N., E, M., P, L. K., Komarasamy, D., and S, M. (2024). Image segmentation using deeplab v3+ for diabetic retinopathy. In *2024 1st International Conference on Innovative Sustainable Technologies for Energy, Mechatronics, and Smart Systems (ISTEMS)*, pages 1–6.

Schenkel, F. and Middelmann, W. (2020). Domain adaptation for semantic segmentation of aerial imagery using cycle-consistent adversarial networks. In *IGARSS 2020 - 2020 IEEE International Geoscience and Remote Sensing Symposium*, pages 1448–1451.

Segu, G. S. P. K., Sivannarayana, A. D. S. N., and Ramesh, S. (2024). Real time road lane detection and vehicle detection on yolov8 with interactive deployment. In *2024 IEEE 16th International Conference on Computational Intelligence and Communication Networks (CICN)*, pages 267–272.

Shang, P., Hu, J., and He, B. (2023). Georeferencing algorithm for uav images based on semantic segmentation. In *International Conference on Artificial Intelligence and Big Data (ICAIBD)*.

Shetty, A. D. and Ashwath, S. (2023). Animal detection and classification in image & video frames using yolov5 and yolov8. In *Inter. Conf. on Electronics, Communication and Aerospace Technology (ICECA)*.

Silva, L., Ferreira, J. Q., Rezeck, P., Silva, M. M., and Gomes, T. L. (2024). Photo-realistic and labeled synthetic uav flight data generation using ros and gazebo. In *Latin American Robotics Symposium (LARS)*.

Singh, A., Rajan, S., Amini, M., Green, J. R., and Dick, K. (2023). Critical electrical infrastructure segmentation in arctic conditions. In *IEEE Sensors Applications Symposium (SAS)*.

Sohn, K., Berthelot, D., Li, C.-L., Zhang, Z., Carlini, N., Cubuk, E. D., Kurakin, A., Zhang, H., and Raffel, C. (2020). Fixmatch: Simplifying semi-supervised learning with consistency and confidence.

Speth, S., Gonçalves, A., Rigault, B., Suzuki, S., Bouazizi, M., Matsuo, Y., and Prendinger, H. (2022). Deep learning with rgb and thermal images onboard a drone for monitoring operations. *Journal of Field Robotics*.

Sultanov, R., Lavrenov, R., Sulaiman, S., Bai, Y., Svinin, M., and Magid, E. (2023). Object detection methods for a robot soccer. In *2023 7th International Conference on Information, Control, and Communication Technologies (ICCT)*, pages 1–5.

Sun, B. and Saenko, K. (2016). Deep coral: Correlation alignment for deep domain adaptation. In Hua, G. and Jégou, H., editors, *Computer Vision – ECCV 2016 Workshops*, pages 443–450, Cham. Springer International Publishing.

Sun, H., Mai, H., Mao, Y., Li, Q., Guo, M., Liu, Y., Feng, Z., Feng, H., Guo, W., Yang, G., Deng, X., and Song, X. (2025). Multi-variety monitoring of potato late blight severity using UAV data with improved SMOTE-CS for small sample modeling and deep feature learning. *European Journal of Agronomy*.

Sun, Z., Li, J., and Mu, Y. (2024). Exploring orthogonality in open world object detection. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 17302–17312.

Tabor, T., Pezzementi, Z., Vallespi, C., and Wellington, C. (2015). People in the weeds: Pedestrian detection goes off-road. In *Inter. Symposium on Safety, Security, and Rescue Robotics (SSRR)*.

Taigman, Y., Yang, M., Ranzato, M., and Wolf, L. (2014). Deepface: Closing the gap to human-level performance in face verification. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1701–1708.

Tang, X., Wang, X., Yan, N., Fu, S., Xiong, W., and Liao, Q. (2022). A new ore image segmentation method based on swin-unet. In *China Automation Congress (CAC)*.

Tarvainen, A. and Valpola, H. (2018). Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results.

Touvron, H., Cord, M., Douze, M., Massa, F., Sablayrolles, A., and Jégou, H. (2021). Training data-efficient image transformers & distillation through attention. In *International Conference on Machine Learning (ICML)*, pages 10347–10357. PMLR.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017). Attention is all you need. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 5998–6008.

Wang, J., Sun, K., Cheng, T., Jiang, B., Deng, C., Zhao, Y., Liu, D., Mu, Y., Tan, M., Wang, X., Liu, W., and Xiao, B. (2020). Deep high-resolution representation learning for visual recognition. In *arXiv Pre-Print*.

Wang, K., Fu, X., Huang, Y., Cao, C., Shi, G., and Zha, Z.-J. (2023). Generalized uav object detection via frequency domain disentanglement. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1064–1073.

Whitney, C. D. and Norman, J. (2024). Real risks of fake data: Synthetic data, diversity-washing and consent circumvention. FAccT '24, page 1733–1744, New York, NY, USA. Association for Computing Machinery.

Xue, X. and Du, W. (2024). Retinal fluid segmentation from oct b-scan using swin-unet. In *International Conference on Computer Graphics and Image Processing (CGIP)*.

Yang, L., Jiang, W., Ji, H., Zhao, Z., Zhu, X., and Hou, A. (2019). Automatic brain tumor segmentation using cascaded fcn with densecrf and k-means. In *2019 IEEE/CIC International Conference on Communications in China (ICCC)*, pages 545–549.

Yang, Y., Yang, J., Xu, Y., Zhang, J., Lan, L., and Tao, D. (2022). Apt-36k: A large-scale benchmark for animal pose estimation and tracking. In *Advances in Neural Information Processing Systems*.

Yu, F., Chen, H., Wang, X., Xian, W., Chen, Y., Liu, F., Madhavan, V., and Darrell, T. (2020). Bdd100k: A diverse driving dataset for heterogeneous multitask learning. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2633–2642.

Yuan, Y., Chen, X., Chen, X., and Wang, J. (2021). Segmentation transformer: Object-contextual representations for semantic segmentation. In *arXiv Pre-Print*.

Zhao, S., Yue, X., Zhang, S., Li, B., Zhao, H., Wu, B., Krishna, R., Gonzalez, J. E., Sangiovanni-Vincentelli, A. L., Seshia, S. A., and Keutzer, K. (2022). A review of single-source deep unsupervised visual domain adaptation. *IEEE Transactions on Neural Networks and Learning Systems*, 33(2):473–493.

Zhao, X., Schulter, S., Sharma, G., Tsai, Y., Chandraker, M., and Wu, Y. (2020). Object detection with a unified label space from multiple datasets. *CoRR*, abs/2008.06614.

Zhou, B., Zhao, H., Puig, X., Xiao, T., Fidler, S., Barriuso, A., and Torralba, A. (2018). Semantic understanding of scenes through the ade20k dataset.

Zhou, X., Koltun, V., and Krähenbühl, P. (2022). Simple multi-dataset detection. In *CVPR*.

Çallı, E., Sogancioglu, E., van Ginneken, B., van Leeuwen, K. G., and Murphy, K. (2021). Deep learning for chest x-ray analysis: A survey. *Medical Image Analysis*, 72:102125.

Öztürk, A. E. and Erçelebi, E. (2021). Real uav-bird image classification using cnn with a synthetic dataset. *Applied Sciences*, 11.