

الْجُمْهُورِيَّةُ الْجَزَائِرِيَّةُ الدِّيمَقْرَاطِيَّةُ الشَّعْبِيَّةُ
PEOPLE'S DEMOCRATIC REPUBLIC OF ALGERIA

وزارة التعليم العالي والبحث العلمي
MINISTRY OF HIGHER EDUCATION AND SCIENTIFIC RESEARCH
جامعة عمار تليجي بالأغواط
UNIVERSITY OF Ammar Telidji LAGHOUAT



كلية العلوم
FACULTY OF SCIENCES
قسم الاعلام الالي
DEPARTMENT OF COMPUTER SCIENCE

Field: Mathematics and Computer Science

Option: Computer Science

Specialization: Information and Decision Systems

**Thesis Presented in Partial Fulfilment of The Requirements
for The Degree of Master of Science in Computer Science**

**Towards A Powered AI GIS Platform for Agricultural Implantation
Identification**

Author(s)

DJOUDI Imane

MAATI Boutheina

Jury members

Ms.	Narjes HAMINI	MC(A)	University of Laghouat	President
Mr.	Mohamed Al Habib MAICHA	MC (A)	University of Laghouat	Examiner
Mr.	Moustapha BOUAKKAZ	Professor	University of Laghouat	Examiner
Mr.	Younes GUELLOUMA	MC(A)	University of Laghouat)	Advisor

Academic Year: 2023 – 2024

ACKNOWLEDGMENTS

We are profoundly grateful to the almighty Allah for His guidance and the strength He gave us throughout this journey. Without His grace and support, none of this would have been possible.

First and foremost, we extend our deepest appreciation to our parents and families. Their unwavering love, patience, and belief in our abilities have been the cornerstone of our success. Their encouragement has been a constant source of inspiration, and we are eternally indebted to you for your steadfast support.

We would also like to express our sincere thanks to our Advisor, Mr. Guellouma, and our Co-advisor, Ms. Chettih. Your insightful mentorship, constructive feedback, and assistance have been invaluable to our research and the completion of this thesis. We extend our gratitude to all members of the jury for examining our work.

Our genuine thanks go to the teachers of computer science department who have guided us throughout our academic journey. Your contributions have been pivotal to our achievements. The knowledge we have acquired from your teachings have been instrumental.

Finally, we extend our gratitude to our colleagues. Your camaraderie and support have made this journey enjoyable and enriching. Thank you.

DEDICATIONS

I would like to dedicate this work to my parents, who have always supported and encouraged me to pursue my dreams. Your love and guidance have been invaluable throughout my academic journey. Thanks to them, I am here today, I wish for nothing but to make them proud.

To my dear sister, Ismahane, and my brothers, Ihab, Ayoub, and Iyad—thank you for always being there with advice and inspiration. You are my pillars of strength and motivation.

To all the colleges and to all my teachers.

I will always be grateful.

Boutheina, June 2024

DEDICATIONS

To my dear Parents and Siblings,

Your perpetual inspiration and consistent motivation have been the foundation of my achievements. I am deeply grateful for your resolute belief in me, which has been a constant source of drive and strength. This thesis is dedicated to you with immense love and heartfelt gratitude.

Here's to the milestones we've achieved together.

To all my Mentors and Colleagues,

To my Nephew Elias

Imane, June 2024.

ABSTRACT

With the expansion of agricultural areas and the diversity of crops, manually identifying and precisely locating these areas has become increasingly challenging and time-consuming. In this work, we propose a technique that combines the power of Deep Learning, specifically Convolutional Neural Networks (CNNs or ConvNet) for identifying agricultural areas using aerial imagery, and Geographic Information System (GIS) for spatial mapping.

In this study, we use a dataset of scene images collected from different remote imaging sensors, and apply appropriate preprocessing techniques to improve the data quality. We trained four models on the preprocessed dataset, including our convolutional neural network (CNN) model that was built from scratch, and leveraging the power of transfer learning, we adapted the pretrained VGG16, DenseNet121, and ResNet50 models on the same dataset to establish the most appropriate model for our task.

The pre-trained VGG16 model demonstrated the highest performance with an accuracy of 96%, followed by DenseNet121 at 94%. Our CNN model achieved an accuracy of 89%, whereas ResNet50 recorded the lowest accuracy of 62%. Then using GIS techniques, we developed a desktop application to efficiently classify and visualize geotagged images in the Assafia region map, with a particular focus on farmlands. This application leverages the predictive results of our customized VGG16 model, chosen for its exceptional performance in our specific task. Our work demonstrates the effectiveness of transfer learning techniques, particularly the VGG16 model, for accurate and timely land type classification. Additionally, it underscores the utility of GIS techniques in visualizing spatial data effectively.

Keywords: Deep Learning, Convolutional Neural Networks (CNNs), Transfer Learning, Classification, VGG16, DenseNet121, ResNet50, Smart Farming, Geographic Information System (GIS), Aerial Imagery.

ملخص

مع توسع المناطق الزراعية وتنوع المحاصيل، أصبح التعرف على هذه المناطق يدوياً وتحديد مواقعها بدقة أمراً صعباً ويستغرق وقتاً طويلاً. في هذا العمل، نقترح تقنية تجمع بين قوة التعلم العميق، تحديداً الشبكات العصبية التلافيفية (CNNs) للتعرف على المناطق الزراعية باستخدام الصور الجوية، ونظام المعلومات الجغرافية (GIS) لرسم الخرائط المكانية. في هذه الدراسة، نستخدم مجموعة بيانات من صور المشاهد المجمعّة من مستشعرات التصوير عن بعد المختلفة، ونطبق تقنيات المعالجة المسبقة المناسبة لتحسين جودة البيانات. قمنا بتدريب أربعة نماذج على مجموعة البيانات التي تم معالجتها مسبقاً، بما في ذلك نموذج الشبكة العصبية التلافيفية (CNN) الذي تم بناؤه من الصفر، وباستغلال قوة نقل التعلم، قمنا بتكييف النماذج المدربة مسبقاً VGG16 وDenseNet121 وResNet50 على نفس مجموعة البيانات لتحديد النموذج الأكثر ملاءمة لمهمتنا. أظهر النموذج المدرب مسبقاً VGG16 الأداء الأعلى بدقة بلغت 96%، يليه DenseNet 121 بدقة 94%.

حقق نموذج CNN الخاص بنا دقة بلغت 89%، في حين سجل ResNet50 أقل دقة بنسبة 62%. ثم باستخدام تقنيات GIS، قمنا بتطوير تطبيق مكتبي لتصنيف الصور الجغرافية وعرضها بكفاءة في خريطة منطقة العسافية، مع التركيز بشكل خاص على الأراضي الزراعية. يستفيد هذا التطبيق من النتائج التنبؤية لنموذج VGG16 المخصص الخاص بنا، الذي تم اختياره لأدائه الاستثنائي في مهمتنا المحددة. يوضح عملنا فعالية تقنيات نقل التعلم، خصوصاً نموذج VGG16، في تصنيف أنواع الأراضي بدقة وفي الوقت المناسب. بالإضافة إلى ذلك، يؤكد على فائدة تقنيات GIS في تصور البيانات المكانية بفعالية.

الكلمات المفتاحية: التعلم العميق، الشبكات العصبية التلافيفية (CNNs)، التعلم بالنقل، التصنيف، VGG16، DenseNet121، ResNet50، الزراعة الذكية، نظام المعلومات الجغرافية (GIS)، الصور الجوية.

RÉSUMÉ

Avec l'expansion des zones agricoles et la diversité des cultures, l'identification manuelle et la localisation précise de ces zones sont devenues de plus en plus difficiles et chronophages. Dans ce travail, nous proposons une technique qui combine la puissance de l'apprentissage profond, plus précisément les Réseaux Neuronaux Convolutifs (RNC), pour identifier les zones agricoles à l'aide d'images aériennes, et les Systèmes d'Information Géographique (SIG) pour la cartographie spatiale.

Dans notre étude, nous utilisons un ensemble de données d'images de scènes collectées à partir de différents capteurs d'imagerie à distance, et appliquons des techniques de prétraitement appropriées pour améliorer la qualité des données. Nous avons entraîné quatre modèles sur l'ensemble de données prétraité, dont notre modèle de Réseau de Neurones Convolutifs (CNN) construit à partir de zéro. En exploitant la puissance du transfert d'apprentissage, nous avons adapté les modèles pré-entraînés VGG16, DenseNet121 et ResNet50 sur le même ensemble de données pour établir le modèle le plus approprié pour notre tâche.

Le modèle VGG16 pré-entraîné a démontré les meilleures performances avec une précision de 96%, suivi de DenseNet121 à 94%. Notre modèle CNN a atteint une précision de 89%, tandis que ResNet50 a enregistré la plus faible précision de 62%. Ensuite, en utilisant des techniques SIG, nous avons développé une application de bureau pour classer efficacement et visualiser des images géolocalisées sur la carte de la région d'Assafia, en mettant particulièrement l'accent sur les terres agricoles. Cette application exploite les résultats prédictifs de notre modèle VGG16 personnalisé, choisi pour ses performances exceptionnelles dans notre tâche spécifique.

Notre travail démontre l'efficacité des techniques de transfert d'apprentissage, en particulier le modèle VGG16, pour la classification précise et opportune des types de terrains. De plus, il souligne l'utilité des techniques SIG dans la visualisation efficace des données spatiales.

Mots Clés : Apprentissage Profond (Deep Learning), Réseaux de Neurones Convolutifs (RNC), Transfert d'Apprentissage, Classification, VGG16, DenseNet121, ResNet50, Agriculture Intelligente, Système d'Information Géographique (SIG), Imagerie Aérienne.

CONTENTS

ACKNOWLEDGMENTS	1
ABSTRACT.....	IV
LIST OF FIGURES.....	VIII
LIST OF TABLES.....	IX
INTRODUCTION	1
1. Chapter I – Generalities.....	3
1.1. Introduction.....	4
1.2. Agricultural Revolutions.....	4
1.3. Agriculture 4.0: Definition and Overview	6
1.4. The importance of Identifying Agricultural Areas	7
1.5. The Evolution of Agricultural Areas Identification Methods.....	8
1.6. Conclusion	10
2. Chapter II – GIS, Deep Learning and Literature Review	11
2.1. Introduction.....	12
2.2. Geographic Information System	12
2.3. Deep Learning	17
2.4. Literature Review	29
2.5. Conclusion	34
3. Chapter III – Methodology and Results	35
3.1. Introduction.....	36
3.2. Used tools.....	36
3.3. Model Development	41
3.4. Geospatial Data Processing and Analysis	54
3.5. Conclusion	63
GENERAL CONCLUSION	64
BIBLIOGRAPHY	65

LIST OF FIGURES

Figure 1 : Timelines of the Evolution Process from Agriculture 1.0 to Agriculture 4.0.	6
Figure 2 : Agriculture 4.0 Core Technologies	7
Figure 3 : Plant Sampling by Agricultural Expert	9
Figure 4: Aerial Image of Agricultural Land	9
Figure 5 : Remote Sensing Technologies – Unmanned Aerial Vehicles (UAVs)	10
Figure 6: Types of information stored in a GIS.	13
Figure 7: Vector Data Types	14
Figure 8: Example of a Raster Data File	15
Figure 9: Example of a GIS Map	16
Figure 10: Data layers combined through GIS – Integrated Digital Maps of Earth’s Surface.	16
Figure 11: Neuron in Artificial Neural Network	18
Figure 12: Multi-Layer Neural Network	19
Figure 13: Basic CNN Architecture	20
Figure 14: Convolution Operation.....	21
Figure 15: Illustration of Max Pooling	22
Figure 16: The usage of Fully Connected Layers in the classification section of CNNs.	23
Figure 17: Transfer Learning Diagram	25
Figure 18: Architecture of VGG-16 and VGG-19	26
Figure 19: Skip Connection.....	27
Figure 20 : Python Logo	36
Figure 21 : Anaconda Logo	37
Figure 22 : PostgreSQL Logo	38
Figure 23 : PostGis Logo	39
Figure 24: QGIS Logo.....	39
Figure 25: Qt Designer Logo	40
Figure 26: GeoServer Logo	40
Figure 27: Google Earth Pro Logo	41
Figure 28: Some samples with their classes in AID dataset.....	42
Figure 29: Some samples with their classes in the RSSCN7 dataset.....	43
Figure 30: The CNN Architecture.	46
Figure 31: Transfer Learning Methodology.....	47
Figure 32: VGG16 Training and Validation Accuracy.	51
Figure 33: VGG16 Training and Validation Loss.....	51
Figure 34: The Assafia Region Localization on The Map	54

Figure 35: Assafia Border layer	55
Figure 36: Geotagged Images Dataset.....	56
Figure 37: The Custom Plugin interface.....	57
Figure 38: Preview of the attributes table.....	58
Figure 39: Visualizing Geotagged Images of Assafia on the Map.....	58
Figure 41: Aggregate Layer.....	60
Figure 42: The aggregate layer view in OpenLayers.....	61
Figure 43 : The aggregate layer view in Google Earth Pro.....	61

LIST OF TABLES

Table 1: Architectures of different variants of ResNets	28
Table 2: Training and Testing accuracy for Individual datasets	49
Table 3: Number of trainings, testing and validation images in each category	50
Table 4: Training and testing accuracy on the combined dataset	50

INTRODUCTION

The agricultural sector serves as the linchpin of global economies, crucial for ensuring food security and promoting economic development. In the context of a rapidly changing agricultural landscape, driven by technological advancements and evolving practices, the accurate identification of agricultural implantations has become increasingly challenging. In fact, with the increase in the number of agricultural implantations and the diversity of crop types, manual identification of agricultural areas has become difficult, time-consuming and inefficient.

The process of identifying agricultural implantations involves a multifaceted approach encompassing several key challenges. Firstly, accurately delineating the boundaries of agricultural plots or perimeters is a major obstacle. This difficulty is often due to the irregular shapes and sizes of agricultural plots, compounded by variations in land use practices and boundary demarcations. Precise mapping of implantations is crucial for efficient land allocation and resource optimization, which in turn enhances overall agricultural productivity and sustainability. Secondly, determining the nature of agricultural implantations involves categorizing them based on factors like crop types, which necessitates comprehensive data collection and sophisticated classification frameworks, in order to discern the diverse characteristics of these entities which is essential for tailored management strategies. Thirdly, the dynamic nature of agricultural establishments, which may change in size, shape, or crop type, poses additional challenges for consistent identification. Monitoring these changes requires the analysis of temporal datasets to facilitate instant and accurate identification of issues, thereby enabling targeted interventions.

To address these challenges, it is necessary to explore automated solutions. Recent advancements in Geographic Information Systems (GIS) and deep learning techniques have significantly improved the identification of agricultural areas. By integrating techniques such as aerial imagery, satellite imagery, and remote sensing data, GIS provides a robust platform for the organization, analysis, and visualization of spatial data. Concurrently, Deep Learning Algorithms, particularly Convolutional Neural Networks (CNNs), excel in image classification tasks through their ability to automatically learn and recognize features. Hence, through the synergy between GIS for geospatial analysis and Deep Learning for image recognition, the identification of agricultural areas achieves unprecedented levels of accuracy and efficiency.

This thesis presents the development of an aerial image identifier for agricultural implantations, leveraging deep learning techniques and integrating the results into a Geographic Information System (GIS) framework. By training Convolutional Neural Networks (CNNs) and utilizing transfer learning models, we classify aerial images to identify agricultural areas. The classification is performed within QGIS using the trained model. A custom desktop application was developed in QGIS, incorporating a specialized plugin to

manage and analyze spatial data. This application connects to PostGIS and PostgreSQL for data management and GeoServer for data publication, enhancing the visualization and interaction with spatial data. This approach aims to improve the accuracy and efficiency of identifying agricultural areas, contributing to more effective and sustainable agricultural practices.

Along with an introduction and a general conclusion, the structure of this master thesis unfolds across three pivotal chapters:

Chapter I serves as our foundational cornerstone, delving into the historical contexts of agricultural revolutions and the emergence of Agriculture 4.0. Within this chapter, we scrutinize the significance of identifying agricultural areas and trace the evolution of methods employed for this purpose. Through an in-depth analysis, we seek to establish a contextual framework that underpins subsequent research endeavors.

In Chapter II we explore the synergies between Geographic Information Systems (GIS) and Deep Learning. This chapter begins with an introduction to GIS, highlighting its role in agricultural landscape analysis. We then delve into Deep Learning, examining its applications and impact on spatial data analysis. Through a concise literature review, we synthesize existing research findings, identifying key insights and gaps that inform our methodological approaches. This chapter aims to provide a comprehensive overview while setting the stage for our contributions in the subsequent chapter.

Chapter III emerges as the culmination of our theoretical explorations, translating conceptual frameworks into tangible research methodologies and empirical findings. Within this chapter, we elucidate the tools employed, detail the development of our models, and outline the geospatial data processing and analysis techniques utilized. Through a meticulous exposition of methodology and results, we endeavor to shed light on the intricacies of agricultural landscape identification and analysis.

These chapters collectively form the backbone of our thesis, each contributing a unique perspective to the overarching narrative of agricultural landscape analysis in the era of technological innovation and data-driven insights.

1. Chapter I – Generalities

1.1.	Introduction	4
1.2.	Agricultural Revolutions	4
1.2.1.	Agriculture 1.0.....	4
1.2.2.	Agriculture 2.0.....	5
1.2.3.	Agriculture 3.0.....	5
1.2.4.	Agriculture 4.0.....	5
1.3.	Agriculture 4.0: Definition and Overview	6
1.4.	The importance of Identifying Agricultural Areas	7
1.4.1.	Optimizing Resource Allocation	7
1.4.2.	Monitoring Agricultural Practices	8
1.4.3.	Ensuring Food Security.....	8
1.4.4.	Promoting Environmental Sustainability	8
1.5.	The Evolution of Agricultural Areas Identification Methods	8
1.5.1.	Traditional Methods: Manual surveys	8
1.5.2.	Remote Sensing Technologies	9
1.5.3.	AI, Deep learning and GIS.....	9
1.6.	Conclusion	10

1.1. Introduction

The ever-growing global population necessitates sustainable solutions to food security, one that efficiently utilizes arable land while minimizing environmental impact. A critical component of these solutions is the accurate identification and delineation of agricultural areas.

The history of agriculture is punctuated by transformative revolutions, each bringing radical changes to farming practices and efficiency. From the labor-intensive methods of Agriculture 1.0 to the advanced technological systems of Agriculture 4.0, the evolution of agriculture has been characterized by continual innovation and adaptation [1].

This chapter examines the critical role of these agricultural revolutions, with a particular focus on Agriculture 4.0 and its integration of digital technologies. The research underscores the necessity of precise agricultural area identification and explores the evolution of methodologies employed for this purpose. Traditional techniques are contrasted with modern approaches such as Geographic Information Systems (GIS), illustrating how advancements in artificial intelligence (AI), deep learning, and GIS have revolutionized agricultural identification and mapping. These technological innovations are essential for promoting a more efficient, sustainable, and secure future in food production.

1.2. Agricultural Revolutions

Agriculture encompasses various ways in which crop plants and domestic animals sustain the global human population by providing food and other essential products. Over the years, agriculture has undergone numerous revolutions around the world, each bringing radical changes. Over half of the global population participates in agriculture to fulfil the world's total food requirements [2]. The evolution of agriculture can be categorized into four distinct stages [3]:

1.2.1. Agriculture 1.0

The agricultural technology revolution commenced with Agriculture 1.0, characterized traditional farming practices, dating back to ancient times when farmers relied predominantly on natural methods for farming activities, harnessing the capabilities of nature itself (ex., indigenous tools, animal power, etc.) [4]. This initial stage represents traditional farming practices that have been utilized by humans for centuries which persisted until the early 20th century. This form of peasant farming demanded a manual labor-intensive system, resulting in relatively low productivity levels [5]. Techniques such as the use of natural fertilizers, pesticides, and weed control products are employed to support crop growth and manage agricultural processes [6]. Agriculture 1.0 sustained the population's food needs, but required a vast number of small farms and one-third of the population actively engaged in primary agricultural production.

1.2.2. Agriculture 2.0

Agriculture 2.0, defined by the advent of the combustion engine unfolded in the late 20th century marking the transition into the 21st century, during which the transition from steam to oil and gas as the primary energy sources, coupled with innovations in the transportation industry, played a pivotal role in advancing the agri-food supply chain [7]. During this period, researchers and technicians focused on addressing agricultural challenges to achieve greater productivity with reduced labor input. New agronomic management practices and forms of information technology allowed to take advantage of relatively cheap inputs, thus significantly increasing yield potential and growing returns to scale at all levels (ex., the use of synthetic pesticides, fertilizers and more efficient specialized machinery such as tractors, combines, and automobiles to mechanize agricultural processes. The introduction of agricultural machinery for tasks such as seedbed preparation, sowing, irrigation, weeding, and harvesting, assembly-line-based mass production techniques and advanced data analysis and monitoring techniques, revolutionized agricultural efficiency while reducing reliance on manual labor and maximizing harvest yields across industries.

1.2.3. Agriculture 3.0

Agriculture 3.0 has arisen recently, characterized by the introduction of guidance systems and precision farming techniques, which align with the accessibility of military GPS signals for public use in the mid-1990s [8]. Early adopters built further on technology used in aerial spraying. During this period, advancements in embedded systems, software engineering, and communication technologies enhanced the automation capabilities of manufacturing equipment [7]. By using techniques such as yield monitoring, variable rate applications, and guidance farming systems [7], Agriculture 3.0 aims to leverage information technologies for precision agriculture [9].

1.2.4. Agriculture 4.0

The ongoing fourth industrial revolution, known as Industry 4.0, is characterized by the integration of emerging technologies such as the Internet of Things (IoT), robotics, big data, artificial intelligence (AI), and blockchain technology. This fusion has led to industrial production processes and supply chains becoming increasingly autonomous and intelligent [1]. The integration of Industry 4.0 principles into agriculture has given rise to the concept of Agriculture 4.0 (see § section 1. 3).

The following figure provides an overview of the historical progression of agricultural revolutions that led up to Agriculture 4.0 (Cf. Figure 1).

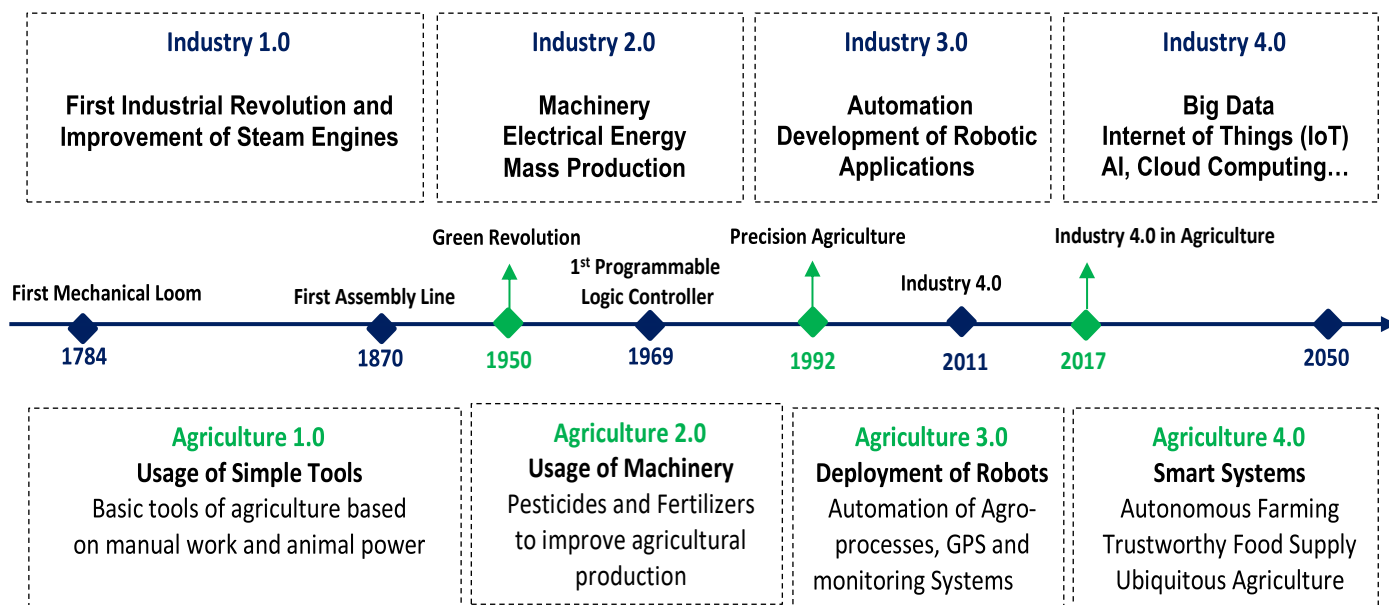


Figure 1 : Timelines of the Evolution Process from Agriculture 1.0 to Agriculture 4.0.

Source: own elaboration, according to [6] [9] [10].

1.3. Agriculture 4.0: Definition and Overview

Agriculture 4.0, also referred to as farming 4.0 or Smart Agriculture, represents the next generation of industrial agriculture. It entails the incorporation of cyber-physical characteristics and advancements of Industry 4.0 into agriculture to create sustainable and intelligent farming practices. It encompasses the integration of a suite of digital technologies interconnected by software, systems, and equipment capable of optimizing agricultural production across the entire value chain, from upstream processes to midstream operations and downstream activities [10]. This involves real-time collection, processing, and analysis of spatiotemporal data across all aspects of the agricultural industry, including food production, processing, distribution, and consumer experience [7].

Agriculture 4.0 is an integrative approach that leverages the power of digital technologies to transform traditional farming practices, as advances in sensor technology have allowed farmers to monitor specific parameters in real-time and high levels of automation, while robotics have supported a better automation of the processes[11]. This has significantly enhanced productivity, agri-food supply chain efficiency, food safety, and the sustainable use of natural resources. Additionally, computing power has become more accessible and affordable, which has also helped the creation of new decision support tools and data-driven intelligent decision-making for better agricultural management [11]. For instance, big data

supports a high-volume of real-time and historical data and AI-based methods transform these data into added value and actionable knowledge.

An illustration of the data flow between the identified technologies is presented in Figure 2.

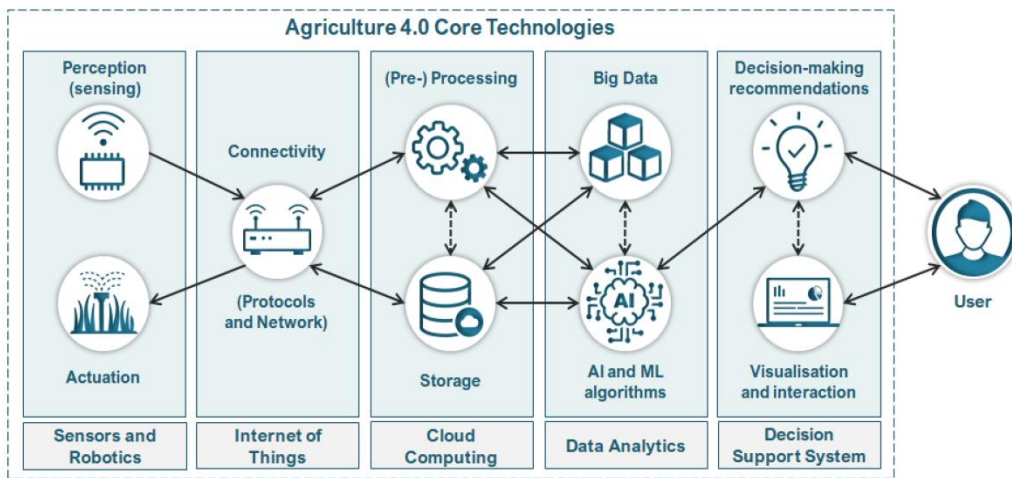


Figure 2 : Agriculture 4.0 Core Technologies [10].

Precise land use is crucial, and one key aspect of Agriculture 4.0 that involves the identification and mapping of agricultural areas using advanced technologies.

1.4. The importance of Identifying Agricultural Areas

In the ever-evolving landscape of agriculture, pinpointing the exact location and characteristics of agricultural areas and implantations is no longer a simple formality. It has become a cornerstone for optimizing practices, ensuring sustainability, and ultimately, achieving food security for a growing global population. the following points are e key reasons why identifying agricultural areas is so crucial:

1.4.1. Optimizing Resource Allocation

Identification of agricultural land and its precise location allows for targeted resource allocation. By being able to efficiently distribute fertilizers, pesticides, and irrigation water only to the areas where they are truly needed and manpower being strategically distributed. This not only reduces waste but also ensures that crops receive the necessary resources to reach their full potential, maximizing yields, leading to increased efficiency and cost savings.

1.4.2. Monitoring Agricultural Practices

Precise identification of agricultural areas acts as a powerful tool for tracking changes in land use and agricultural practices over time. This information empowers policymakers to assess the effectiveness of agricultural programs. For instance, by identifying areas with declining soil fertility or inefficient irrigation practices, corrective measures can be implemented to improve overall agricultural productivity.

1.4.3. Ensuring Food Security

Identification of agricultural land is essential for estimating food production potential. This information is crucial for policymakers to formulate effective policies aimed at ensuring food security for a growing population. By accurately assessing agricultural capabilities, proactive measures can be taken to address potential food shortages or imbalances in production.

1.4.4. Promoting Environmental Sustainability

By accurately mapping agricultural land, we can create buffer zones to protect sensitive ecosystems like forests or wetlands. These buffer zones act as a shield, mitigating the negative environmental impact of agricultural activities like fertilizer runoff or pesticide use.

This approach promotes a balance between agricultural production and environmental well-being, fostering sustainable development.

1.5. The Evolution of Agricultural Areas Identification Methods

The agricultural industry has come a long way since its inception, evolving from traditional farming methods to more technologically advanced practices that leverage artificial intelligence and remote sensing. Here are the explanations of some of the main methods:

1.5.1. Traditional Methods: Manual surveys

Prior to advancements in technology, pinpointing agricultural areas primarily depended on manual surveys. This traditional approach involves physically visiting and visually assessing land cover by agricultural experts. This includes visually inspecting fields or taking samples of plants for laboratory analysis as shown in the following picture (*Cf.* Figure 3). While offering detailed ground-truthing, it is a time-consuming, labor-intensive, and geographically limited method.



Figure 3 : Plant Sampling by Agricultural Expert [12].

1.5.2. Remote Sensing Technologies

The advent of remote sensing technologies, including satellite and aerial imagery (refer to Figure 4), has revolutionized agricultural mapping [13]. Images captured from airplanes or drones allowed for a broader view of land use patterns, enabling faster identification of agricultural areas. However, interpreting aerial photographs still required human expertise and lacked the ability to automatically differentiate between different crop types.



Figure 4: Aerial Image of Agricultural Land [14].

1.5.3. AI, Deep learning and GIS

Nowadays, we stand at the forefront of a new era grounded on Artificial Intelligence (AI), Deep Learning, and Geographic Information Systems (GIS). Deep Learning sophisticated algorithms for data analysis and pattern recognition, particularly Convolutional Neural Networks (CNNs), analyze vast amounts of satellite

and aerial imagery, an example of that is illustrated in Figure 5 where a drone likely equipped with multispectral sensors, capturing data over green parcels of a field.



Figure 5 : Remote Sensing Technologies – Unmanned Aerial Vehicles (UAVs) [15].

The data is analysed using Deep Learning and GIS techniques. Unlike traditional methods, they automatically identify agricultural areas with exceptional accuracy and efficiency, significantly reducing the time and expertise required and differentiating between crop types based on subtle spectral variations even in complex landscapes with diverse vegetation types. GIS serves as the central platform, managing and visualizing the data generated by Deep Learning. This data can be enriched with soil type, weather patterns, and elevation information within the GIS framework.

1.6. Conclusion

This chapter explored the critical importance of accurately identifying agricultural areas and traced the evolution of the methods used to achieve this task. Starting with traditional techniques and approaches, the discussion advanced to the modern incorporation of cutting-edge technologies such as artificial intelligence (AI), deep learning, and Geographic Information Systems (GIS).

The traditional methods, while foundational, often relied heavily on manual labor and human expertise, which limited their efficiency and scalability. With the advent of AI and deep learning, the ability to process vast amounts of data and recognize complex patterns has revolutionized the field. These technologies enable precise and automated identification of different crop types, soil conditions, and other critical agricultural parameters, which were previously challenging to determine, facilitating better decision-making and resource management.

The following chapter will investigate these advanced technologies, offering an in-depth examination of their functionalities and applications in agricultural identification and mapping.

2. Chapter II – GIS, Deep Learning and Literature Review

2.1.	Introduction	12
2.2.	Geographic Information System	12
2.2.1.	Definition	12
2.2.2.	Spatial Data	13
2.2.3.	Spatial Data Models	14
2.2.4.	What are GIS Maps.....	15
2.3.	Deep Learning	17
2.3.1.	Artificial Neural Networks.....	17
2.3.2.	Convolutional Neural Network	20
2.3.3.	Transfer Learning	24
2.4.	Literature Review	29
2.4.1.	CNN-based Approaches for Agricultural Image Classification	29
2.4.2.	Applications of GIS in Agricultural Management and Spatial Mapping.....	31
2.4.3.	Combining GIS and CNN in Plant Classification and Mapping	31
2.4.4.	Summary.....	33
2.5.	Conclusion	34

2.1. Introduction

This chapter aims to explore the foundational concepts of Geographic Information Systems (GIS) and deep learning, with a particular focus on Convolutional Neural Networks (CNNs). We also discuss the concept of transfer learning and its popular pretrained models.

We review existing literature on the effectiveness of CNNs in recognizing and categorizing various agricultural features with high accuracy, such as classifying aerial images of agricultural areas.

Additionally, we investigate research that employed GIS for agricultural mapping, highlighting the capabilities of GIS in spatial analysis and data visualization. Moreover, we discuss research that integrates both techniques, utilizing CNNs for image classification and GIS for storing and visualizing the results. This integrated approach leverages the strengths of both CNNs and GIS, providing a powerful toolset for precise agricultural mapping and decision-making.

By examining these studies, we aim to provide a comprehensive understanding of how CNNs and GIS can be applied to agricultural mapping, and how the combination of these technologies can enhance the accuracy and efficiency of identifying and analyzing agricultural areas.

2.2. Geographic Information System

Geographic Information Systems (GIS) is a dynamic and rapidly evolving field that holds a crucial role in numerous industries and applications. It facilitates the analysis, visualization, and interpretation of spatial data. In this section, we will explain the fundamental aspects of GIS, with a focus on its role in spatial data analysis.

2.2.1. Definition

Geographic Information Systems are powerful tools designed for capturing, storing, analyzing, and displaying spatial and geographic data. They integrate various types of information, such as maps, satellite imagery, and statistical data, to provide a comprehensive view of a particular location or area. Breaking down the acronym GIS further clarifies its essence: "G" signifies a focus on the spatial identity or locality of entities on, under, or above the earth's surface; "I" Implies the need to be informed in order to make decisions.

Data or raw facts are interpreted to create information that is useful for decision-making; and "S" emphasizes the need for staff, computer hardware and procedures, which can produce the information required for decision-making, that is data collection, processing, and presentation [16].

2.2.2. Spatial Data

The cornerstone of any GIS is its data. This data, known as geographic or spatial data, along with related tabular data, can be sourced either internally or from commercial providers. As a fundamental element of GIS, Spatial Data refers to any type of data that has a geographic or location component. This can include different types of data such as maps, satellite imagery, aerial imagery, GPS data, and geographic survey information. Spatial data can be represented in a variety of ways and used to create different types of spatial analysis and visualization [17].

The terms spatial data and geoinformation are often used interchangeably, but there is a significant difference. While spatial data provides information about the spatial relationships between objects or phenomena, geoinformation combines spatial data with other types of data to provide a more comprehensive understanding of the world around us [17]. The following figure illustrates various types of information stored in GIS, including tabular data, maps, aerial photos, satellite images, and multimedia.



Figure 6: Types of information stored in a GIS.

Source: own elaboration, according to [16].

2.2.3. Spatial Data Models

There are two primary spatial data models, each serving as a blueprint for representing and organizing geographic information. These models provide a structured framework that enables efficient analysis and interpretation of spatial relationships, with each model having its distinct approach to organizing and storing geographic information:

A. Vector Data

The vector data model is based on the assumption that the Earth's surface is composed of discrete objects such as trees, rivers, lagoons, etc. In this model, there are no fundamental units that divide the collected area, but, rather, the variability and characteristics of this area are collected by means of geometric entities. For each geometric entity the characteristics are constant. The form of these entities is explicitly codified, because it models the geographic space through a series of primitive geometrics containing the most outstanding elements of that space [18]. These primitives are of three types: points, lines and polygons (*Cf.* Figure 7).

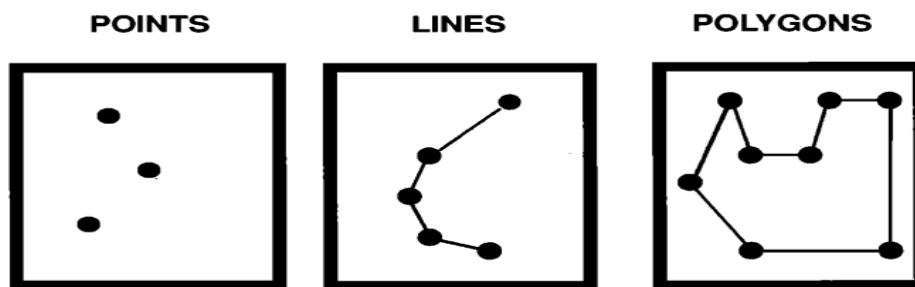


Figure 7: Vector Data Types [19]

– Points

Points represent individual locations and are defined by their X and Y coordinates in the GIS coordinate system. Points are commonly used to represent features such as City locations, monitoring stations, wells, and other discrete objects.

– Lines

Lines are sequences of connected points that represent linear features such as roads, rivers, and boundaries. They can be simple straight lines or more complex curves.

– Polygons

Polygons are closed shapes formed by connecting a series of points. They are used to represent areas such as lakes, parcels of land, and administrative boundaries, etc.

B. Raster Data

Raster is a method for storage, processing and display of spatial data. Each area is divided into rows and columns, which form a regular grid structure Figure 8. Each cell within this matrix contains location coordinates as well as an attribute value. The spatial location of each cell is implicitly contained within the ordering of the matrix, unlike a vector structure which stores topology explicitly. With the raster data model, spatial data is not continuous but divided into discrete units. This makes raster data particularly suitable for certain types of spatial operation, for example overlays or area calculations [20] [21].

Raster structures may lead to increased storage in certain situations, since they store each cell in the matrix regardless of whether it is a feature or simply 'empty' space.

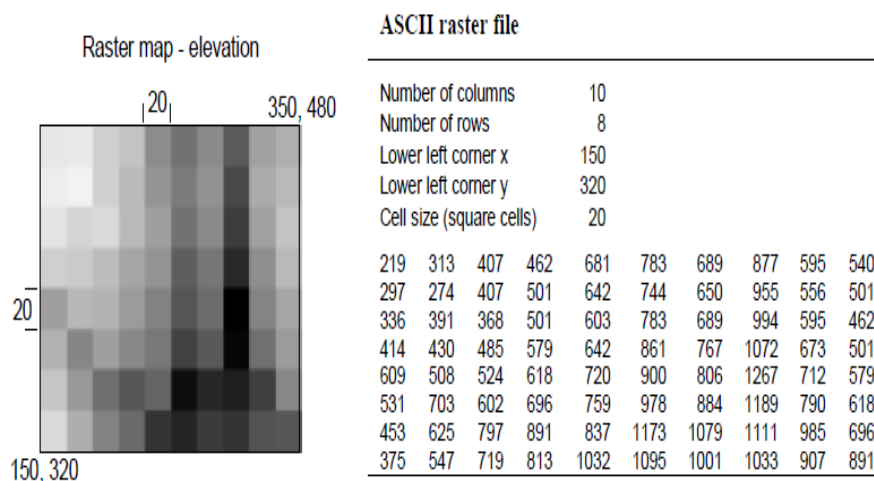


Figure 8: Example of a Raster Data File [22].

2.2.4. What are GIS Maps

Geographic Information System maps, as shown in Figure 9, are digital maps that use geographic data and spatial analysis that have been processed and analyzed using GIS software, to display and analyze information about the Earth's surface. Unlike traditional, tabular maps, GIS maps are dynamic and interactive. There are various types of GIS maps, including: category maps, heat maps, cluster maps, quantity maps [17]. One of the fundamental elements of any GIS is the map layers.

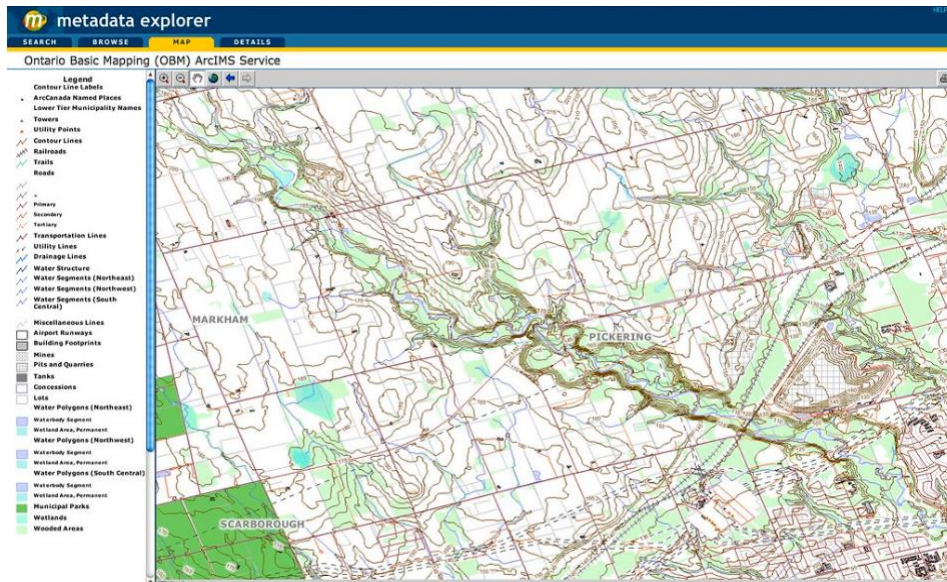


Figure 9: Example of a GIS Map [17].

Each layer in a map represents a different aspect of the geographic data being analyzed, and data layers can be turned on or off to display only the relevant information for a particular analysis or map [17]. An example of data layers is shown in Figure 10.

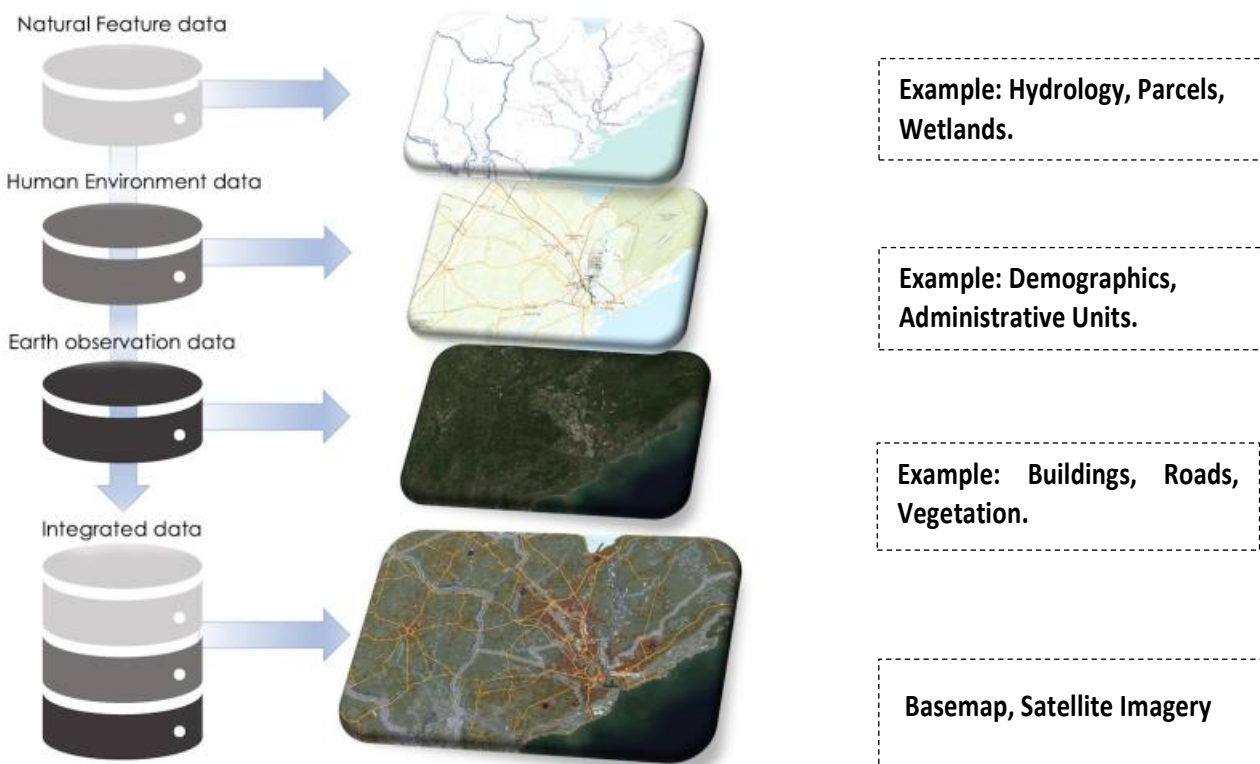


Figure 10: Data layers combined through GIS – Integrated Digital Maps of Earth’s Surface.

Source: Adapted from [23].

2.3. Deep Learning

Deep learning is the subfield of artificial intelligence that focuses on creating large neural network models that are capable of making accurate data-driven decisions. It is particularly suited to contexts where the data is complex and where there are large datasets available [24].

Based on artificial neural networks (ANNs), deep learning has a large number of layers and parameters. It uses a cascade of multiple layers of nonlinear processing units for feature extraction and transformation. The lower layers close to the data input learn simple features, while higher layers learn more complex features derived from lower layer features.

The architecture forms a hierarchical and powerful feature representation. It means that deep learning is suited for analyzing and extracting useful knowledge from both large huge amounts of data collected from different sources [25].

Deep learning algorithms learn to recognize patterns in data by training on large datasets and adjusting the network parameters until the output predictions become more accurate. It has achieved significant success in various fields, including image recognition, natural language processing, speech recognition, and recommendation systems. In recent years, deep learning has been widely used in the agricultural field in order to solve complex agricultural challenges. This technology enables the automation of tasks like the identification and classification of agricultural implantations, yield prediction, etc.

2.3.1. Artificial Neural Networks

Artificial neural networks are the functional unit of Deep Learning and are known to mimic the behavior of the human brain to solve complex data-driven problems. The input data is processed through different layers of artificial neurons stacked together to produce the desired output [26]. Artificial neural networks have been used in a varied set of domains [27].

A. Key Components of the Neural Network Architecture

Neural networks, as mathematical constructs, are composed of essential components called neurons, working harmoniously to process information, recognize patterns, and facilitate data-driven predictions.

The simplest neuron that includes the main features of a biological neural network parallelism and high connectivity was proposed by McCulloch and Pitts (1943), and still is the most used model in different neural network architectures [28].

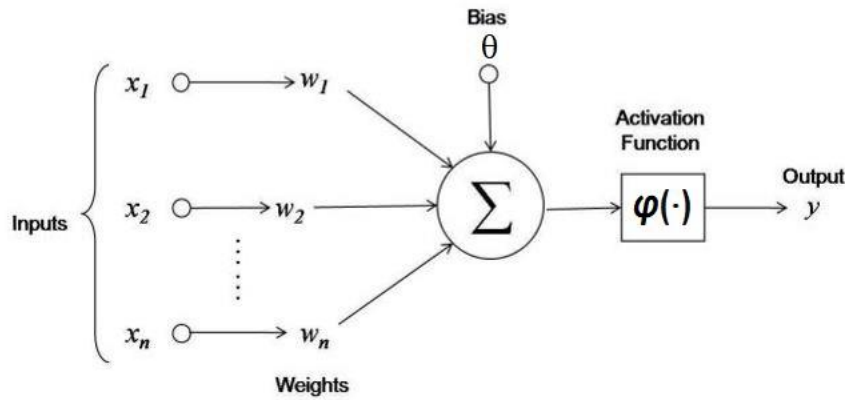


Figure 11: Neuron in Artificial Neural Network [27].

A neuron consists of several components (*Cf.* Figure 11), including inputs, weights, bias, summing function, activation function, and outputs.

Weights are assigned once an input layer has been defined. These weights, which are assigned at random at start, assist to define the importance of any given variable, with bigger ones contributing more substantially to the output than smaller ones [29].

As shown in the formula below, all inputs are multiplied by their respective weights, then summed and bias added to it:

$$Z = \sum_{i=1}^n w_i \cdot x_i + \theta$$

Where X_i represents the i^{th} input to the neuron, W_i represents the weight associated with the i^{th} input, θ is the bias, n is the number of inputs to the neuron, and Z is the weighted sum of inputs plus bias.

Next, this sum is passed through an activation function, which introduces non-linearity to the model. As shown in the formula below:

$Y = \varphi(Z)$ Where $\varphi(\cdot)$ the activation function, and y is the output of the neuron after activation.

B. Layers

When multiple neurons are stacked together in a row, they constitute a layer, and multiple layers piled next to each other are called a multi-layer neural network (Figure 12). The explanations of its main layers are provided below:

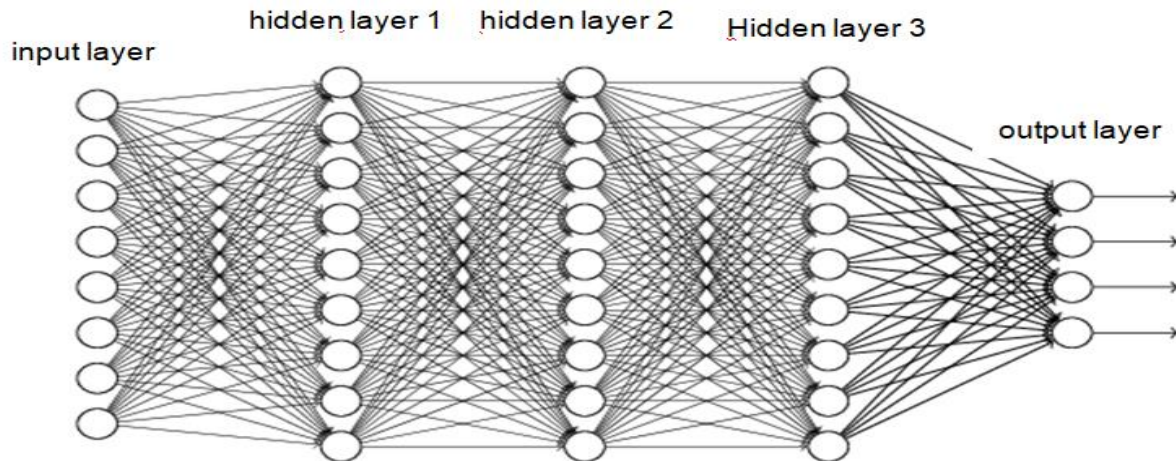


Figure 12: Multi-Layer Neural Network [30].

– Input Layer

The data that we feed to the model is loaded into the input layer from external sources like a CSV file or a web service. It is the only visible layer in the complete Neural Network architecture that passes the complete information from the outside world without any computation [27].

– Hidden Layers

The hidden layers are what makes deep learning what it is today. They are intermediate layers that do all the computations and extract the features from the data. There can be multiple interconnected hidden layers that account for searching different hidden features in the data. For example, in image processing, the first hidden layers are responsible for higher-level features like edges, shapes, or boundaries. On the other hand, the later hidden layers perform more complicated tasks like identifying complete objects (a car, a building, a person) [27].

– Output Layer

The output layer takes input from preceding hidden layers and comes to a final prediction based on the model’s learnings. It is the most important layer where we get the final result.

In the case of classification/regression models, the output layer generally has a single node. However, it is completely problem-specific and dependent on the way the model was built [27].

In the following, we consider convolutional neural networks. In fact, CNNs are very suitable for image classification problems, we choose therefore to explore their potential in identification and classification of agricultural implantations.

2.3.2. Convolutional Neural Network

Convolutional Networks also known as Convolutional Neural Networks or CNNs, invented by Yann LeCun et al in 1989 [31], are a specialized kind of neural network for processing data that has a known, grid-like topology. Examples include time-series data, which can be thought of as a 1D grid taking samples at regular time intervals, and image data, which can be thought of as a 2D grid of pixels. Convolutional networks have been tremendously successful in practical applications. The name “convolutional neural network” indicates that the network employs a mathematical operation called convolution.

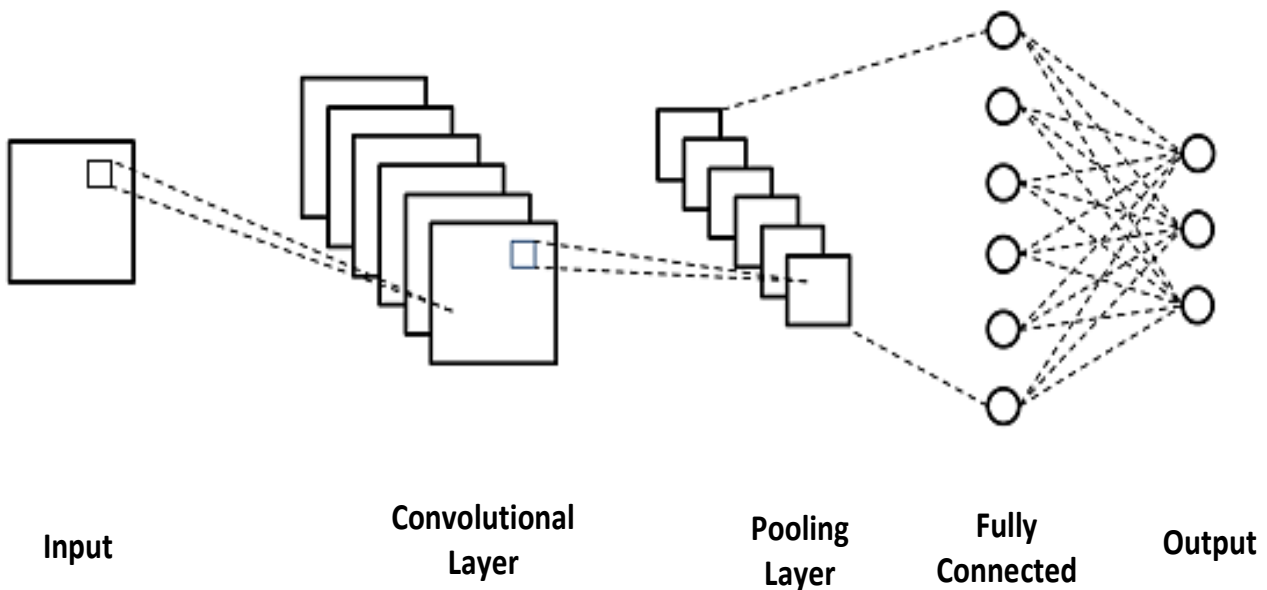


Figure 13: Basic CNN Architecture [32].

Convolution is a specialized kind of linear operation [33]. The basic CNN architecture is represented in Figure 13. This figure shows that the CNN architecture comprises a series of layers that transform the input image data to generate the desired output.

A. Types of CNN Layers

Convolutional neural networks (CNNs) typically consist of a series of layers, each serving a specific purpose in extracting features and learning hierarchical representations. These layers include convolutional layers, max pooling layers, and fully connected layers. Below is a description of these different layers:

– Convolutional Layers

A CNN is mainly composed of layers called convolutional layers that filter their layer inputs to find useful features within those inputs. This filtering operation is called convolution, which gives rise to the name of this kind of neural network [34].

The convolution operation uses a set of learnable filters (kernels). The filters slide across the image and perform element-wise multiplication with the input pixels to produce a feature map. The resulting feature maps highlight different aspects of the image, such as edges, corners, and textures, that are relevant to the task at hand [35]. Convolutional layers are typically followed by activation functions such as ReLU to introduce non-linearity in the model. Figure 14 shows the 2D convolution operation on an image and its result. It is important to remember that the filter has a depth that matches the depth of the input [34].

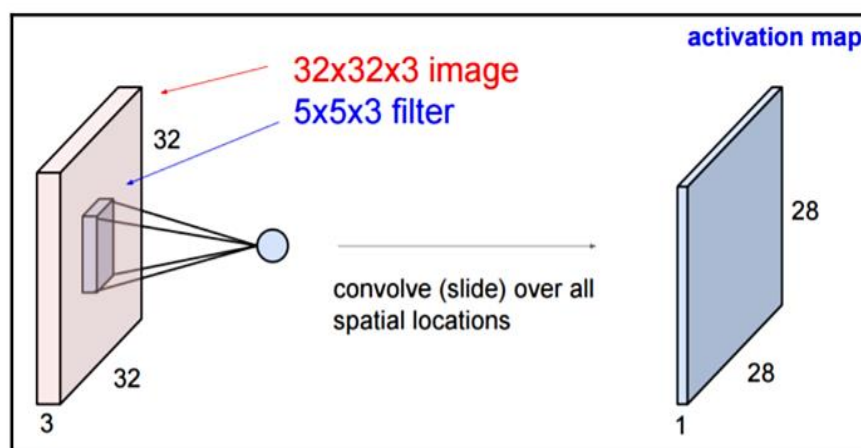


Figure 14: Convolution Operation [31].

To calculate the size of the feature map resulting from a convolutional operation, we can use the following formula:

$$\text{Output Size} = \left\lfloor \frac{\text{Input Size} - \text{Filter Size} + 2 \times \text{Padding}}{\text{Stride}} \right\rfloor + 1$$

Where [36]:

Input Size: The size (width or height) of the input feature map.

Filter size: The size of the convolutional filter/kernel.

Padding: The number of zero-padding pixels added to the input feature map. It helps to preserve the spatial dimensions of the input and output feature maps.

Stride: The step size at which the filter/kernel slides over the input feature map.

– Pooling Layers

These layers down sample the feature maps produced by the convolutional layers by aggregating adjacent pixels. This reduces the spatial size of the feature maps and makes the model more computationally efficient.

One of the big advantages of pooling, that it has no parameters to learn, is also its biggest disadvantage because pooling can end up just throwing important information away. As a result, pooling is starting to be used less frequently in CNN now [34]. Figure 2-15 shows the most common type of pooling “max-pooling”, it slides a window, like a normal convolution, and then at each location, sets the biggest value in the window as the output.

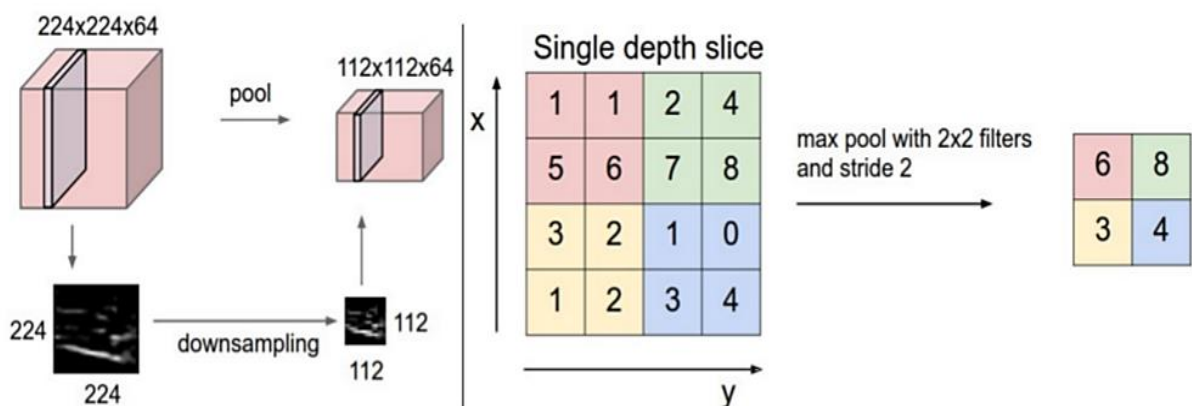


Figure 15: Illustration of Max Pooling [34].

– Fully Connected Layers

These layers are similar to the ones used in traditional neural networks. They take the flattened output from the preceding convolutional and pooling layers and produce the final output by applying a set of weights and biases. Fully connected layers are typically used at the end of the network to produce a classification or regression output. The Figure 2-16 illustrate the usage of Fully Connected Layers in the classification section of CNNs.

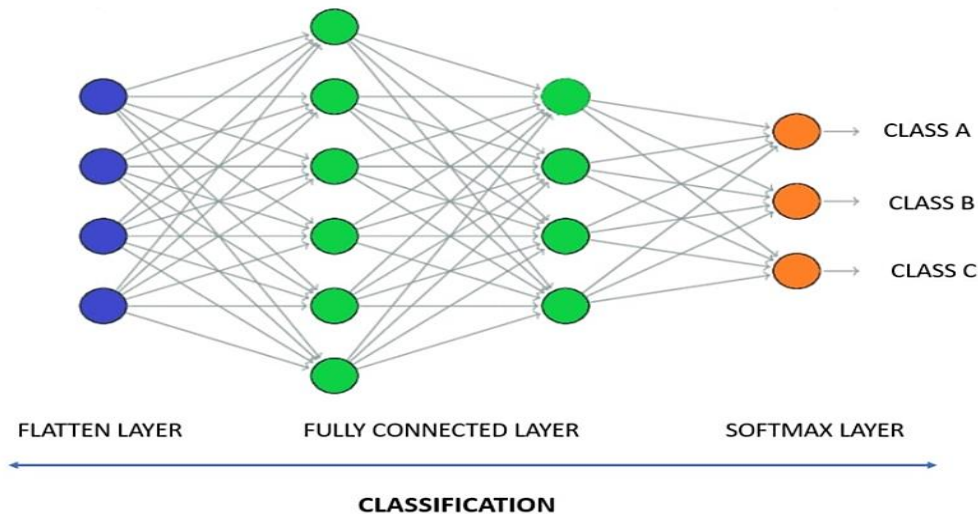


Figure 16: The usage of Fully Connected Layers in the classification section of CNNs [37].

B. CNNs Applications

Convolutional Neural Networks (CNNs) have revolutionized Machine Learning and Artificial Intelligence, powering a vast array of applications. Thus, understanding their unique uses and advantages provides valuable insights into the future of computer vision and beyond. Here are some notable applications of CNNs:

– Image Classification

Image classification involves categorizing an entire image into a particular class or label. CNNs have significantly advanced this field, allowing for accurate and efficient classification [35].

Use Cases: Facial recognition, disease diagnosis through medical imagery, categorizing objects in satellite imagery, etc.

– Object Detection

Object detection goes beyond classification and also identifies the location of objects within an image. Several CNN-based methods have been developed for this. R-CNN, Fast R-CNN, Faster R-CNN: A series of models that use region proposals and CNNs to detect objects.

SSD (Single Shot Multibox Detector): Detects objects in a single forward pass of the network, making it faster than the R-CNN series.

YOLO (You Only Look Once): Even faster than SSD, YOLO divides the image into a grid and predicts bounding boxes and class probabilities simultaneously [35].

Use Cases: Self-driving cars, retail (identifying products on shelves), security (detecting suspicious activities), etc.

– Semantic Segmentation

Semantic segmentation involves classifying each pixel in an image, leading to a detailed, pixel-level understanding of the image's contents [35].

Use Cases: Medical imaging (e.g., tumor detection), autonomous driving (e.g., road segmentation), agricultural field monitoring, etc.

2.3.3. Transfer Learning

The idea that people can apply knowledge already learned from one task to faster and with better success, led to the study of transfer learning [39]. Transfer learning is the reuse of a pre-trained model on a new problem. It's currently very popular in deep learning because it can train deep neural networks with comparatively little data. This is very useful in the data science field since most real-world problems typically do not have millions of labeled data points to train such complex models [40].

A. Definition

Transfer Learning is a Machine Learning technique whereby a model is trained and developed for one task and is then re-used on a second related task. It refers to the situation whereby what has been learnt in one setting is exploited to improve optimization in another

setting. Transfer Learning is usually applied when there is a new dataset smaller than the original dataset used to train the pre-trained model [41].

Figure 17 shows more details about the transfer learning method, where Task 1 is fully trained using a large dataset, typically a dataset with millions of images like ImageNet. And Task 2 uses the knowledge obtained from Task 1 to improve accuracy and learn faster.

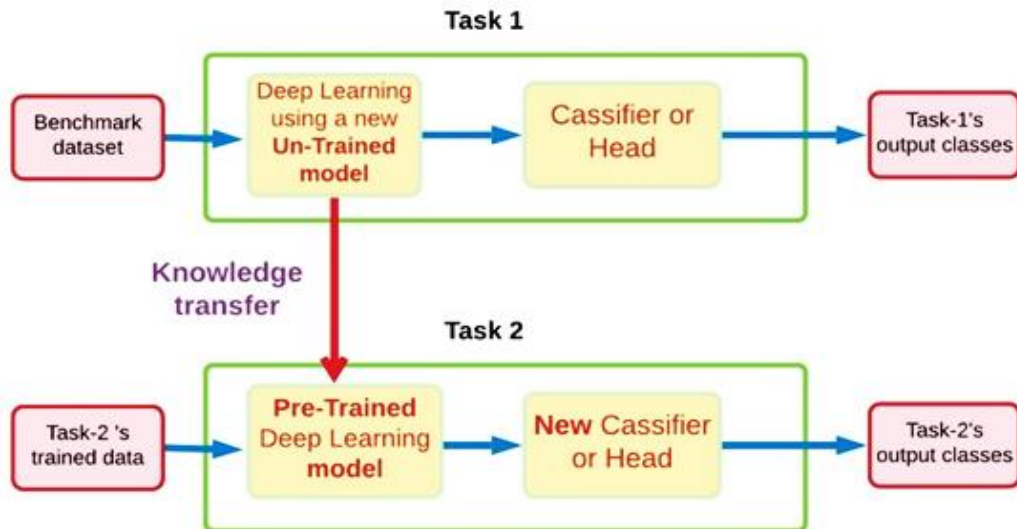


Figure 17: Transfer Learning Diagram [42]

With Transfer Learning, we basically try to exploit what has been learned in one task to improve generalization in another. We transfer the weights that a network has learned at “task A” to a new “task B”.

B. Popular Pre-Trained Models

Several pre-trained models have gained popularity due to their effectiveness across various tasks. Here are some of the most notable ones:

1. VGGNet (Visual Geometry Group)

VGGNet, is a deep convolutional neural network architecture designed by the Visual Geometry Group at the University of Oxford. It was introduced by Karen Simonyan and Andrew Zisserman in their paper “Very Deep Convolutional Networks for Large-Scale Image Recognition” in 2014. Innovative object identification models are built using the VGG architecture. The VGGNet, created as a deep neural network, outperforms benchmarks on a variety of tasks and datasets outside of ImageNet. It also remains one of the most often used image recognition architectures today [43].

The original VGGNet consists of 16 or 19 layers, depending on the variant. The 16-layer version is referred to as VGG-16, while the 19-layer version is called VGG-19.

Their architectures are represented in Figure 18.

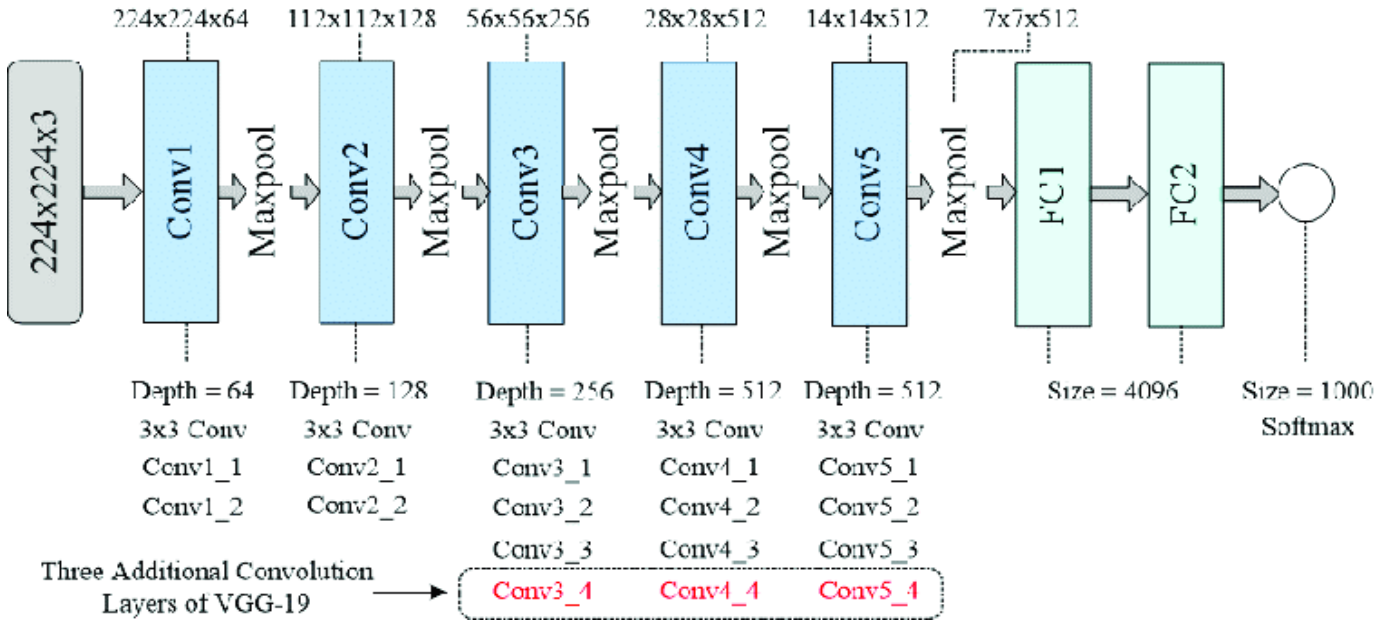


Figure 18: Architecture of VGG-16 and VGG-19 [44].

As shown in Figure 18, VGG-16 architecture consists of 13 convolutional layers utilizing 3x3 filters, each followed by rectified linear unit (ReLU) activation functions. These convolutional layers are interspersed with max-pooling layers using 2x2 filters with a stride of 2. The network concludes with three fully connected layers, each comprising 4096 neurons, also followed by ReLU activation. The final layer consists of 1000 neurons for ImageNet classification, typically followed by a softmax activation function.

– VGG-19

The VGG19 architecture has the same basic idea as the VGG16 model, with the exception that it supports 19 layers. The numbers “16” and “19” refer to the model’s weight layers (convolutional layers). In comparison to VGG16, VGG19 contains three extra convolutional layers [44].

2. ResNet (Residual Network)

ResNet short for Residual Networks is a classic neural network used as a backbone for many computer vision tasks. The fundamental breakthrough with ResNet was it allowed us to train extremely deep neural networks with 150+ layers successfully [45].

It is an innovative neural network that was first introduced by Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun in their 2015 computer vision research paper entitled: “Deep Residual Learning for Image Recognition”.

Convolutional Neural Networks have a major disadvantage — “Vanishing Gradient Problem”. During backpropagation, the value of gradient decreases significantly, thus hardly any change comes to weights. To overcome this, ResNet is used. It makes use of “Skip Connection”, which is a direct connection that skips over some layers of the model. It adds the original input to the output of the convolutional block as shown in Figure 19.

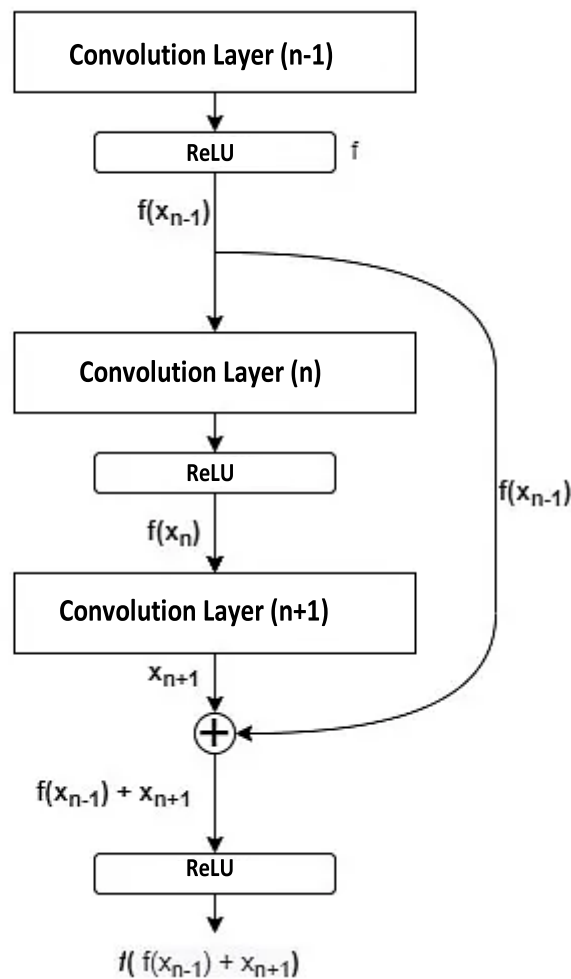


Figure 19: Skip Connection [46].

The ResNet architecture has several variants, including ResNet-18, ResNet-34, ResNet-50, ResNet-101, and ResNet-152. The number in each variant corresponds to the number of layers in the Network. For example, ResNet-50 has 50 layers, while ResNet-152 has 152 layers. Their architectures are shown in Table 1.

layer name	output size	18-layer	34-layer	50-layer	101-layer	152-layer
conv1	112×112	7×7, 64, stride 2				
		3×3 max pool, stride 2				
conv2_x	56×56	$\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$
conv3_x	28×28	$\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 8$
conv4_x	14×14	$\begin{bmatrix} 3 \times 3, 256 \\ 3 \times 3, 256 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 256 \\ 3 \times 3, 256 \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 23$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 36$
conv5_x	7×7	$\begin{bmatrix} 3 \times 3, 512 \\ 3 \times 3, 512 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 512 \\ 3 \times 3, 512 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$
	1×1	average pool, 1000-d fc, softmax				
FLOPs		1.8×10^9	3.6×10^9	3.8×10^9	7.6×10^9	11.3×10^9

Table 1: Architectures of different variants of ResNets [47].

3. DenseNet (Densely Connected CNN)

In a traditional feed-forward Convolutional Neural Network (CNN), each convolutional layer except the first one (which takes in the input), receives the output of the previous convolutional layer and produces an output feature map that is then passed on to the next convolutional layer. Therefore, for 'L' layers, there are 'L' direct connections; one between each layer and the next layer [48]. However, as the number of layers in the CNN increases, the 'vanishing gradient' problem arises.

This means that as the path for information from the input to the output layers increases, it can cause certain information to 'vanish' or get lost which reduces the ability of the network to train effectively. DenseNets resolve this problem by modifying the standard CNN architecture and simplifying the connectivity pattern between layers.

In a DenseNet architecture, each layer is connected directly with every other layer, hence the name Densely Connected Convolutional Network. For 'L' layers, there are $L(L+1)/2$ direct connections. In each layer, the feature maps of all the previous layers are not summed, but concatenated and used as inputs [49]. Consequently, DenseNets require fewer parameters than an equivalent traditional CNN, and this allows for feature reuse as redundant feature maps are discarded.

In DenseNet architecture the l th layer receives the feature-maps of all preceding layers, x_0, \dots, x_{l-1} , as input:

$$X_l = H_l([X_0, X_1, \dots, X_{l-1}])$$

Where $[x_0, x_1, \dots, x_{l-1}]$ is the concatenation of the feature-maps, i.e., the output produced in all the layers preceding l ($0, \dots, l-1$). The multiple inputs of H_l are concatenated into a single tensor to ease implementation [49].

2.4. Literature Review

This section provides a review of previous literature that utilizes Convolutional Neural Networks (CNNs) for the classification of agricultural images, encompassing crop types and vegetation. Additionally, it explores some literature employing Geographic Information Systems (GIS) for agricultural management and mapping plant distribution, as well as studies that integrate both GIS and CNN techniques.

2.4.1. CNN-based Approaches for Agricultural Image Classification

In recent years, Convolutional Neural Networks (CNNs) have emerged as a cornerstone technology in image classification tasks, revolutionizing various fields including agriculture. These networks excel at automatically extracting intricate spatial features and patterns from images, enabling precise classification of various elements such as crop types and vegetation coverage. In this review, we explore some of the existing literature on CNN-based approaches for agricultural images classification.

[Bouguettaya et al \(2022\)](#), reviewed the recent CNN-based methods for crop/plant classification to help researchers and farmers to decide what algorithms they should use accordingly to their studied crops and the used hardware. The readers of this review could acquire the most challenging issues facing researchers to classify different crop types and their potential solutions to improve the performance of deep learning-based algorithms [50].

[Kamilaris et al \(2018\)](#), surveyed the research efforts that employ convolutional neural networks (CNNs) applied to various agricultural challenges. In particular, plant recognition, crop type classification, and land cover classification. All papers discussed were published after 2014. In this research, they examined agricultural problems under study, models employed, sources of data used, and the overall accuracy achieved according to the performance metrics used by the authors. Convolutional neural networks are compared with other existing techniques. Their findings indicate that CNN constitutes a promising technique with high performance in terms of classification accuracy [51].

[Teixeira et al \(2023\)](#), evaluated the effectiveness of deep learning techniques for crop classification using aerial imagery. The reviewed papers highlighted that deep learning

techniques, particularly those based on CNNs, are commonly used for crop classification and tend to outperform machine learning models when sufficient data is available. This systematic review provides insights into the current state of deep learning models for crop classification and highlights important factors that affect their performance, including the requirement for large amounts of training data and the incorporation of non-crop classes to enhance accuracy. These findings can assist researchers and practitioners in selecting appropriate models and datasets for crop classification tasks [52].

AJAYI et al (2023), investigated the application of ResNet architectures, to automate crop type identification. They used a dataset consisting of 1,488 aerial images covering the study area. These images are employed to train three distinct ResNet architectures, namely ResNet-50, ResNet-101, and ResNet-152. The evaluation of these models was based on accuracy and processing time. Notably, ResNet-50 emerged as the most proficient, achieving an accuracy rate of 82%, while ResNet-101 and ResNet-152 architectures achieved less accuracy. They concluded that the ResNet-50 architecture, even with a limited dataset, is a valuable tool for precise crop-type classification within the precision agriculture domain [53].

Chew et al (2020), used RGB images collected from unmanned aerial vehicles (UAVs) to develop a deep learning algorithm for identifying crop types, specifically bananas, maize, and legumes [54]. Their model leverages advances in deep convolutional neural networks and transfer learning, employing the VGG16 architecture and the publicly accessible ImageNet dataset for pretraining. The developed model performs with an overall test set F1 of 0.86, with individual classes ranging from 0.49 to 0.96 [55].

Nowakowski et al (2021), used high-resolution RGB drone images as input data for the classification performed using a transfer learning approach. In this study, they used VGG16 and GoogLeNet, which are pre-trained CNNs. They found that thanks to the transferred knowledge, the proposed models can successfully classify the studied crop types with high overall accuracy for two considered cases, achieving up to almost 83% for the Malawi dataset and up to 90% for the Mozambique dataset. Based on their experiments with different TL approaches, they show that the number of frozen layers is an important parameter of TL and a fine-tuning of all the CNN weights results in significantly better performance than the approaches that apply fine-tuning only on some numbers of last layers [56].

2.4.2. Applications of GIS in Agricultural Management and Spatial Mapping

Geographic Information Systems (GIS) have emerged as indispensable instruments in the modernization of agricultural practices. This section provides a review of some of the existing literature on various applications of GIS in agricultural management and spatial mapping.

Mathenge et al (2022), reviewed existing evidence on GIS application in enhancing evidence-informed policy and practice for improving agriculture sustainability, and identifying obstacles to their application, particularly in low- and middle-income countries. They identified 2,113 articles published between 2010–2021. Their results show that GIS technology application in agriculture has gained prominence in the last decade, with 66% of selected papers being published in the last six years. The main GIS application areas identified included: Crop pattern mapping, crop yield estimation, soil fertility assessment, drought assessment, pest and crop disease detection and management, precision agriculture, and fertilizer and weed management [57]. “

Biswas et al (2017), created maps of medicinal plants using Geographic information system (GIS). In this study, a total of 56 plant species from 33 families were identified and their exact GPS locations were recorded to map their geographical distribution through creation of various thematic maps under the GIS environment. The multilayered database so created had information like location of plant species, utilities, frequency of occurrence, etc. The thematic maps and attribute data were integrated through ArcGIS [58].

Sonti et al (2015), explored the potential application of Geographic Information System (GIS) technology in forest management in 3 African countries, in order to make better decisions, improve productivity, save time, money and manpower in forest management activities. The range of applications reviewed in their article is clear evidence of the significant value of forests and the potential of GIS to aid in their management, and they reach a number of broad conclusions about the role of GIS in forestry [59].

2.4.3. Combining GIS and CNN in Plant Classification and Mapping

In the realm of plant classification and mapping, the integration of Geographic Information Systems (GIS) with Convolutional Neural Networks (CNNs) marks a pivotal advancement. Below are some of the recent studies in which GIS and CNNs have been combined. CNNs are used in the classification task, and GIS is used to store results and mapping.

Abd Mubin et al (2019), combined Geographic Information system (GIS) with deep learning approach, where they used two different convolution neural networks (CNNs) to detect young and mature oil palm separately in satellite imagery, and then used GIS software during data processing and result storage process [60], and create an oil palm prediction map for mature and young oil palm [61].

Garlito et al (2023), developed a new methodology capable of automatically finding the geolocation of the most frequent areas of accumulation of invasive aquatic plants in the Guadiana River, Spain. They used deep learning [a convolutional neural network (CNN)] to detect these plants, a subsequent analysis is carried out using geographic information systems (GIS) to map the areas where water hyacinth patches are most frequently found [62].

Wang et al (2023), discussed a highly practical and effective working pipeline to weed map a wheat field combining GIS and deep learning technology. This pipeline is an end-to-end process including using an unoccupied aerial vehicle (UAV) to collect ultra-high-definition whole-field images, labeling and training deep learning models and an efficient evaluation process for the resulting weed map. They showed that their method can generate accurate weed maps by only training on small regions of the field [54].

2.4.4. Summary

The recent literature reviewed in this section provides valuable insights into the application of convolutional neural networks (CNNs) and geographic information systems (GIS) in agricultural practices, ranging from crop and plant classification to spatial mapping and management. They demonstrated the effectiveness of CNN-based approaches, especially when integrated with transfer learning, for accurately classifying crops and plants through aerial imagery.

The significance of sufficient datasets, model selection, and parameters optimization is underscored to maximize classification performance. The literature accentuates the superior performance of CNNs in contrast to alternative techniques. Additionally, the section highlighted the diverse applications of Geographic Information System (GIS) technology in agricultural management, including mapping the distribution of medicinal plants, forest management, crop pattern mapping, etc.

Furthermore, the studies that combine the power of GIS with deep learning, particularly CNNs, have demonstrated the synergistic potential of these technologies when combined together, in plant detection, and mapping their distribution.

2.5. Conclusion

This chapter has provided a comprehensive overview of Geographic Information Systems (GIS), Deep Learning, and their applications in agricultural research, particularly through a review of relevant literature.

The exploration of GIS fundamentals established a foundation for understanding spatial data, its various models (vector and raster), and its role in creating informative maps.

Deep learning, specifically Convolutional Neural Networks (CNNs), were then introduced, delving into their architecture, components, and applications. The concept of transfer learning was explored, highlighting its effectiveness in leveraging pre-trained models for image classification tasks.

The literature review section provided valuable insights into the synergy between these technologies. Studies showcased the successful application of CNNs for agricultural image classification, particularly when combined with transfer learning techniques. The importance of data quality, model selection, and parameter optimization for achieving optimal classification performance was emphasized. Additionally, the review highlighted the diverse applications of GIS in agricultural management and spatial mapping.

Finally, the chapter emphasized the potential of combining GIS and CNNs. Studies demonstrated the effectiveness of this approach in plant detection and mapping their distribution across landscapes. This synergistic approach offers exciting possibilities for future advancements in agricultural practices.

3. Chapter III – Methodology and Results

3.1.	Introduction	36
3.2.	Used tools	36
3.2.1.	Python	36
3.2.2.	Anaconda.....	37
3.2.3.	Jupyter Notebook.....	37
3.2.4.	TensorFlow	37
3.2.5.	Keras	37
3.2.6.	NumPy	38
3.2.7.	Matplotlib.....	38
3.2.8.	Kaggle	38
3.2.9.	Postgres	38
3.2.10.	PostGIS.....	39
3.2.11.	Quantum Geographic Information System (QGIS).....	39
3.2.12.	Qt Designer.....	39
3.2.13.	PyQt5	40
3.2.14.	PIL	40
3.2.15.	Geoserver	40
3.2.16.	Google Earth Pro	41
3.3.	Model Development	41
3.3.1.	Data Collection	41
3.3.2.	Data Preprocessing	43
3.3.3.	Experiments and Results	45
3.3.4.	Discussion	52
3.4.	Geospatial Data Processing and Analysis	54
3.4.1.	Project Initialization and Data Preparation	55
3.4.2.	Plugin Development	56
3.4.3.	Classification and Analysis	57
3.4.4.	Connecting QGIS to PostgreSQL and Exporting Data	59
3.4.5.	Publishing Data in GeoServer.....	59
3.4.6.	Observations	62
3.5.	Conclusion	63

3.1. Introduction

This chapter outlines the methodology and results of our study, demonstrating the practical application of our models. We describe the CNN architecture trained from scratch to classify aerial images of agricultural and other areas, and test transfer learning performance with VGG16, DenseNet121, and ResNet50 models, comparing the results. Tools, datasets, and preprocessing steps are detailed to provide a comprehensive account of our experimental procedures and results. Additionally, we utilized QGIS to manage and analyze spatial data for the Assafia region. A new project was created, incorporating different layers. A custom QGIS plugin was developed to classify images from diverse location of Assafia by land type. PostGIS and PostgreSQL were integrated to manage spatial data, with layers exported from QGIS and published via GeoServer. This setup enabled visualization and interaction with the data using Google Earth Pro.

3.2. Used tools

In this section, we describe the tools and libraries we use in our work steps.

3.2.1. Python

Python is an interpreted, high-level programming language with dynamic semantics. Python's simple, easy to learn syntax emphasizes readability and therefore reduces the cost of program maintenance. It was first released in 1991 by Guido van Rossum and has since become one of the most popular programming languages in the world. It is widely used for a variety of purposes, including web development, scientific computing, data analysis, artificial intelligence, and machine learning.



Figure 20 : Python Logo [63].

3.2.2. Anaconda

Anaconda is an open-source distribution of the Python programming language. It includes a wide range of Python packages, tools, and environments for scientific computing, data analysis, and machine learning. Package versions in Anaconda are managed by the package management system, conda, which analyzes the current environment before executing an installation to avoid disrupting other frameworks and packages [64].



Figure 21 : Anaconda Logo [65].

3.2.3. Jupyter Notebook

Jupyter Notebook is a web-based interactive computing environment that you can use to create and share documents that contain live code, equations, visualizations, and text. It is widely used in data science and scientific computing for exploring and visualizing data, prototyping machine learning models, and sharing research and analysis results. It allows users to write, execute, and debug code within the note-book environment.

3.2.4. TensorFlow

TensorFlow is an open-source machine learning framework developed by Google. It can be used across a range of tasks but has a particular focus on training and inference of deep neural networks. TensorFlow enables developers to design dataflow graphs, which are structures that define how data flows via a graph or set of processing nodes. Each node in the graph symbolizes a mathematical process, and each edge between nodes is a tensor, a multi-layered data array. It can be used in a wide variety of programming languages, including Python, JavaScript, C++, and Java, facilitating its use in a range of applications in many sectors [66].

3.2.5. Keras

Keras is the high-level API of the TensorFlow platform. It provides an approachable, highly-productive interface for solving machine learning (ML) problems, with a focus on modern deep learning. Keras covers every step of the machine learning workflow, from data processing to hyperparameter tuning to deployment. With Keras, you have full access to the scalability and cross-platform capabilities of TensorFlow [67].

3.2.6. NumPy

NumPy is the fundamental package for scientific computing in Python that provides powerful tools for working with arrays, matrices, and other multidimensional data structures, and includes a wide range of mathematical functions [68].

3.2.7. Matplotlib

Matplotlib is an open-source Python library. It is used to create high-quality plots, histograms, bar charts, and various types of graphs with just a few lines of code. It's a comprehensive tool that enables the generation of highly detailed data visualizations [69].

3.2.8. Kaggle

A subsidiary of Google, it is an online community of data scientists and machine learning engineers, with powerful tools and resources to help you achieve your data science goals. It allows users to find datasets they want to use in building AI models, publish datasets, work with other data scientists and machine learning engineers, and enter competitions to solve data science challenges [70].

3.2.9. Postgres

PostgreSQL is a robust, open-source object-relational database system that enhances SQL with advanced features for securely storing and scaling complex data workloads. It supports developers in building applications, ensures data integrity for administrators, and manages datasets of any size. Highly extensible, PostgreSQL allows custom data types, functions, and code in various programming languages without recompilation. With the PostGIS extension, it offers powerful tools for storing, querying, and analyzing geographic datasets, making it ideal for geospatial data management.



Figure 22 : PostgreSQL Logo [71]

3.2.10. PostGIS

PostGIS is a free and open-source software program that extends the capabilities of the PostgreSQL object-relational database system. Initially developed by Refractions Research Inc. as a spatial database technology research project, PostGIS allows users to store, index, and query geospatial data within a PostgreSQL database.



Figure 23 : PostGis Logo [72].

This functionality makes it a valuable tool for Geographic Information Systems (GIS) applications.

3.2.11. Quantum Geographic Information System (QGIS)

QGIS (Quantum Geographic Information System) is a free and open-source software program that allows users to work with geographic data. It provides tools for creating maps and visualizing data, analyzing spatial patterns, and managing data tied to specific locations and overall geographic information.



Figure 24: QGIS Logo [73]

3.2.12. Qt Designer

Qt Designer is a comprehensive tool for designing and constructing graphical user interfaces (GUIs) using Qt. It offers a user-friendly drag-and-drop environment for creating windows, dialogs, and other interface elements, simplifying the process of designing complex UIs without the need for code.

The designs are saved as .ui files, which can be converted to source code and integrated for example into Qgis using tools like PyQt5.



Figure 25: Qt Designer Logo [74]

3.2.13. PyQt5

PyQt5 is a set of Python bindings for the Qt application framework, allowing Python developers to create cross-platform applications with native-looking GUIs. It integrates seamlessly with Qt libraries, providing a comprehensive set of tools to design and manage user interfaces. QGIS uses PyQt5 to handle its graphical user interface.

3.2.14. PIL

PIL refers to the "Python Imaging Library", a Python module for working with image data. It provides functionalities for opening, manipulating, and saving various image formats. In QGIS, PIL can be utilized within Python scripts or plugins to handle image processing tasks, such as raster data manipulation or generating image outputs from geospatial data.

3.2.15. Geoserver

GeoServer is an open-source server for sharing geospatial data online. It supports various formats, including KML, and serves data through protocols like WMS, WFS, and WCS. GeoServer facilitates the publication of spatial data stored in PostgreSQL databases, including data managed with PostGIS extensions, allowing users to visualize and share geospatial data from PostgreSQL/PostGIS databases via GeoServer's web services. Additionally, GeoServer enables data visualization in applications like Google Earth Pro by providing data through protocols such as WMS.



Figure 26: GeoServer Logo [75].

3.2.16. Google Earth Pro

Google Earth Pro is a desktop application by Google, serving as a robust Geographic Information System (GIS) tool. It enables users to explore the Earth's surface through satellite imagery, aerial photography, and 3D terrain models. Specifically designed for GIS applications, Google Earth Pro allows users to visualize and analyze geographical data layers from servers like GeoServer. It offers interactive maps, customizable layers, and facilitates spatial analysis, urban planning, environmental monitoring, and more.



Figure 27: Google Earth Pro Logo [76]

3.3. Model Development

3.3.1. Data Collection

The selection of suitable datasets is crucial for training and evaluating machine learning models effectively. In this study, we collected two datasets from Kaggle to train our convolutional neural network (CNN) models.

A. AID: A Scene Classification Dataset

Aerial Image Dataset (AID) is a new large-scale aerial image dataset, created by collecting sample images from Google Earth imagery [77], actually this dataset has 30 aerial scene types: airport, bare land, baseball field, beach, bridge, center, church, commercial, dense residential, desert, farmland, forest, industrial, meadow, medium residential, mountain, park, parking, playground, pond, port, railway station, resort, river, school, sparse residential, square, stadium, storage tanks and viaduct. And about 200 to 400 samples in each class.

The images in AID are actually multi-source, as Google Earth images are from different remote imaging sensors. Moreover, all the sample images per each class in AID are carefully chosen from different countries and regions around the world, mainly in China, the United States, England, France, Italy, Japan, Germany, etc. and they are extracted at different times and seasons under different imaging conditions, which increases the intra-

class diversities of the data [78]. Some samples with their classes of AID dataset are shown in Figure 28.

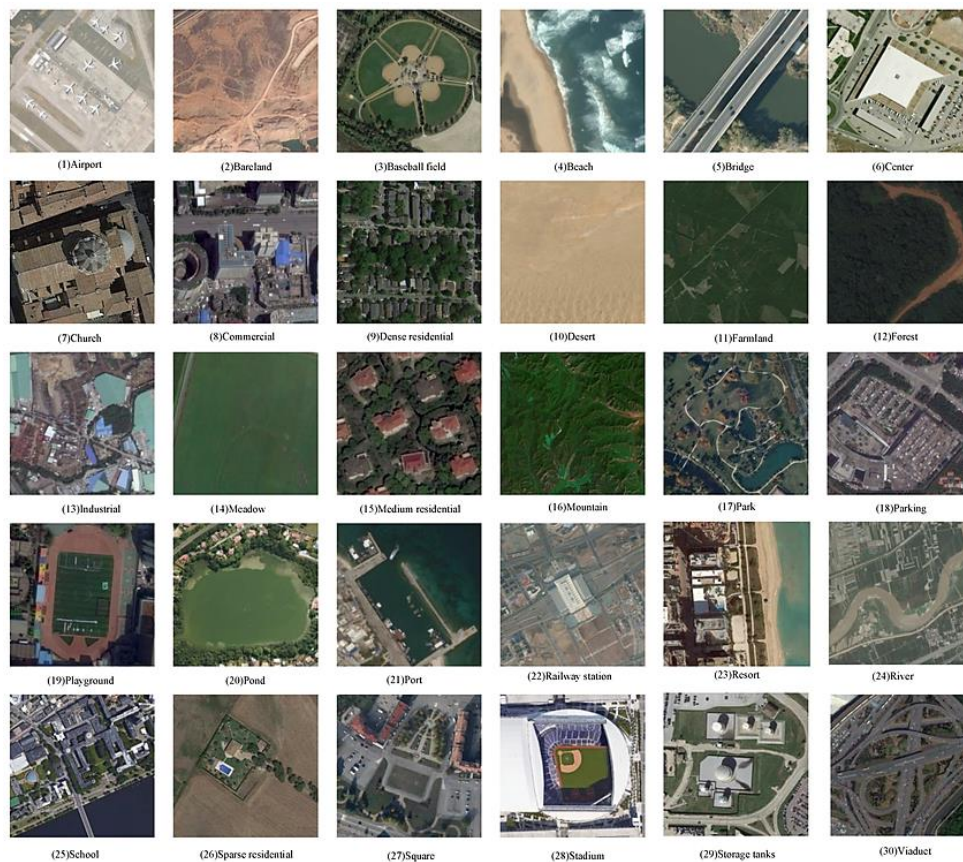


Figure 28: Some samples with their classes in AID dataset [79].

From this dataset, we focus only on the categories relevant to our task, including farmland, meadow, forest and all other categories are grouped under a category called “Others”.

We change the number of images in each class to 250 images, this helps in training the CNN more effectively. It ensures that the network learns from a balanced dataset, preventing biases towards overrepresented or underrepresented categories. This balance is crucial for the CNN to generalize well to unseen data. Consequently, we have the following scene classes: farmland, meadow, forest and Others, with 250 images in each class.

A. RSSCN7: Remote Sensing Scene Classification Dataset

This dataset is specifically designed for scene classification tasks in remote sensing imagery. It consists of seven scene categories, including: Grass, Field, Industry, River Lake, Forest, Resident and Parking [80]. With 400 images in each class. This dataset is rather challenging due to the wide diversity of the scene images that are captured under changing seasons and varying weathers and sampled on different scales [81].

In this data set also, we focus only on the categories relevant to our task, which are field, forest, grass, and we consider the rest of categories as a one category “Others”. Figure 29 represents some samples in each category in this dataset.



Figure 29: Some collected samples with their classes in the RSSCN7 dataset.

After the data collection process, another step is necessary for the success of any machine learning algorithm. It is the pre-processing of collected data in order to train powerful CNN models and achieve high accuracy.

3.3.2. Data Preprocessing

The raw image data collected from Kaggle required preprocessing before it could be used for training the CNN models. The preprocessing steps included data splitting, resizing the images, normalizing the pixel values, and data augmentation.

A. Data Splitting

To facilitate robust model training and evaluation, we use split folders tool which is a Python package designed to efficiently divide a directory of data into separate train, validation, and test sets, we use a ratio of 80:10:10 for both datasets.

B. Resizing the Images

The raw images in the two datasets were of different sizes. For the CNN model built from scratch, we resized all images to a consistent size of 112 x 112 pixels, while transfer learning architectures including VGG16, DenseNet121, and ResNet50 require input images of 224 x 224 pixels. This ensures that all images are of the same size, which is required to feed data into the CNN model.

C. Normalizing the Pixel Values

To improve the convergence of the CNN model during training, we normalize the pixel values of the images to be between 0 and 1. This step aids in stabilizing the learning process and improving the performance of models on unseen data.

D. Data Augmentation

We also applied data augmentation to increase the size of the training dataset and its diversity, improve the model's robustness and prevent overfitting. For this we used a utility provided by the Keras library called ImageDataGenerator. Which is a high-level neural networks API running on top of TensorFlow or Theano. It is used for real-time data augmentation during training of the CNN model. The Data augmentation techniques used include Rotation, Horizontal and Vertical Flip, Width and Height Shift, Shear, and Zoom. After the data preprocessing steps there is another essential step. Choosing appropriate convolutional neural network architecture.

3.3.3. Experiments and Results

As a first step in our study, we built a CNN model and trained it from scratch on the two datasets separately. Then, using Transfer Learning, we tried to adapt the pre-trained models we selected to our datasets in order to identify the most appropriate model for each dataset. VGG16, DenseNet121 and ResNet50 are the models used. In the second step we combined the two datasets into a larger one in order to increase the diversity of the data.

Step 1: Individual Datasets Analysis

First, we conducted the study on the AID dataset. We built a CNN model and trained it on this dataset. We then adapted pre-trained models including VGG16, DenseNet121, and ResNet50 to this dataset, in order to improve the accuracy results and reduce overfitting. Next, we replicated the identical procedures on the second dataset.

A. Model Building

Our Convolutional Neural Network architecture for both datasets take an input shape of (112, 112, 3), indicating images of size 112x112 pixels with three color channels (red, green, blue). The model consists of six convolutional layers, each followed by rectified linear unit (ReLU) activation functions. In some convolutional layers we used padding=same, this ensures that the spatial dimensions of the input are preserved by the convolutional layers, which can help in capturing features effectively.

Each convolutional layer has 128 filters, except for the first layer, which contains 64 filters. Each pair of convolutional layers is followed by a Max-pooling layer with a pool size of 2x2 and a stride of 2, reducing spatial dimensions. A dropout layer with a rate of 0.5 is employed for regularization to prevent overfitting.

The feature maps are then flattened and passed through a dense layer with 1024 units and ReLU activation function. Finally, a dense layer containing a number of units equal to the number of classes in each dataset, and softmax activation is added for multi-class classification, producing probabilities for each class.

This architecture is illustrated in Figure 30.

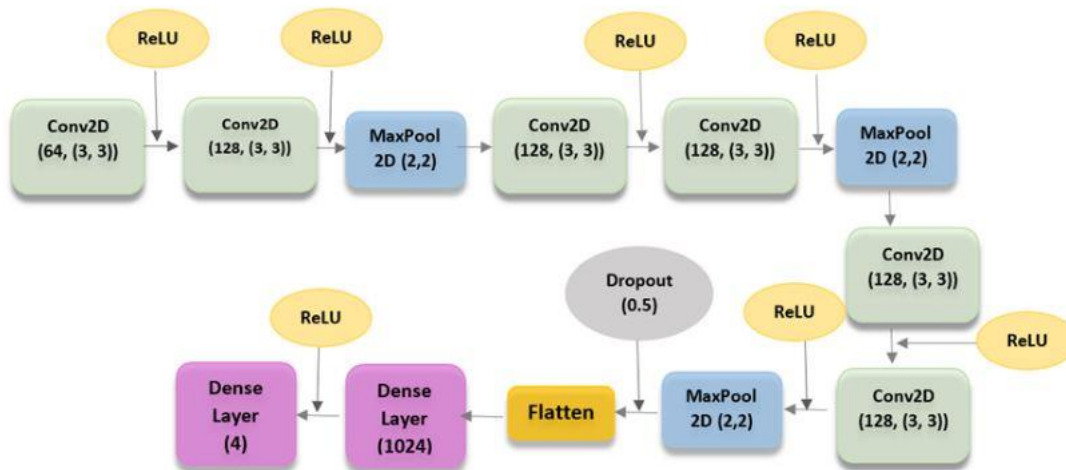


Figure 30: The CNN Architecture.

We compiled our model for both datasets using Adam optimizer with a learning rate of 0.0001, and categorical cross-entropy as the loss function, and accuracy as the evaluation metric.

We trained our CNN model on the two training datasets separately. We monitored the model's performance on the validation set to detect overfitting and adjust hyperparameters accordingly.

To evaluate the performance of the trained model, we use separate test sets that are completely different from those used in training, in order to evaluate their generalization ability.

B. Transfer Learning Methodology

To transfer learning, we use the methodology shown in Figure 31, starting with data collection and preprocessing to evaluate the model.

The data collection and preprocessing steps remain consistent with what we discussed previously.

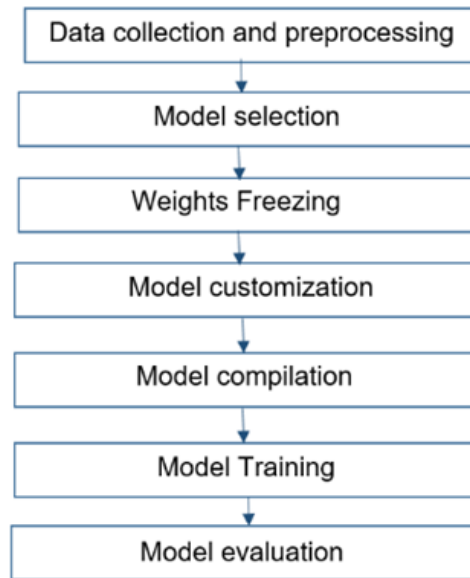


Figure 31: Transfer Learning Methodology.

– **Model Selection**

We selected the pre-trained VGG16, DenseNet121, and ResNet50, which are trained on the ImageNet dataset. They learned general features from a vast amount of data. By using these pre-trained models as a starting point, we can transfer this knowledge to our new task with smaller datasets. This often leads to better performance, faster convergence, and reduced need for large amounts of labeled data. We loaded the pre-trained models without the top classification layer using the following code:

```
vgg = VGG16(weights = 'imagenet', include_top = False, input_shape = (224, 224, 3))
```

– **Freezing the Weights of the Pre-trained Layers**

We freeze the weights of the pre-trained layers. It means that the parameters (weights and biases) of layers that have been pre-trained on a large dataset, will not be updated during training, by using the following code:

```
for layer in vgg.layers: layer.trainable = False [82].
```

– **Model Customization**

We added new layers tailored to our specific task on top of the remaining layers of the pre-trained models. These new layers include Dropout and Dense layers.

– **Model Compilation**

We compiled all transfer learning models used for both datasets using Adam optimizer, and categorical cross-entropy as the loss function, and accuracy as the metric to monitor during training.

– **Model Training**

We trained the models for a different number of epochs, depending on the model and dataset used. Since we use transfer learning, we need fewer epochs compared to training from scratch. In some models, we used an early stopping technique to prevent overfitting and saved the best model based on validation performance.

– **Model Evaluation**

We tested the final models on separate test datasets that are completely different from those used for training, visualized training and validation loss and accuracy curves to understand the model's behavior during training and detect issues like overfitting.

C. Results Obtained

The training and testing accuracies we obtained after training the models on the two datasets are shown in Table 2, where:

Training Accuracy is the accuracy of a convolutional neural network (CNN) model on the training dataset. It measures how well the model predicts the correct output for examples it was trained on. The accuracy equation is defined by:

$$\text{Accuracy} = \frac{\text{Number of correctly classified samples}}{\text{Total number of samples}} \times 100\%$$

Testing Accuracy is the accuracy of a CNN model on a separate testing dataset, which was not seen by the model during training. It evaluates the generalization ability of the model, indicating how well it performs on unseen data.

		Training accuracy	Testing accuracy
AID Dataset	Our architecture	90%	86%
	VGG16	95%	96%
	DenseNet12	98%	99%
	ResNet50	76%	74%
RSSCN7 Dataset	Our architecture	91%	89%
	VGG16	94%	90%
	DenseNet12	91%	90%
	ResNet50	71%	66%

Table 2: Training and Testing accuracy for Individual datasets.

The training and testing results reveal that both the VGG16 and DenseNet121 models exhibited the highest accuracy across both the AID and RSSCN7 datasets. Specifically, VGG16 achieved a remarkable 95% accuracy on the AID dataset and 94% on the RSSCN7 dataset. In comparison, DenseNet121 demonstrated superior performance with an accuracy of 98% on the AID dataset and 91% on the RSSCN7 dataset.

Following these, the CNN model trained from scratch performed moderately well, achieving an accuracy of 86% and 89% on the AID and RSSCN7 datasets respectively. Meanwhile, ResNet50 demonstrated the lowest accuracy on both datasets.

Step 2: Combined Dataset Analysis

In the second step, we combine the two datasets into a single, larger dataset to increase the number of training data. This combination of the datasets provides several benefits:

Firstly, it enhances generalization. With a more extensive and diverse training dataset, CNN can learn a better representation of the underlying patterns in the data. This, in turn, improves the model's capacity to generalize to unseen examples, resulting in better performance on real-world data.

Secondly, it reduces overfitting. As the number of data increases, the risk of overfitting decreases. Overfitting occurs when a model memorizes the training data rather than learning from it. A larger dataset helps alleviate this issue by exposing the model to a wider array of scenarios.

Thirdly, it boosts robustness. A CNN trained on a larger dataset is likely to exhibit greater resilience to variations and distortions in the input data.

In our combined dataset, we have 2520 images divided equally into the following categories.

Scene Categories	Training set	Testing set	Validation set
Farmland (Field)	520	55	55
Forest	520	55	55
Meadow (Grass)	520	55	55
Others	520	55	55

Table 3: Number of trainings, testing and validation images in each category.

First, we trained the same CNN architecture shown in **Figure 30** on this combined dataset. Next, we applied the same methodology illustrated in **Figure 31** to adapt the pre-trained VGG16, DenseNet121, and ResNet50 models to this dataset as well, in order to compare the results and choose the most accurate model for our task.

Achieved Results

The training and testing accuracies obtained after training the models on the combined dataset are summarized in Table 4.

Architecture used	Training Accuracy	Testing Accuracy
Our architecture	92%	89%
VGG16	96%	95%
DenseNet121	94%	94%
ResNet50	62%	56%

Table 4: Training and testing accuracy on the combined dataset.

During the training phase of each model, we monitored accuracy and loss on both the training and validation datasets, to determine if the model is converging or if there are signs of overfitting, facilitating adjustments to hyperparameters or the model architecture as needed for optimal performance. As shown in Figure 32 and Figure 33 for the VGG16 experiment.

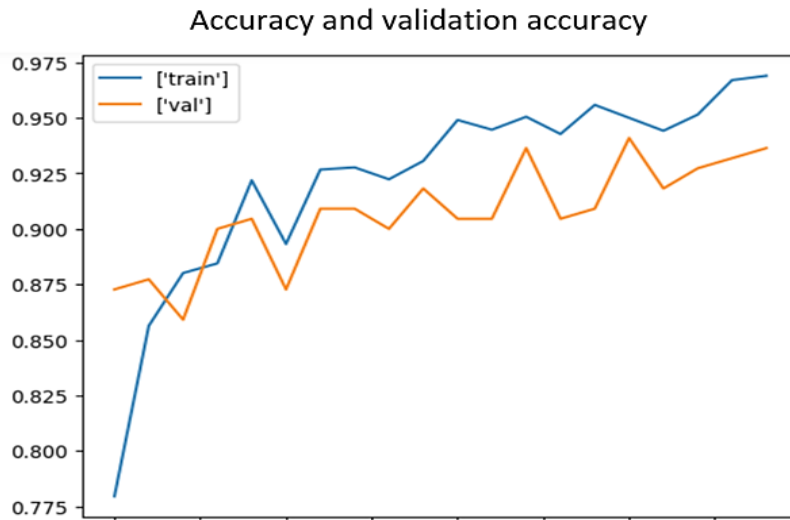


Figure 32: VGG16 Training and Validation Accuracy.

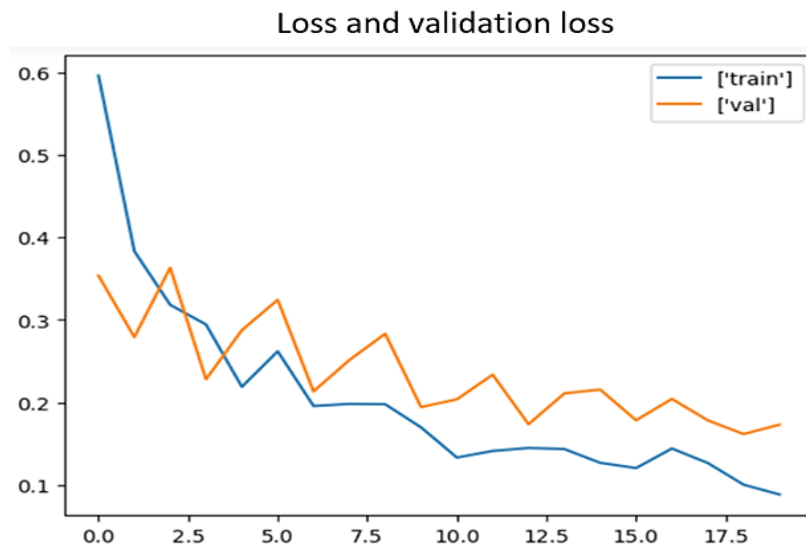


Figure 33: VGG16 Training and Validation Loss.

- **Training Accuracy Curve:** represented in blue in Figure 32. This curve shows how accurately the model predicts the training data during each epoch. It started low and then increased as the model learned from the data.
- **Training Loss Curve:** represented in blue in Figure 33. This curve shows the value of the loss function (a measure of the error between predictions and true labels) on the

training data during each epoch. At first, the loss was high because the model's predictions were far from the truth. As training progressed, the loss generally decreased as the model improved its predictions.

- **Validation accuracy curve:** represented in orange in Figure 32. This curve shows how accurately the model predicts the validation data over each epoch. It gives an indication of how well the model generalizes to new, unseen examples. The validation accuracy followed a similar trend as the training accuracy, it gradually increasing over time.
- **Validation Loss Curve:** represented in orange in Figure 33. Similar to the training loss curve, this curve shows the value of the loss function on the validation data during each epoch. It provides insight into how well the model performs on unseen data. The gradual decrease in validation loss indicates that the model is improving its generalization performance.

3.3.4. Discussion

The results obtained provide valuable insights into the performance and suitability of the four models we used in our task of classifying aerial images into four categories: farmland, meadow, forest, and others. We observed varying levels of performance among different models across both individual datasets and the combined dataset.

While our CNN architecture demonstrated reasonable performance, the pre-trained VGG16 and DenseNet121 models consistently outperformed it.

VGG16 consistently emerged as the top-performing model for our task. It is a powerful pre-trained model known for its good performance on various computer vision tasks. By leveraging the knowledge learned from large datasets during pre-training, VGG16 is able to generalize well to our specific task.

DenseNet121 also exhibited strong performance, indicating the effectiveness of its densely connected layers in extracting features from our datasets. However, both VGG16 and DenseNet121 models performed better on the AID dataset, where its performance decreased on the RSSCN7 dataset, and then increased again after merging the two datasets. This is because the RSSCN7 dataset contains slightly more diverse and complex patterns than the AID dataset.

ResNet50, showed comparatively lower performance compared to the other architectures. We conducted numerous experiments aiming to enhance its performance on our dataset.

These experiments included adjusting the learning rate, adding dropout and dense layers, and experimenting with various optimization algorithms such as Adam, SGD, and RMSprop. However, the results persisted in being unsatisfactory. Additionally, we employed the fine-tuning technique on the pre-trained ResNet50 model with our dataset. This involves updating the weights of this model on our specific data while leveraging the knowledge learned from the original dataset it was pretrained on, but this led to overfitting issues. This indicates that ResNet50 may not be suitable for our specific task and our datasets. This poor performance for ResNet50 on our datasets may be due to several reasons, including:

1. Dataset Characteristics

ResNet50 might not be well-suited for the specific characteristics of our datasets. It's possible that the features learned by ResNet50 do not align well with the nuances present in our aerial images, leading to suboptimal performance.

2. Model Complexity

ResNet50 is a relatively deep and complex model compared to VGG16 and DenseNet121. The increased complexity might lead to challenges in training, such as vanishing gradients or optimization difficulties, especially with smaller datasets like ours.

Overall, the pretrained VGG16 model adapted to our dataset (a combination of AID and RSSCN7 dataset), which achieved a training accuracy of 96% and a testing accuracy of 95%, is the most suitable model for classifying aerial images into the following classes: farmland (field), meadow (grass), forest, and others.

3.4. Geospatial Data Processing and Analysis

This section summarizes the methodology used to create, manage, and analyze spatial data in QGIS for identifying and classifying land types within the Assafia region in Laghouat state, Algeria illustrated in Figure 34. We developed a custom QGIS plugin to classify geotagged images based on a pre-trained CNN model. The spatial data was managed using PostgreSQL with PostGIS, ensuring secure storage and easy accessibility. We integrated QGIS with PostgreSQL to export the classified layers and published the spatial data via GeoServer, allowing for enhanced visualization and interaction.

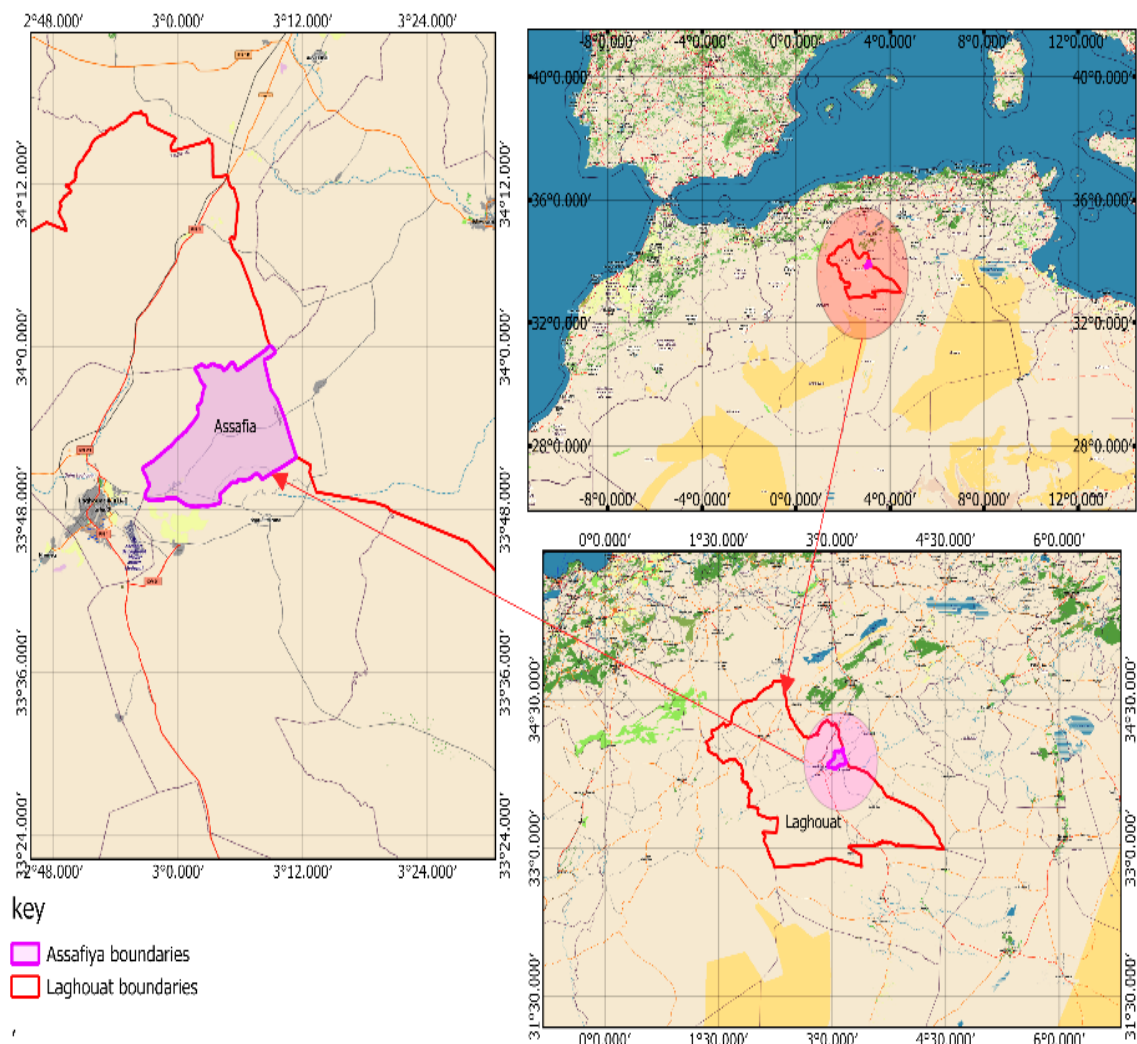


Figure 34: The Assafia Region Localization on The Map

3.4.1. Project Initialization and Data Preparation

This section describes the initial steps for setting up the GIS project and collecting the required data.

A. Creating the QGIS Project

A new QGIS project was initiated to manage and analyze the spatial data specific to the Assafia region. This project serves as the framework for subsequent data layers and analyses.

B. Adding the Google Satellite Layer

To provide a high-resolution basemap for visualizing the spatial context of the Assafia region, a Google Satellite layer was integrated into the project. The Google Satellite layer was subsequently added to the project, offering detailed satellite imagery as a backdrop for further spatial analysis.

C. Adding and Filtering the Border Layer

A vector layer of polygon type was added to the project as shown in Figure 35. This layer utilized a shapefile sourced from the GADM database, which represents all of Algeria's subnational administrative boundaries. To focus the analysis specifically on the targeted region, a filter was applied to the layer to display only the polygons corresponding to the Assafia region.

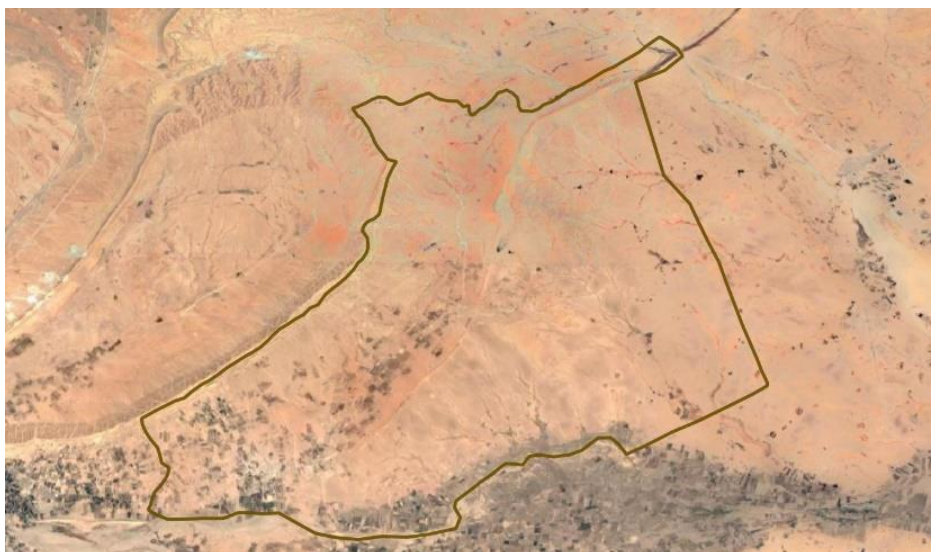


Figure 35: Assafia Border layer.

D. Adding the Images Layer

A vector layer of point type was added to the project. This layer is intended to store geotagged images which, after processing through the CNN recognition model, are used to classify land types.

E. Dataset Description

The dataset comprised 47 geotagged satellite images shown in Figure 36 captured from various locations within the Assafia region, each containing GPS information (latitude and longitude).

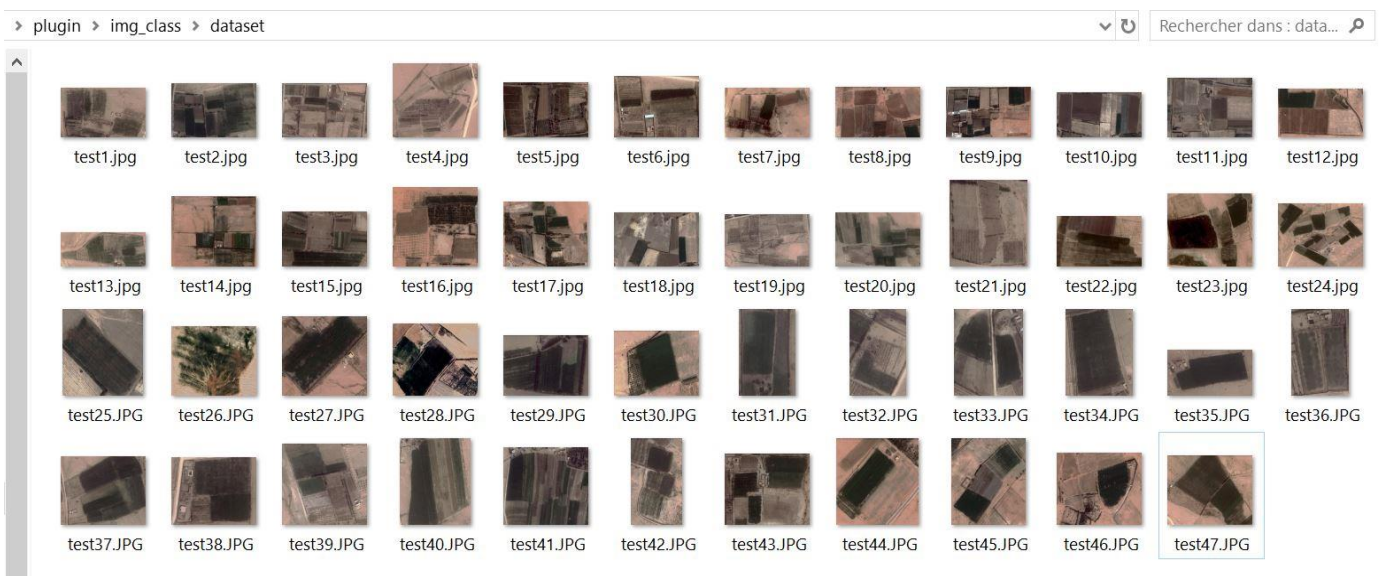


Figure 36: Geotagged Images Dataset.

3.4.2. Plugin Development

A. Creating the Plugin

To facilitate the classification of geotagged images, we developed a custom QGIS plugin. This plugin is designed to classify (label) a given image by land type, leveraging a pre-trained CNN model. It streamlines the process of importing geotagged images, applying the classification model, and visualizing the results directly within QGIS. This integration allows users to efficiently manage and analyze spatial data, enhancing their ability to identify and monitor different land types, with a particular focus on agricultural areas.

B. Designing the Plugin Interface

The plugin's interface (base.ui) demonstrated in Figure 37 was designed using Qt Designer.

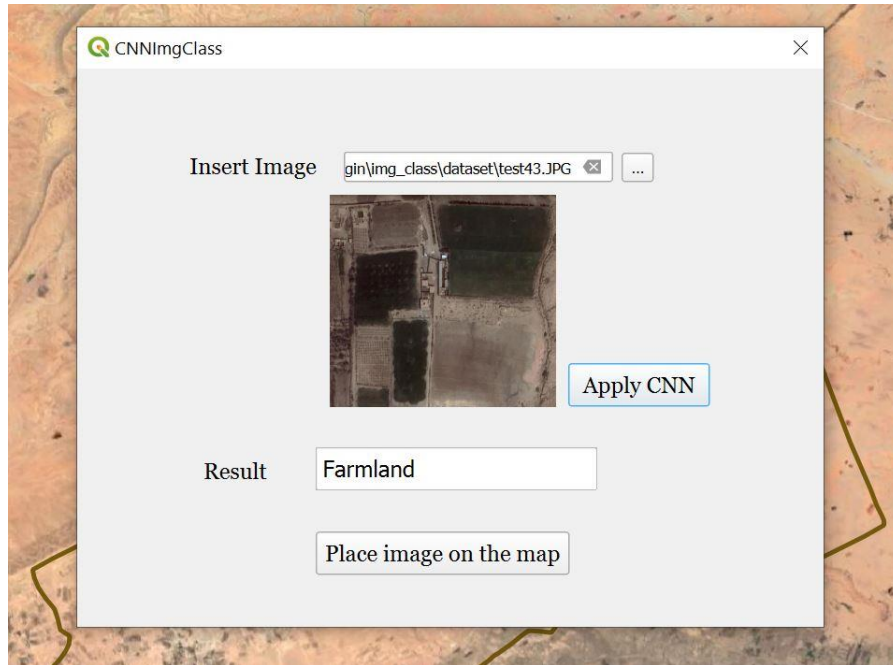


Figure 37: The Custom Plugin interface.

The interface includes:

- **A Widget** on top for loading and displaying the geotagged image.
- **An "Apply CNN" button** that, when clicked, applies the VGG16 model to the loaded image.
- **A Result Field** displaying the classification result.
- **A "Place on the Map" button** that, when clicked, inserts the classified image along with its metadata into the "images" layer on the map, visually representing its location and classification result.

3.4.3. Classification and Analysis

We explain the process of classifying and analyzing the data, visualising and interpreting the results.

A. Applying the CNN Model and Storing information

Upon loading an image into the plugin, the "Apply CNN" button triggers the VGG16 model to classify the image.

The classification results, along with other image information (name, file type, longitude, latitude, etc.), were added to the attributes table depicted in Figure 38 of the images layer.

fid	Name	Land type	Actions	File type	Latitude	Longitude
1	test1.jpg	Farmland	Vie...age	image/jpeg	33.82430156000278	3.0005975599967893
2	test2.jpg	Farmland	Vie...age	image/jpeg	33.81838688999173	2.9919912
3	test3.jpg	Farmland	Vie...age	image/jpeg	33.84156343005952	2.998063600019502
4	test4.jpg	Farmland	Vie...age	image/jpeg	33.846632590003004	2.999894539955428

Figure 38: Preview of the attributes table.

B. Visualizing Results

Each image was represented as a red rectangle on the QGIS map, highlighting its location within the Assafia region. The "Place on the Map" button was used to insert the classified images into the map, providing a visual representation of the areas identified by the CNN model as illustrated in Figure 39. The results of the classification, which identified the land type, are also displayed as labels on the map.



Figure 39: Visualizing Geotagged Images of Assafia on the Map.

For this study, all 47 images were correctly classified, and each one resulted in the land type "Farmland." This consistent classification can be attributed to our specific focus on farmland identification, which guided both the selection of our dataset and the training of our model. Consequently, the lack of diversity in the classification results aligns with the study's concentrated objective of detecting and analysing farmland areas.

3.4.4. Connecting QGIS to PostgreSQL and Exporting Data

We detail the process of connecting QGIS to the PostgreSQL database and exporting data. This includes configuring the database connection in QGIS and exporting spatial data layers to PostgreSQL.

A. Setting up PostGIS and establishing a Geodatabase

To enable efficient storage and management of spatial data within PostgreSQL, we installed the PostGIS extension. This extension adds geospatial capabilities to PostgreSQL, allowing it to handle spatial data effectively. Additionally, we created a new PostgreSQL database and integrated the PostGIS extension into it. This configuration equips the database with essential functionalities required for geospatial operations.

B. Creating a Connection in QGIS

Utilizing the connectivity features in QGIS, we established a direct link to the PostgreSQL database. This connection acts as a conduit between QGIS and the PostgreSQL database, enabling smooth data exchange.

C. Exporting spatial layers to PostgreSQL

We exported the border and images layers from QGIS to the PostgreSQL database. This process ensures that all spatial data, including both table attributes and geometry, is stored within the PostgreSQL database. By consolidating spatial data in PostgreSQL, we ensure robust data management capabilities and facilitate further geospatial analysis. This streamlined approach enhances data organization and accessibility, paving the way for efficient spatial data management and analysis workflows.

3.4.5. Publishing Data in GeoServer

We describe the procedure for publishing layers exported from PostgreSQL in GeoServer. This includes publishing the data and viewing it in OpenLayers and Google Earth Pro.

A. Workspace and Data Store Setup

In GeoServer, the spatial data stored in the PostgreSQL database was organized and made accessible as follows:

1. Workspace creation

We established a new workspace to categorize and manage the spatial data effectively.

2. Data Store addition

A new data store was integrated, linking directly to the PostgreSQL Geodatabase.

This entailed specifying connection parameters to ensure seamless access to the spatial data.

B. Publishing Layers

The border and images layers have been successfully incorporated into GeoServer, enriching their availability for visualization and interaction. This integration process involved adding the layers within the GeoServer environment, allowing users seamless access to the spatial data. Additionally, meticulous configuration was performed on each layer to ensure accurate representation and enhanced usability within GeoServer.

C. Aggregate Layer Creation

To provide a comprehensive view, an aggregate layer shown in Figure 40 was generated by combining the border and images layers. This integrated layer offers a unified perspective, encompassing both administrative boundaries and labeled geotagged images.

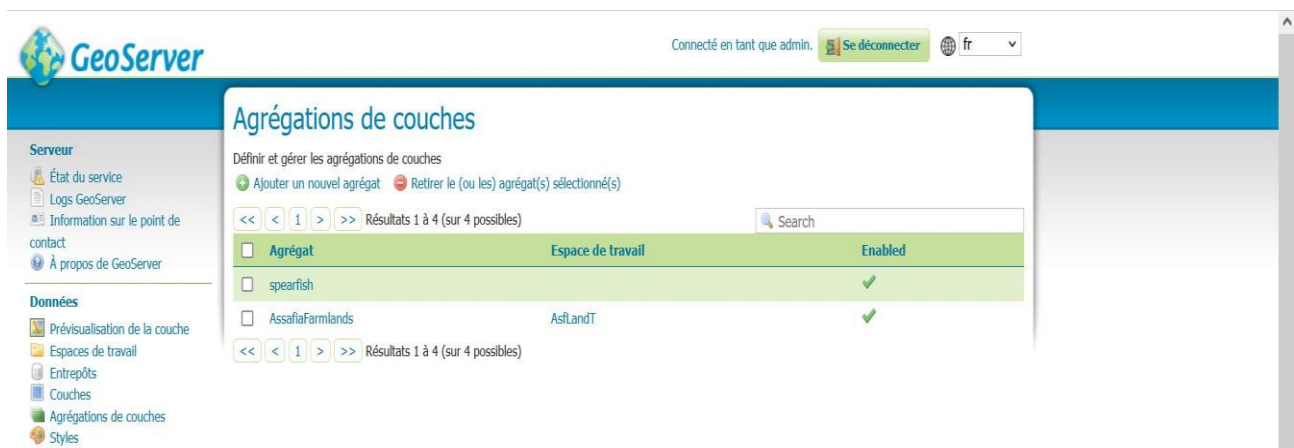


Figure 40: Aggregate Layer.

D. Viewing and Exporting the Data

In the "Layer Preview" section of GeoServer, the aggregate layer underwent visual confirmation through OpenLayers. As illustrated in Figure 41, ensuring the effective integration of the spatial data.

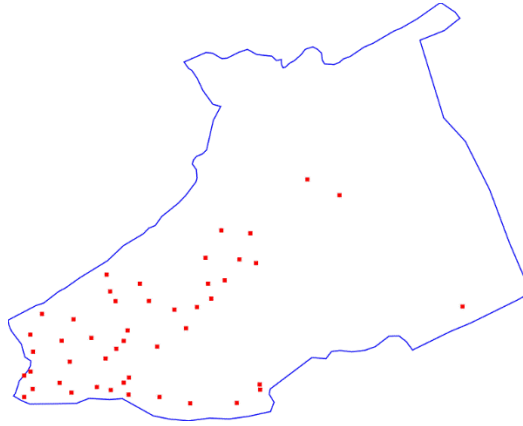


Figure 41: The aggregate layer view in OpenLayers.

Furthermore, the layer was exported as a .kml file, affording users the opportunity for additional validation and exploration within Google Earth Pro as depicted in Figure 42. This process facilitated a more comprehensive understanding of the spatial data by enabling immersive 3D visualization and interaction, contributing to the rigor and depth of the analysis.

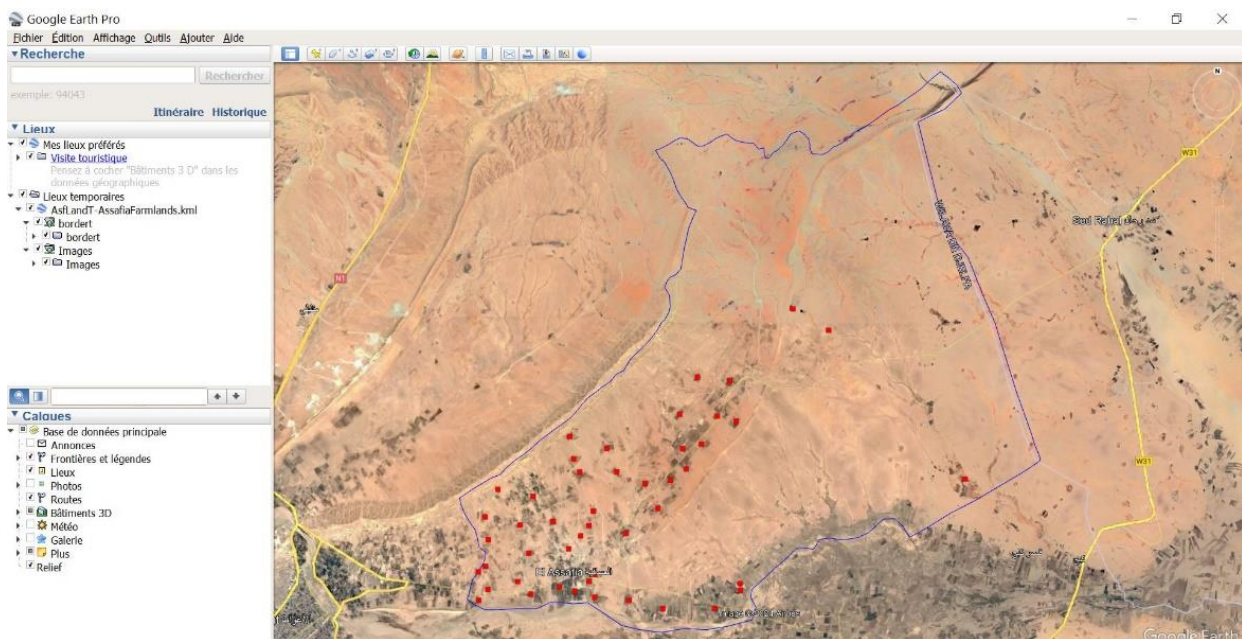


Figure 42 : The aggregate layer view in Google Earth Pro.

3.4.6. Observations

The methodology described facilitated the efficient classification and visualization of geotagged images within the Assafia region. All 47 images were successfully classified as "Farmland" and accurately mapped in the QGIS project.

This process demonstrated the applicability of the CNN model for land type classification and also highlighted the utility of custom QGIS plugins in spatial analysis. The integration of the layers and the developed plugin into QGIS formed a comprehensive desktop application, enhancing the ability to manage and analyze geospatial data within a single environment.

The connection to the PostgreSQL database ensured secure storage and easy accessibility for further analysis and reporting. Moreover, the publication of the spatial data in GeoServer and its visualization in Google Earth Pro provided a robust platform for disseminating and interacting with the spatial information.

3.5. Conclusion

This chapter has presented the methodology employed for classifying aerial images and the results obtained from applying various deep learning models. The analysis revealed that pre-trained models, particularly VGG16, achieved superior performance compared to our custom CNN architecture and ResNet50 model. VGG16's strong performance can be attributed to its ability to leverage knowledge from large datasets and generalize effectively to our specific task. While ResNet50 underperformed, further investigation into its limitations in the context of our data is recommended.

Furthermore, the chapter demonstrated the successful application of the chosen methodology for a real-world scenario - classifying land types in the Assafia region. All images were accurately classified and mapped, showcasing the effectiveness of CNNs for land cover classification and the value of custom QGIS plugins for spatial analysis. The integration of these elements resulted in a comprehensive desktop application for managing and analyzing geospatial data.

The secure storage in a PostgreSQL database and publication in GeoServer and Google Earth Pro further enhance data accessibility and dissemination. In essence, this chapter highlights the importance of selecting appropriate models for specific tasks and demonstrates the efficacy of the developed methodology for land cover classification and spatial data management.

GENERAL CONCLUSION

In conclusion, our master thesis has successfully showcased the integration of Deep Learning, particularly Convolutional Neural Networks (CNNs), with Geographic Information Systems (GIS) to accurately identify and map agricultural regions using aerial imagery. After providing a brief overview of the recent advancements in utilizing Convolutional Neural Networks (CNNs) for classifying field types and land cover through aerial imagery, as well as the application of Geographic Information Systems (GIS) for generating spatial maps and visualizing geographic data, we proceeded to implement our proposed technique that integrates these two powerful tools. Moreover, our study involved the use of a dataset consisting of 2520 scene images collected from various remote imaging sensors. To ensure the quality of the data, appropriate preprocessing techniques were applied. Subsequently, a CNN model was built from scratch and trained on the preprocessed dataset. Additionally, three pretrained models - VGG16, DenseNet121, and ResNet50 - were adapted on the same dataset to determine the most accurate one for our task. The evaluation revealed that the VGG16 model outperformed all other models used, including the CNN model built from scratch, achieving an impressive accuracy of 96%.

Our work highlights the remarkable effectiveness of the Transfer Learning approach, specifically the VGG16 model, even with limited data, in identifying and categorizing diverse landscapes, such as farmland, forests, meadows, and others. In the next phase of this work, we utilized Geographic Information System (GIS) techniques to develop a desktop application for the efficient classification and visualization of geotagged images. Initially, we acquired images along with their geographical coordinates from the Assafia region in the state of Laghouat, Algeria, using Google Earth. Subsequently, a custom Plugin was created using GIS tools. This Plugin takes a geotagged image as input, utilizes a Convolutional Neural Network (CNN) model to assign a category to the input image, and accurately places it on the map along with all its metadata, including land type.

Our proposed future work includes focusing on classifying the types of implantations within agricultural lands rather than simply categorizing them as farmland. This will involve acquiring a specialized dataset with detailed annotations of various types of implantations. Additionally, we plan to expand the GIS desktop application to include tools for predictive modeling, trend analysis, and decision support systems for agricultural planning. We also aim to develop web-based and mobile versions of the GIS application to increase accessibility for farmers, researchers, and policymakers.

BIBLIOGRAPHY

- [1] J. Liu, L. Shu, X. Lu, and Y. Liu, 'Survey of Intelligent Agricultural IoT Based on 5G', *Electronics*, vol. 12, no. 10, p. 2336, May 2023, doi: 10.3390/electronics12102336.
- [2] B. A. Stout and B. Cheze, 'CIGR Handbook of Agricultural Engineering, Volume III Plant Production Engineering, Front Matter and Index', in *CIGR Handbook of Agricultural Engineering Volume III Plant Production Engineering*, American Society of Agricultural and Biological Engineers, 1999. doi: 10.13031/2013.36362.
- [3] I. Kovács and I. Husti, 'The role of digitalization in the agricultural 4.0 – how to connect the industry 4.0 to agriculture?', *Hung. Agric. Eng.*, no. 33, pp. 38–42, 2018, doi: 10.17676/HAE.2018.33.38.
- [4] R. I. Iorgulescu, 'Aspects of the transition to a digital bioeconomic society: Agriculture 4.0 and Organic 3.0', *Inst. Econ. Forecast.*, p. 6, 2022.
- [5] O. Bongomin *et al.*, 'Industry 4.0 Disruption and Its Neologisms in Major Industrial Sectors: A State of the Art', *J. Eng.*, vol. 2020, pp. 1–45, Oct. 2020, doi: 10.1155/2020/8090521.
- [6] P. B. Shirsath, P. K. Aggarwal, P. K. Thornton, and A. Dunnett, 'Prioritizing climate-smart agricultural land use options at a regional scale', *Agric. Syst.*, vol. 151, pp. 174–183, Feb. 2017, doi: 10.1016/j.agsy.2016.09.018.
- [7] Y. Liu, X. Ma, L. Shu, G. P. Hancke, and A. M. Abu-Mahfouz, 'From Industry 4.0 to Agriculture 4.0: Current Status, Enabling Technologies, and Research Challenges', *IEEE Trans. Ind. Inform.*, vol. 17, no. 6, pp. 4322–4334, Jun. 2021, doi: 10.1109/TII.2020.3003910.
- [8] A. Marucci, A. Colantoni, I. Zambon, and G. Egidi, 'Precision Farming in Hilly Areas: The Use of Network RTK in GNSS Technology', *Agriculture*, vol. 7, no. 7, p. 60, Jul. 2017, doi: 10.3390/agriculture7070060.
- [9] F. Yandun Narvaez, G. Reina, M. Torres-Torriti, G. Kantor, and F. A. Cheein, 'A Survey of Ranging and Imaging Techniques for Precision Agriculture Phenotyping', *IEEEASME Trans. Mechatron.*, vol. 22, no. 6, pp. 2428–2439, Dec. 2017, doi: 10.1109/TMECH.2017.2760866.
- [10] J. G. Araujo Schmidt, I. R. D. Alcantara, C. E. D. F. Vian, and G. Belardo, 'AGRICULTURE 4.0: Origin, features, and consequences in the world and Brazil', *Quaestum*, vol. 2, pp. 1–14, Dec. 2021, doi: 10.22167/2675-441X-20210564.
- [11] S. O. Araújo, R. S. Peres, J. Barata, F. Lidon, and J. C. Ramalho, 'Characterising the Agriculture 4.0 Landscape—Emerging Trends, Challenges and Opportunities', *Agronomy*, vol. 11, no. 4, Art. no. 4, Apr. 2021, doi: 10.3390/agronomy11040667.
- [12] 'Types Of Crops: How To Classify And Manage Different Plants'. Accessed: Jun. 13, 2024. [Online]. Available: <https://eos.com/blog/types-of-crops/>
- [13] P. O. Ogunbamowo, O. Olaniyi M. B., and O. S. Ariwoola, 'Digital Innovations for Sustainable Forest Management: A Review of Strategies and Applications in the Nigerian Context.', presented at the e-Proceedings of the Faculty of Agriculture International Conference, 2024, pp. 124–129.
- [14] 'Projects Archives - Page 4 of 4 - DHI'. Accessed: Jun. 13, 2024. [Online]. Available: <https://eo.dhigroup.com/category/projects/page/4/>
- [15] 'Agriculture with Remote Sensing Technology - Africa Agricultural Network'. Accessed: Jun. 13, 2024. [Online]. Available: <https://africaagriculturalnetwork.com/the-best-ways-consumer-trends-impact/>
- [16] Ershad Ali, 'Geographic Information System (GIS) : Definition, Development, Applications & Components | Download Free PDF | Spatial Analysis | Geographic Information System', Scribd. Accessed: May 30, 2024. [Online]. Available: <https://www.scribd.com/document/511975962/GeographicInformationSystemGISDefinitionDevelopmentApplicationsComponents>
- [17] Z. Pajorska, 'What is GIS: a Complete Guide to Geographic Information Systems', Stratoflow. Accessed: May 29, 2024. [Online]. Available: <https://stratoflow.com/what-is-gis/>
- [18] F. Pucha-Cofrep, A. Fries, F. Cánovas-García, F. Oñate-Valdivieso, V. González-Jaramillo, and D. Pucha-Cofrep, *Fundamentals of GIS: Applications with ArcGIS*. Franz Pucha Cofrep, 2018.
- [19] Russell G. Congalton and Kass Green, 'The ABCs of GIS'.
- [20] E. Zhovinsky *et al.*, 'Water Quality Improvements through Fluoride Reduction in Groundwater of Central Europe', *Final Rep. Water Qual. Improv. Fluoride Reduct. Groundw. Cent. Eur.*, p. 292, 2001.
- [21] F. Escobar, G. Hunte, I. Bishop, and A. Zerger, *Introduction to GIS*. 2008.
- [22] United Nations, Ed., *Handbook on geographic information systems and digital mapping*. in Studies in methods, no. no. 79. New York: United Nations, 2000.
- [23] N. Earth Science Data Systems, 'Geographic Information Systems Data Pathfinder | Earthdata'. Accessed: Jun. 13, 2024. [Online]. Available: <https://www.earthdata.nasa.gov/learn/pathfinders/gis-pathfinder>
- [24] J. D. Kelleher, *Deep Learning*. MIT Press, 2019.
- [25] P. P. Shinde and S. Shah, 'A Review of Machine Learning and Deep Learning Applications', in *2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBEA)*, Pune, India: IEEE, Aug. 2018, pp. 1–6. doi: 10.1109/ICCUBEA.2018.8697857.
- [26] S. Iliopoulou, 'Deep Learning Based Image Compression', 2022.
- [27] P. Baheti, 'The Essential Guide to Neural Network Architectures'. Accessed: Jun. 13, 2024. [Online]. Available: <https://www.v7labs.com/blog/neural-network-architectures-guide>
- [28] I. N. da Silva, D. H. Spatti, R. A. Flauzino, L. H. Bartocci Liboni, and S. F. dos Reis Alves, 'Introduction', in *Artificial Neural Networks*, in Springer International Publishing Switzerland. , 2017. Accessed: Jun. 13, 2024. [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-319-43162-8_1

- [29] Y. Pandey, 'NEURAL NETWORKS'. Accessed: Jun. 13, 2024. [Online]. Available: <https://www.linkedin.com/pulse/neural-networks-yuvaraj-pandey>
- [30] O. Duerr, B. Sick, and E. Murina, *Probabilistic Deep Learning: With Python, Keras and TensorFlow Probability*. Simon and Schuster, 2020.
- [31] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, 'Gradient-based learning applied to document recognition', *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998, doi: 10.1109/5.726791.
- [32] M. Reza Keyvanpour and M. B. Shirzad, 'Chapter 14 - Machine learning techniques for agricultural image recognition', in *Application of Machine Learning in Agriculture*, M. A. Khan, R. Khan, and M. A. Ansari, Eds., Academic Press, 2022, pp. 283–305. doi: 10.1016/B978-0-323-90550-3.00011-4.
- [33] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. in Adaptive computation and machine learning. Cambridge, Mass: The MIT press, 2016.
- [34] I. Zafar, G. Tzanidou, R. Burton, N. Patel, and L. Araujo, *Hands-On Convolutional Neural Networks with TensorFlow: Solve computer vision problems with modeling in TensorFlow and Python*. Packt Publishing Ltd, 2018.
- [35] G. Rivera, A. Rosete, B. Dorronsoro, and N. Rangel-Valdez, *Innovations in Machine and Deep Learning: Case Studies and Applications | SpringerLink*, Springer Nature Switzerland., Studies in Big Data (SBD, volume 134) vols. 2023. Accessed: Jun. 13, 2024. [Online]. Available: <https://link.springer.com/book/10.1007/978-3-031-40688-1>
- [36] D. Raval and J. N. Undavia, 'A Comprehensive assessment of Convolutional Neural Networks for skin and oral cancer detection using medical images', *Healthc. Anal.*, vol. 3, p. 100199, Nov. 2023, doi: 10.1016/j.health.2023.100199.
- [37] 'Fully Connected Layers in Convolutional Neural Networks', IndianTechWarrior. Accessed: Jun. 13, 2024. [Online]. Available: <https://indiantechwarrior.com/fully-connected-layers-in-convolutional-neural-networks/>
- [38] 'Convolutional Neural Network — Lesson 13: Applications of CNNs', Medium. Accessed: May 30, 2024. [Online]. Available: <https://medium.com/@nerdjock/convolutional-neural-network-lesson-13-applications-of-cnns-7cd7ec7c04a>
- [39] R. Ribani and M. Marengoni, 'A Survey of Transfer Learning for Convolutional Neural Networks', Oct. 2019, pp. 47–57. doi: 10.1109/SIBGRAPI-T.2019.00010.
- [40] U. Maheswari, 'DL-Transfer learning.', Coursehero.com. Accessed: May 30, 2024. [Online]. Available: <https://www.coursehero.com/file/103722529/DL-Transfer-learningpptx/>
- [41] M. Hussain, J. Bird, and D. Faria, 'A Study on CNN Transfer Learning for Image Classification', Jun. 2018.
- [42] A. Sufian, A. Ghosh, A. Sadiq, and F. Smarandache, 'A Survey on Deep Transfer Learning and Edge Computing for Mitigating the COVID-19 Pandemic', *J. Syst. Archit.*, vol. 108, Aug. 2020, doi: 10.1016/j.sysarc.2020.101830.
- [43] S. Bangar, 'VGG-Net Architecture Explained', Medium. Accessed: May 30, 2024. [Online]. Available: <https://medium.com/@siddheshb008/vgg-net-architecture-explained-71179310050f>
- [44] I. Ali, M. Muzammil, Dr. I. Ulhaq, A. Khaliq, and S. Malik, 'Deep Feature Selection and Decision Level Fusion for Lungs Nodule Classification', *IEEE Access*, vol. 9, pp. 18962–18973, Jan. 2021, doi: 10.1109/ACCESS.2021.3054735.
- [45] P. Dwivedi, 'Understanding and Coding a ResNet in Keras', Medium. Accessed: May 30, 2024. [Online]. Available: <https://towardsdatascience.com/understanding-and-coding-a-resnet-in-keras-446d7ff84d33>
- [46] R. Modi, 'ResNet — Understand and Implement from scratch | by Rohit Modi | Analytics Vidhya | Medium', Medium. Accessed: Jun. 13, 2024. [Online]. Available: <https://medium.com/analytics-vidhya/resnet-understand-and-implement-from-scratch-d0eb9725e0db>
- [47] S. T. Mummadi, 'Unraveling the ResNet Architecture: A Deep Dive into Deep Learning', Medium. Accessed: May 30, 2024. [Online]. Available: <https://medium.com/@saitejamummadi/unraveling-the-resnet-architecture-a-deep-dive-into-deep-learning-72b757810d1e>
- [48] A. Ahmed, 'Architecture of DenseNet-121', OpenGenus IQ: Learn Algorithms, DL, System Design. Accessed: Jun. 13, 2024. [Online]. Available: <https://iq.opengenus.org/architecture-of-densenet121/>
- [49] 'Architecture of DenseNet-121', OpenGenus IQ: Learn Algorithms, DL, System Design. Accessed: May 30, 2024. [Online]. Available: <https://iq.opengenus.org/architecture-of-densenet121/>
- [50] A. Bouguettaya, H. Zarzour, A. Kechida, and A. M. Taberkit, 'Deep learning techniques to classify agricultural crops through UAV imagery: a review', *Neural Comput. Appl.*, vol. 34, no. 12, pp. 9511–9536, Jun. 2022, doi: 10.1007/s00521-022-07104-9.
- [51] A. Kamilaris and F. X. Prenafeta-Boldú, 'A review of the use of convolutional neural networks in agriculture', *J. Agric. Sci.*, vol. 156, no. 3, pp. 312–322, Apr. 2018, doi: 10.1017/S0021859618000436.
- [52] I. Teixeira, R. Morais, J. J. Sousa, and A. Cunha, 'Deep Learning Models for the Classification of Crops in Aerial Imagery: A Review', *Agriculture*, vol. 13, no. 5, Art. no. 5, May 2023, doi: 10.3390/agriculture13050965.
- [53] O. G. Ajayi and O. O. Olufade, 'DRONE-BASED CROP TYPE IDENTIFICATION WITH CONVOLUTIONAL NEURAL NETWORKS: AN EVALUATION OF THE PERFORMANCE OF RESNET ARCHITECTURES', *ISPRS Ann. Photogramm. Remote Sens. Spat. Inf. Sci.*, vol. X-1-W1-2023, pp. 991–998, Dec. 2023, doi: 10.5194/isprs-annals-X-1-W1-2023-991-2023.
- [54] H. Wang, J. Feng, and H. Yin, 'Improved Method for Apple Fruit Target Detection Based on YOLOv5s', *Agriculture*, vol. 13, no. 11, Art. no. 11, Nov. 2023, doi: 10.3390/agriculture13112167.
- [55] R. Chew *et al.*, 'Deep Neural Networks and Transfer Learning for Food Crop Identification in UAV Images', *Drones*, vol. 4, no. 1, Art. no. 1, Mar. 2020.
- [56] A. Nowakowski *et al.*, 'Crop type mapping by using transfer learning', *Int. J. Appl. Earth Obs. Geoinformation*, vol. 98, p. 102313, Jun. 2021, doi: 10.1016/j.jag.2021.102313.

- [57] M. Mathenge, B. G. J. S. Sonneveld, and J. E. W. Broerse, 'Application of GIS in Agriculture in Promoting Evidence-Informed Decision Making for Improving Agriculture Sustainability: A Systematic Review', *Sustainability*, vol. 14, no. 16, Art. no. 16, Jan. 2022, doi: 10.3390/su14169974.
- [58] Biswas, Brototi and Mayank, Varun, 'Web GIS based identification and mapping of medicinal plants : A case study of Agra (U.P.), India', 2017.
- [59] Sonti SH, 'Application of Geographic Information System (GIS) in Forest Management', 2015.
- [60] F. M. C. D. Oliveira and E. Borin, 'Partitioning Convolutional Neural Networks to Maximize the Inference Rate on Constrained IoT Devices', *Future Internet*, vol. 11, no. 10, p. 209, Sep. 2019, doi: 10.3390/fi11100209.
- [61] N. Abd Mubin, 'Young and mature oil palm tree detection and counting using convolutional neural network deep learning method'. Accessed: May 30, 2024. [Online]. Available: <https://www.tandfonline.com/doi/abs/10.1080/01431161.2019.1569282>
- [62] E. C. Rodríguez-Garlito, A. Paz-Gallardo, and A. J. Plaza, 'Mapping the Accumulation of Invasive Aquatic Plants in the Guadiana River, Spain, Using Multitemporal Remote Sensing', *IEEE Geosci. Remote Sens. Lett.*, vol. 20, pp. 1–5, 2023, doi: 10.1109/LGRS.2023.3277366.
- [63] T. Admin, 'Why Python Programming', Prgramly. Accessed: Jun. 13, 2024. [Online]. Available: <https://www.prgramly.com/why-python-programming/>
- [64] 'What is Anaconda? | Domino Data Science Dictionary'. Accessed: May 30, 2024. [Online]. Available: <https://domino.ai/data-science-dictionary/anaconda>
- [65] 'Comment installer Anaconda sous Linux'. Accessed: Jun. 13, 2024. [Online]. Available: <https://geekflare.com/fr/install-anaconda-on-linux/>
- [66] 'How Does TensorFlow Work and Why is it Vital for AI? - Spiceworks', Spiceworks Inc. Accessed: May 30, 2024. [Online]. Available: <https://www.spiceworks.com/tech/devops/articles/what-is-tensorflow/>
- [67] 'Keras: The high-level API for TensorFlow | TensorFlow Core', TensorFlow. Accessed: May 30, 2024. [Online]. Available: <https://www.tensorflow.org/guide/keras>
- [68] 'What is NumPy? — NumPy v1.26 Manual'. Accessed: May 30, 2024. [Online]. Available: <https://numpy.org/doc/stable/user/whatisnumpy.html>
- [69] Melanie, 'Matplotlib: Master Data Visualization in Python', Data Science Courses | DataScientest. Accessed: May 30, 2024. [Online]. Available: <https://datascientest.com/en/matplotlib-master-data-visualization-in-python>
- [70] 'What is a kaggle? | Kaggle'. Accessed: May 30, 2024. [Online]. Available: <https://www.kaggle.com/discussions/general/a>
- [71] 'What Is PostgreSQL?' Accessed: May 13, 2024. [Online]. Available: <https://kinsta.com/knowledgebase/what-is-postgresql/>
- [72] 'PostGIS', PostGIS. Accessed: Jun. 13, 2024. [Online]. Available: <https://postgis.net/>
- [73] 'QGIS Project Logo', OSGeo. Accessed: Jun. 13, 2024. [Online]. Available: <https://www.osgeo.org/projects/qgis/qgis-logo/>
- [74] 'Qt Development Brand Guide: Logo'. Accessed: Jun. 13, 2024. [Online]. Available: <https://www.qt.io/brand/development/logo>
- [75] 'About - GeoServer'. Accessed: Jun. 13, 2024. [Online]. Available: <https://geoserver.org/about/>
- [76] 'Présentation – Google Earth', Google Earth. Accessed: Jun. 13, 2024. [Online]. Available: <https://www.google.com/intl/fr/earth/about/>
- [77] G.-S. Xia *et al.*, 'AID: A Benchmark Dataset for Performance Evaluation of Aerial Scene Classification', *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 7, pp. 3965–3981, Jul. 2017, doi: 10.1109/TGRS.2017.2685945.
- [78] 'AID: A scene classification dataset'. Accessed: May 30, 2024. [Online]. Available: <https://www.kaggle.com/datasets/jiayuanhengala/aid-scene-classification-datasets>
- [79] C. Shi, X. Zhao, and L. Wang, 'A Multi-Branch Feature Fusion Strategy Based on an Attention Mechanism for Remote Sensing Image Scene Classification', *Remote Sens.*, vol. 13, no. 10, p. 1950, May 2021, doi: 10.3390/rs13101950.
- [80] R. Cheng, H. Wang, and P. Luo, 'Remote sensing image super-resolution using multi-scale convolutional sparse coding network', *PLOS ONE*, vol. 17, no. 10, p. e0276648, Oct. 2022, doi: 10.1371/journal.pone.0276648.
- [81] 'RSSCN7'. Accessed: May 30, 2024. [Online]. Available: <https://www.kaggle.com/datasets/yangpeng1995/rsscn7>
- [82] I. den Bakker, *Python Deep Learning Cookbook*. Packt Publishing, 2017.