Democratic and Popular Republic of Algeria

Ministry of Higher Education and Scientific Research

University of Amar Telidji Laghouat

Faculty of Technology

Departement of Electronics

Master's Degree in Network And Telecommunications

Smart Vision System for Sheep Face Detection using Deep Neural Networks

MASTER DEGREE THESIS

Presented by
Lobna Chellali
Nermine Bouzaher

Supervisor
Dr.Mohammed Bouzidi
Dr.Kamal Sehairi

Jury members
President : Dr.Chaker Mohsen Naser Saleh
Examiner : Dr.Mourad Reguiegue

ACADEMIC YEAR 2023/2024
2024/06/20

بِسْمِ ٱللَّهِ ٱلرَّحْمَٰنِ ٱلرَّحِيمِ

الاهداء

الحمد لله على توفيقه وتيسيره، وأسأله أن يجعل هذا العمل خالصًا لوجهه الكريم، وأن ينفع به.
اللهم بارك لنا فيما رزقتنا، وزدنا من فضلك وبركتك، واجعله زاداً لنا في كل خير.
إلى من غمروني بالحب والعطاء، إلى أولئك الذين كان لهم الفضل الأكبر في تحقيق هذا الإنجاز.
إلى والدي العزيز(علال)، سندي، الذي كان ا دوماً مصدر إلهامي وقوتي ، وغمرني بعطفه ورعايته طوال مسيرتي التعليمية.
إلى والدتي جنتي (رقية)، التي كانت دومًا مثالًا للتضحية والحنان، والتي لم تدخر وسعًا في تقديم كل ما أحتاجه لتحقيق أحلامي.
إلى أخواتي الغاليات (مريم و هاجر و حنان)، اللاتي كنّ دومًا بجانبي، داعمات ومشجعات، ولم يتوانين عن تقديم يد العون والمساندة في كل لحظة.
إلى إخواني الأعزاء(ياسين و ابراهيم و عمار و محمد)، الذين كانوا دومًا سندًا لي.
إلى أحفاد العائلة (هيثم ,رقية ندى ,نور الهدى ,منار, محمد,هناء,سيراج الدين و نورسين)، الذين يملؤون حياتنا بالسعادة والسرور، خفظهم الله .
إلى عمّي بودالي و عمي عطاالله .
الى خالتي واخوالي وعماتي واعمامي الى كل افراد عائلتي وكل من ساندني.
إلى التوأم(بشير ونذير)، اللذين أضفيا على حياتنا البهجة والمرح.
الى صديقتي دربي و حبيبة قلبي (نرمين) من شاركتني جميع لحظاتي .
إلى صديقاتي العزيزات (اية و سمية و ام الخير) .
الى كل من علمني حرفا الى جميع اساتذتي.
أشكركم جميعًا من أعماق قلبي على دعمكم اللامحدود وتشجيعكم المستمر.

خريجتكم المتواضعة

لبنى شلالي

الاهداء

الحمد لله حبا وشكرا وامتنانا ، ماكنت لافعل هذا لولا فضل الله و توفيقه
فالحمدلله حتى يبلغ الحمد منتهاه
أهدي ثمرة سنيني الى وهج حياتي التي طالما حضنت اسمي بين اكف دعائها و قلبي بين احضان دفئها و حنانها ، الى التي افنت عمرها حتى نصل الى ما نحن عليه اليوم ، الى امرأتي المفضلة أمي .
أهدي هذا النجاح الى من تربيت علي يده ومن علمني القيم والمبادئ الى من كلل العرق جبينه وعلمني ان النجاح لا ياتي الا بصبر والاصرار
الى الرجل الاعظم في حياتي ابي .
الى سندي وعضدي الشديد وجبلي الذي لا يهتز ولا يميل اختي ايه واخي انس وعبد المؤمن .
الى من غادرونا مبكرا
جدي الغالي الحاج مصطفى الذي التمست جميل حظي في اني حفيدة لرجل ملهم و قوي مثله وخالي العزيز على قلبي رضوان الذي كان قدوه لي في صغري وكان بطلي الخارق آمل لو كانا هنا حتى يشهدا تخرجي ويفخرا بي(رحمهما الله).
الى رفقه الطيبه التي امنت به وشجعتني وكانت لي القمر الذي اضاء لي ليلي تقوى، ماجدة و امينة .
الى صديقتي الغاليه التي شاركتني ورده وشوك هذا المشوار والتي ضحكنا وبكينا وسقطنا ونهضنا معا الى نعم الرفيقه لبنى .
الى قوتي المحركة والتي طالما بثت في نفسي الامل والحب والفخر جدتي بدرة،
الى كل اصدقائي وعائلتي واساتذتي وكل من علمني حرفا وكان لهم الفضل علي .
جزاكم الله كل خير.

خريجتكم المتواضعة
نرمين بوزاهر

Acknowledgment

We are profoundly grateful to Allah the Almighty, who bestowed upon us health, strength, patience, and willpower throughout the journey of completing this thesis. His blessings and guidance have been the cornerstone of our perseverance and success.

WE also extend our heartfelt thanks to Dr. Kamel Sehairi and Dr. Mohammed Bouzidi for their invaluable guidance, support, and expertise. Their contributions have played a pivotal role in shaping this work into its final form.

We would also like to express our gratitude to our professors and all the teachers who have contributed to our training. Additionally, we extend our appreciation to all the members of the jury for honoring us by accepting to judge our thesis and for their help in reviewing and improving it.

Finally, we would like to express our heartfelt gratitude to our families and friends who have welcomed and supported us throughout our studies.

To those whom we have unintentionally forgotten and who participated in any way in the preparation of this work, find here the expression of our gratitude.

الملخص

تبحث هذه الأطروحة في تطبيق تقنيات التعلم العميق لاكتشاف وجوه الأغنام، مع التركيز على نموذجين بارزين لاكتشاف الكائنات: SSD MobileNet 320x320 و CenterNet Hourglass 512x512.

تهدف دراستنا إلى تحسين إدارة الماشية من خلال توفير طريقة فعالة لتحديد الأغنام الفردية داخل القطيع. من خلال الاستفادة من خوارزميات اكتشاف الكائنات المتقدمة، نُظهر تحسينات كبيرة في دقة وكفاءة تحديد الأغنام. تشير النتائج إلى أنه بينما يحقق نموذج CenterNet Hourglass 512x512 دقة أعلى، يوفر نموذج SSD MobileNet 320x320 أوقات معالجة أسرع. تساهم هذه الدراسة في مجال الذكاء الاصطناعي الزراعي، حيث تقدم حلولاً عملية لتحسين مراقبة وإدارة الماشية.

كلمات مفتاحية : التعلم العميق ، SSD MobileNet ، CenterNet Hourglass ، الذكاء الاصطناعي.

Abstract

This thesis investigates the application of deep learning techniques to sheep face detection, focusing on two prominent object detection models: SSD MobileNet 320x320 and CenterNet Hourglass 512x512. Our study aims to enhance livestock management by providing an efficient method for identifying individual sheep within a flock. By leveraging advanced object detection algorithms, we demonstrate significant improvements in the accuracy and efficiency of sheep identification. The results indicate that while CenterNet Hourglass 512x512 achieves higher accuracy, SSD MobileNet 320x320 offers faster processing times. This research contributes to the field of agricultural AI, offering practical solutions for enhanced livestock monitoring and management.

Keywords : Deep learning ,SSD MobileNet ,CenterNet Hourglass , AI

CONTENTS

General Introduction 1

1 Object Detection Techniques 3

1.1 Introduction 3
1.2 Artificial Intelligence 3
1.2.1 Artificial Intelligence application 4
1.2.2 Artificial Intelligence Categories 5
1.3 Machine Learning 5
1.3.1 How machine learning work 5
1.3.2 Types of machine learning 7
1.4 Traditional Machine Learning 13
1.4.1 Limitations of Traditional Machine Learning 14
1.4.2 Strengths of Traditional Machine Learning 14
1.4.3 Traditional machine learning techniques 15
1.5 Deep Learning 17
1.5.1 Artificial Neural Network 18
1.5.2 Deep neural network 18
1.5.3 How Deep Learning for Object Detection Works 19
1.5.4 Advantages of Deep Learning for Object Detection 20
1.5.5 Deep Learning Frameworks 20
1.5.6 Deep learning hardware requirements 21
1.5.7 Convolutional Neural Network (CNN or ConvNet) 22
1.5.8 CNNs for Object Detection 22
1.5.9 CNN Architectures 22
1.6 Object detection 24
1.6.1 Object detection datasets 25
1.6.2 Evaluation of a model 26

1.6.3 Popular CNN Architectures for Object Detection 29
1.7 Conclusion 31

2 Artificial intelligence in agriculture **32**
2.1 Introduction 32
2.2 Importance 32
2.2.1 Optimizing Resource Utilization 33
2.2.2 Adapting to Climate Change 33
2.2.3 Remote Monitoring and Management 33
2.2.4 Genetic Improvement and Crop Breeding 33
2.2.5 Health Monitoring and Disease Detection 33
2.2.6 Automated Feed Systems 34
2.2.7 Productivity and Yield 34
2.3 Some applications 34
2.3.1 Soil testing and monitoring 34
2.3.2 Robotics in agriculture 35
2.3.3 Automated weed eradication 36
2.3.4 Intelligent Dairy Farmer's Assistant 36
2.3.5 Chicken Robot 37
2.4 Our applications: sheep face detection state of the art 38
2.4.1 Related Work 38
2.5 Conclusion 41

3 methods results and discussion **42**
3.1 Introduction 42
3.2 Method 42
3.2.1 Development environment -Hardwre- 42
3.2.2 Development environment -Software- 43
3.3 Methods and training steps 43
3.3.1 Collecting Data 43
3.3.2 Data labelling 44
3.3.3 Object detection models 46
3.3.4 Pre-processing 48
3.3.5 Model Training 48
3.4 Results 50
3.4.1 The latest checkpoint 50
3.4.2 Tensorboard Graphs 51
3.4.3 The accuracy 53
3.4.4 Video detection 53
3.5 Conclusion 56

General Conclusion **57**

References **58**

LIST OF FIGURES

1.1 how machine learning work[2]. 6
1.2 Supervised Machine learning [4]. 7
1.3 Unsupervised Machine learning [4]. 9
1.4 Semi-Supervised Machine learning [4]. 10
1.5 Reinforcement Learning [4]. 11
1.6 Traditional computing [5]. 14
1.7 K-Nearest Neighbors (KNN) Algorithm for Classification: Real-world Applications and Examples [6]. 16
1.8 Decision Trees [7]. 17
1.10 DNN Layers [10]. 19
1.11 TensorFlow [12]. 20
1.12 PyTorch [13]. 21
1.13 Keras [14]. 21
1.14 JAX [15]. 21
1.15 CNN Architectures [16]. 23
1.16 Max Pooling [17]. 23
1.17 Fully Connected Layer [18]. 24
1.18 Object Detection [20]. 25
1.19 Intersection over Union (IoU) [21]. 27
1.20 precision and recall [22]. 27
1.21 F1 score [22]. 28
1.22 Confusion matrix and evaluation metrics [24]. 29
1.23 The architecture of the R-CNN framework [25]. 30

2.1 Agropad [27]. 35
2.2 Agricultural robots [26]. 35
2.3 Detecting weeds through Computer Vision by the Start-Ups Weedbot and Blue River Technologies [26]. 36

2.4 IDA application [28]. 37
2.5 Chicken Boy Robot [29]. 37

3.1 dataset. 44
3.2 Labelled image. 45
3.3 Resulted XML file 45
3.4 Architecture of SSD MobileNet model [35]. 46
3.5 CenterNet Keypoint Triplets for Object Detection [36]. 47
3.6 Label map. 48
3.7 The latest checkpoint of SSD MobileNet V2 320x320 50
3.8 The latest checkpoint of CenterNet Hourglass 512x512 51
3.9 Loss regularization. 51
3.10 loss localization. 51
3.11 Loss classification. 52
3.12 Loss total loss. 52
3.13 Learning rate. 52
3.14 Loss box scale. 52
3.15 Loss object center. 52
3.16 Loss box offset. 53
3.17 Loss total loss. 53
3.18 Detection 1 53
3.19 Detection 2 53
3.20 Detection 3 53
3.21 Detection 4 54
3.22 Detection 5 54
3.23 Detection 6 54
3.24 Detection 7 54
3.25 Detection 1 55
3.26 Detection 2 55
3.27 Detection 3 55
3.28 Detection 4 55
3.29 Detection 5 55
3.30 Detection 6 55
3.31 Detection 7 55

LIST OF TABLES

3.1 The accuracy table 53

LIST OF ABBREVIATION

- **AGI : Artificial General Intelligence.**
- **AI : Artificielle Intelligence.**
- **ANI : Artificial Narrow Intelligence.**
- **ANN : Artificial Neural Network.**
- **AP : Average Precision.**
- **ASI : Artificial Super Intelligence.**
- **AUC : Area under the curve.**
- **BiFPN : Bidirectional Feature Pyramid Network.**
- **CPU : Central Processing Unit.**
- **DoG : Difference of Gaussian.**
- **DNN : Deep Neural Network.**
- **Faster R-CNN : Faster Region-based Convolutional Neural Network.**
- **FPR : False Positive Rate.**
- **CNN : Convolutional Neural Network.**
- **GPU : Graphics Processing Unit.**
- **HOG : Histogram of Oriented Gradients.**
- **IBM : International Business Machines Corporation.**
- **IDA : Intelligent Dairy Farmer's Assistant.**

- **IoT : Internet of Things.**
- **IoU : Intersection over Union.**
- **JAX : Just Another X.**
- **KNN : K-Nearest Neighbors.**
- **LBP : Local Binary Patterns.**
- **mAP : Mean Average Precision.**
- **ML : Machine Learning.**
- **MobileNetV3 : Mobile Network version 3.**
- **MS COCO : Microsoft Common Objects in Context.**
- **NLP : Natural Language Processing.**
- **PASCAL VOC : Pattern Analysis, Statistical Modelling, and Computational Learning – Visual Object Classes.**
- **PH : Potential of Hydrogen.**
- **RCNN : Region-based Convolutional Neural Network.**
- **ReLU : Rectified Linear Units.**
- **RL : Reinforcement Learning.**
- **ROC : Receiver Operating Characteristic.**
- **SAC-CBAM : Shuffle Attention Channel-Spatial Attention Module with Convolutional Block Attention Module.**
- **SIFT : Scale-Invariant Feature Transform.**
- **SSD : Single Shot Detector.**
- **SVM : Support Vector Machine.**
- **TPR : True Positive Rate.**
- **YOLO : You Only Look Once.**
- **YOLOv4 : You Only Look Once version 4.**
- **YOLOv5 : You Only Look Once version 5.**

GENERAL INTRODUCTION

Artificial intelligence (AI) is revolutionizing various industries, and agriculture is no exception. The integration of AI into farming practices has led to significant advancements, particularly in livestock management. One of the promising applications in this field is sheep face detection, which can greatly enhance the efficiency and accuracy of managing sheep populations.

Traditional methods of identifying individual sheep, such as ear cutting, branding, and ear tagging (both management and electronic), have several drawbacks. These methods are often inaccurate, impractical, and can cause stress to the animals. Ear tags, in particular, can fall off during movement, leading to further issues. While iris and DNA identification methods are more accurate, they are technically demanding, costly, and not suitable for real-time applications. Therefore, there is a pressing need for an easy-to-use, time-efficient, accurate, and non-contact method of identifying individual sheep. Sheep face recognition technology offers a viable solution, providing a non-invasive, efficient, and reliable means of identification that aligns with the welfare standards of modern intelligent sheep farms. Additionally, this technology is crucial for the security and protection of sheep, helping to prevent theft and ensuring the well-being of each animal [1].

This thesis aims to explore sheep face detection using advanced object detection techniques. The study is structured into three main chapters. Chapter 1 digs into object detection techniques, providing an overview of AI and machine learning, and examining different types of machine learning. It also contrasts traditional machine learning with deep learning, highlighting their respective roles in object detection.

Chapter 2 shifts focus to the application of AI in agriculture, emphasizing its importance and various applications. It specifically addresses the state-of-the-art in sheep face detection, discussing current methodologies and their implications for improving livestock management.

Chapter 3 presents the methods, results, and discussion of the study. It details the data collection process, the implementation of deep learning models, and evaluates the performance of SSD MobileNet 320x320 and CenterNet Hourglass 512x512 in detecting sheep faces. The chapter concludes with a discussion on the findings, their significance, and potential areas for future research.

CHAPTER 1 OBJECT DETECTION TECHNIQUES

1.1 Introduction

The world around us is brimming with objects, each playing a vital role in our daily lives. But how can computers, devoid of human perception, understand these objects within images and videos? This chapter explore into the fascinating realm of object detection, a subfield of Artificial Intelligence (AI) that empowers machines to not only recognize objects but also pinpoint their exact locations.

1.2 Artificial Intelligence

Imagine a robot that can learn and make decisions just like a human. That's basically what artificial intelligence AI is all about. It's like a smart computer that can think and learn on its own. AI is a fancy term for machines that act like intelligent humans. According to Andrew Moore, the Former-Dean of the School of Computer Science at Carnegie Mellon University, "Artificial intelligence is the science and engineering of making computers behave in ways that, until recently, we thought required human intelligence." Artificial Intelligence (AI) refers to the simulation of human intelligence in machines that are programmed to mimic and simulate human actions and cognitive processes. These intelligent systems are designed to perceive their environment, reason about the information, and make decisions to achieve specific goals, all without direct human intervention.

1.2.1 Artificial Intelligence application

Daily life

- **Virtual assistants**
- **Recommendation Systems**
- **Online Booking Systems**

Healthcare sector

- **Medical diagnosis**
- **Drug discovery**
- **Personalized treatment**
- **Data security and privacy**

Business sector

- **Personalized costumer experience**
- **Marketing and sales**
- **Finance and accounting**
- **Data security and privacy**

Transport sector

- **Self-drive cares**
- **Localization and mapping**
- **Smart Traffic Management**

Transport sector

- **Aerial data analysis**
- **Real time crop disease and pest detection**
- **Soil moisture monitoring**
- **Autonomous agricultural vehicles**
- **Increased crop yields**

1.2.2 Artificial Intelligence Categories

Artificial Narrow Intelligence (ANI):

Also known as Weak AI, is the only form of AI that currently exists. It is designed to perform a specific task or set of tasks, often outperforming human capabilities in that area. Examples include Siri, Amazon's Alexa, and OpenAI's ChatGPT. While Narrow AI is highly specialized and efficient in its designated task, it cannot operate outside of that task.

Artificial General Intelligence (AGI):

Or Strong AI, is currently only a theoretical concept. AGI can learn and adapt skills to complete new tasks in different contexts without human intervention. It has the potential to perform any intellectual task that a human can.

Artificial Super Intelligence (ASI):

Also known as Super AI , is a theoretical concept with cognitive abilities surpassing humans. It can think, reason, learn, and make judgements beyond human capacity. Super AI could potentially have emotions, needs, beliefs, and desires of its own.

1.3 Machine Learning

Machine learning involves computers learning from datasets, often facilitated by big amount of data. Machine learning revolves around data. The more data a model is trained on, the better it performs. This data can be anything from images and text to sensor readings and financial records. Services like Netflix are used to extract valuable information from users data and make predictions about what users want to watch next and recommends something that they are likely to prefer, to keep them on the platform longer by using machine learning algorithms. Machine learning allows computers to perform tasks to simulate the human intelligence, like recognizing faces and protection from spams it can also be used in areas like shipping and logistics to make predictions and reduce waste. The field of machine learning focuses on allowing machines to learn like humans. It's constantly evolving, with new algorithms and applications emerging all the time.

1.3.1 How machine learning work

Machine learning algorithms are molded on a training dataset to create a model. As new input data is introduced to the trained ML algorithm, it uses the developed model to make a prediction.

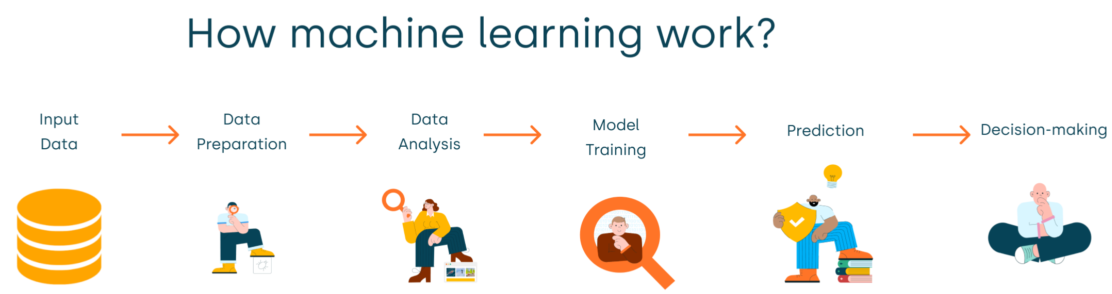

Figure 1.1: how machine learning work[2].

- **Data Collection**

  This data could be images, text, sensor readings, or anything that helps the model to learn.

- **Data Preparation**

  ML depends on the accuracy of the data. Raw data often needs cleaning and organizing. This might involve removing irrelevant information, fixing errors, and ensuring the data format is suitable for the model.

- **Data Analysis**

  Exploratory analysis helps to uncover patterns, trends, and potential issues within the data.

- **Model Training**

  Model training is the heart of machine learning, where an algorithm learns from data to make accurate predictions.

- **Prediction**

  Once trained, the model can make predictions on entirely new, unseen data. It's like putting the knowledge to the test. For example, a spam filter model can predict if a new email is spam based on what it learned during training.

- **Decision Making**

  This is where the predictions become actionable. Based on the model's predictions, The decisions will be able to be informed.

Note: The model learns from its mistakes. If the predictions are wrong, the model adjusts its internal parameters to improve its accuracy for future predictions. This process is iterative, happening many times over with different data points to refine the model's performance.

1.3.2 Types of machine learning

Supervised Machine learning

Supervised Machine learning is based on supervision. It means in the supervised learning technique, we train the machines using the "labelled" dataset, and based on the training, the machine predicts the output. Here, the labelled data specifies that some of the inputs are already mapped to the output. More preciously, we can say; first, we train the machine with the input and corresponding output, and then we ask the machine to predict the output using the test dataset. [3]

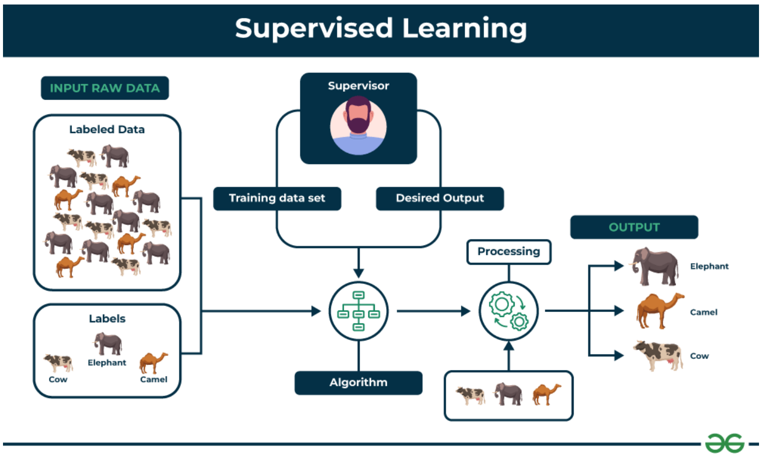

Figure 1.2: Supervised Machine learning [4].

The main goal of the supervised learning technique is to map the input variable(x) with the output variable(y). Some real-world applications of supervised learning are Risk Assessment, Fraud Detection, Spam filtering, etc. Supervised machine learning can be classified into two types of problems, which are:

1. Classification

   Classification algorithms are used to solve the classification problems in which the output variable is categorical, such as "Yes" or No, Male or Female, Red or Blue, etc. The classification algorithms predict the categories present in the dataset. Some real-world examples of classification algorithms are:

   - Spam Detection.

- Email filtering.
- Decision Tree Algorithm.
- Logistic Regression Algorithm.

2. Regression

   Regression algorithms are used to solve regression problems in which there is a linear relationship between input and output variables. These are used to predict continuous output variables, such as market trends, weather prediction, etc. Some popular Regression algorithms are:

   - Simple Linear Regression Algorithm.
   - Multivariate Regression Algorithm.

Advantages and Disadvantages of Supervised Learning

Advantages

- Since supervised learning work with the labelled dataset so we can have an exact idea about the classes of objects.
- These algorithms are helpful in predicting the output on the basis of prior experience.

Disadvantages

- These algorithms are not able to solve complex tasks.
- It may predict the wrong output if the test data is different from the training data.
- It requires lots of computational time to train the algorithm.

Unsupervised Machine Learning

Unsupervised learning is different from the Supervised learning technique; as its name suggests, there is no need for supervision. It means, in unsupervised machine learning, the machine is trained using the unlabeled dataset, and the machine predicts the output without any supervision. In unsupervised learning, the models are trained with the data that is neither classified nor labelled, and the model acts on that data without any supervision.

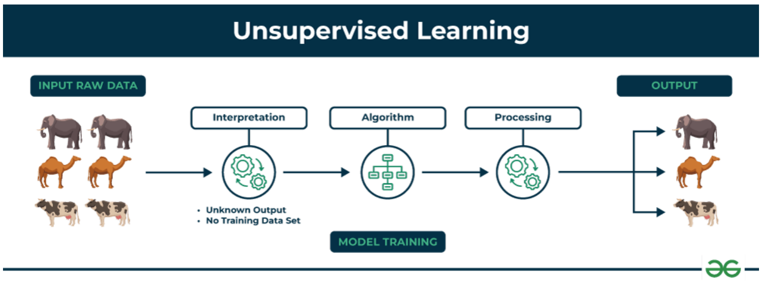

Figure 1.3: Unsupervised Machine learning [4].

The main aim of the unsupervised learning algorithm is to group or categories the unsorted dataset according to the similarities, patterns, and differences. Machines are instructed to find the hidden patterns from the input dataset. Unsupervised Learning can be further classified into two types:

1. Clustering

   The clustering technique is used when we want to find the inherent groups from the data. It is a way to group the objects into a cluster such that the objects with the most similarities remain in one group and have fewer or no similarities with the objects of other groups. Some examples of the clustering algorithm are:

   - Grouping the customers by their purchasing behavior.
   - K-Means clustering algorithm.
   - Mean-shift algorithm.

2. Association

   Association rule learning is an unsupervised learning technique, which finds interesting relations among variables within a large dataset. The main aim of this learning algorithm is to find the dependency of one data item on another data item and map those variables accordingly so that it can generate maximum profit. This algorithm is mainly applied in Market Basket analysis, Web usage mining, continuous production, etc. Some popular algorithms of Association rule learning are:

   - Apriori Algorithm.
   - Eclat.
   - FP-growth algorithm.

Advantages and Disadvantages of Unsupervised Learning

Advantages

- These algorithms can be used for complicated tasks compared to the supervised ones because these algorithms work on the unlabeled dataset.
- Unsupervised algorithms are preferable for various tasks as getting the unlabeled dataset is easier as compared to the labelled dataset.

Disadvantages

- The output of an unsupervised algorithm can be less accurate as the dataset is not labelled, and algorithms are not trained with the exact output in prior.
- Working with Unsupervised learning is more difficult as it works with the unlabelled dataset that does not map with the output.

Semi-Supervised Learning

Semi-Supervised Learning is a machine learning algorithm that works between the supervised and unsupervised learning so it uses both labelled and unlabelled data. It's particularly useful when obtaining labeled data is costly, time-consuming, or resource-intensive. This approach is useful when the dataset is expensive and time-consuming. Semi-supervised learning is chosen when labeled data requires skills and relevant resources in order to train or learn from it. We use these techniques when we are dealing with data that is a little bit labeled and the rest large portion of it is unlabeled. We can use the unsupervised techniques to predict labels and then feed these labels to supervised techniques. This technique is mostly applicable in the case of image data sets where usually all images are not labeled.

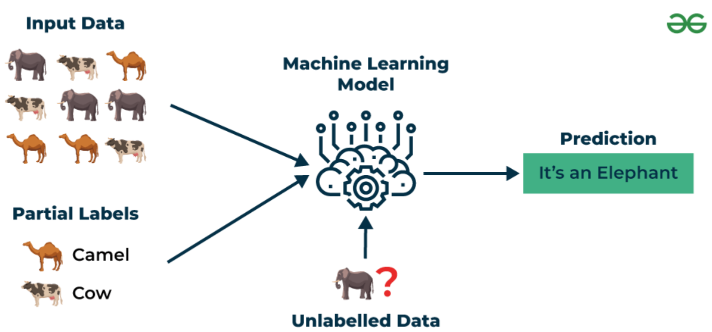

Figure 1.4: Semi-Supervised Machine learning [4].

The main aim of the unsupervised learning algorithm is to group or categories the unsorted dataset according to the similarities, patterns, and differences. Machines are instructed to find the hidden patterns from the input dataset. Unsupervised Learning can be further classified into two types:

Advantages and Disadvantages of Unsupervised Learning

Advantages

- It is simple and easy to understand the algorithm.
- It is highly efficient.
- It is used to solve drawbacks of Supervised and Unsupervised Learning algorithms.

Disadvantages

- Iterations results may not be stable.
- We cannot apply these algorithms to network-level data.
- Accuracy is low.

Reinforcement Learning

Reinforcement learning works on a feedback-based process, in which an AI agent (A software component) automatically explore its surrounding by hitting trail, taking action, learning from experiences, and improving its performance. Agent gets rewarded for each good action and get punished for each bad action; hence the goal of reinforcement learning agent is to maximize the rewards. In reinforcement learning, there is no labelled data like supervised learning, and agents learn from their experiences only.

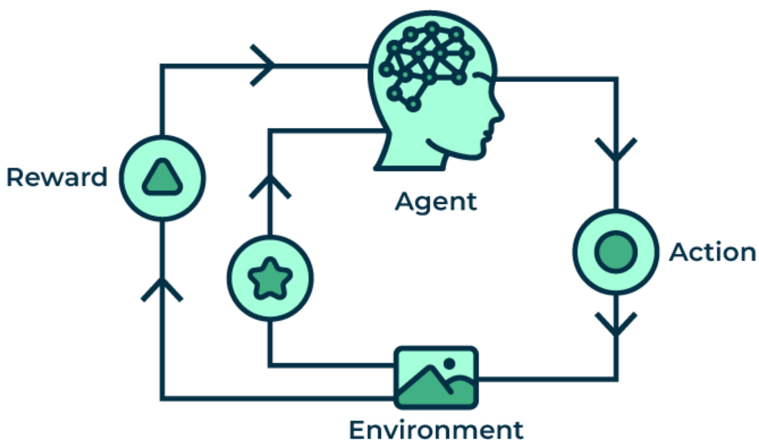

Figure 1.5: Reinforcement Learning [4].

The reinforcement learning process is similar to a human being; for example, a child learns various things by experiences in his day-to-day life Due to its way of working,

reinforcement learning is employed in different fields such as Game theory, Operation Research, Information theory, multi-agent systems.

Categories of Reinforcement Learning

Reinforcement learning is categorized mainly into two types of methods/algorithms:

- **Positive Reinforcement Learning**

  Positive reinforcement learning specifies increasing the tendency that the required behaviour would occur again by adding something. It enhances the strength of the behaviour of the agent and positively impacts it.

- **Negative Reinforcement Learning**

  Negative reinforcement learning works exactly opposite to the positive RL. It increases the tendency that the specific behaviour would occur again by avoiding the negative condition.

Advantages and Disadvantages of Unsupervised Learning

Advantages

- It helps in solving complex real-world problems, which are difficult to be solved by general techniques.
- The learning model of RL is similar to the learning of human beings; hence, most accurate results can be found.

Disadvantages

- RL algorithms are not preferred for simple problems.
- RL algorithms require huge data and computations.
- Too much reinforcement learning can lead to an overload of states which can weaken the results.

1.4 Traditional Machine Learning

Traditional machine learning refers to the classic approach of training algorithms to perform specific tasks by relying on manually engineered features and predefined rules. In contrast, deep learning uses neural networks to automatically learn from raw data without the need for human-designed features. Traditional machine learning methods require human intervention to extract relevant features and design the learning process. Traditional machine learning algorithms learn patterns and relationships by training on labeled datasets. They adjust model parameters based on input features and labels, common algorithms include linear regression, logistic regression, decision trees, support vector machines, and k-nearest neighbors. Traditional machine learning is commonly used in spam detection, sentiment analysis, recommendation systems, and medical diagnosis. However, it may struggle with high-dimensional data or complex relationships that require handcrafted features.

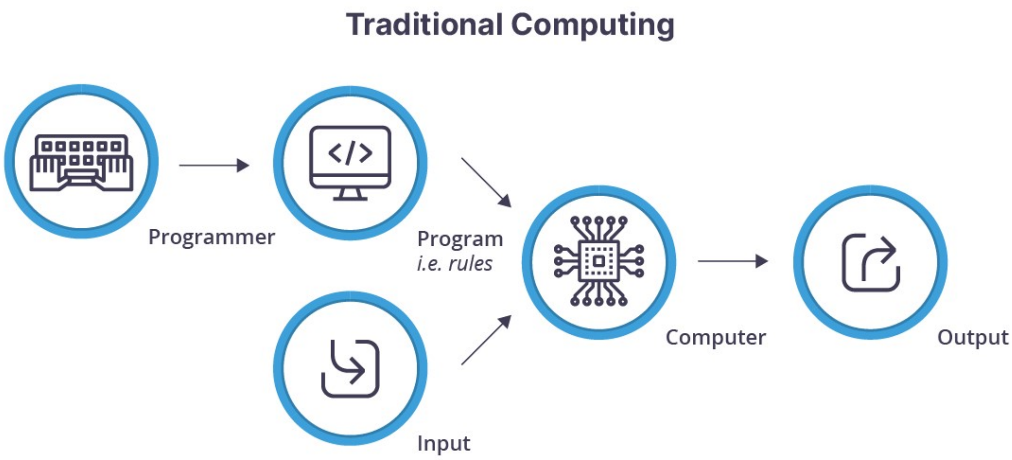

Figure 1.6: Traditional computing [5].

1.4.1 Limitations of Traditional Machine Learning

- **Feature Engineering: These models often rely on handcrafted features, which requires domain expertise to identify the most relevant features from the data. This can be time-consuming and requires human intervention.**
- **Limited Model Complexity: Traditional algorithms might struggle with very complex data or tasks that require learning intricate relationships within the data.**

1.4.2 Strengths of Traditional Machine Learning

- **Interpretability: As mentioned earlier, traditional models are generally easier to understand, making them favorable in applications where understanding the "why" behind a decision is crucial.**
- **Efficiency: These models often require less computational power to train compared to deep learning models. This can be advantageous for resource-constrained environments.**
- **Performance on smaller datasets: Traditional machine learning algorithms can sometimes achieve good performance even with smaller datasets, unlike deep learning models that often require vast amounts of data to perform well.**

1.4.3 Traditional machine learning techniques

Traditional machine learning offers a variety of techniques for object detection in images and videos. While not as powerful as deep learning approaches in raw performance, these methods can be efficient, interpretable, and effective for specific applications. The common techniques:

1. Feature Extraction and Classification: This two-step approach breaks down object detection into separate tasks.

   (a) Feature Extraction

   The first step involves extracting informative features from the image that are relevant for object detection. These features might be edges, shapes, histograms, or gradients. Common techniques include:

   Scale-Invariant Feature Transform (SIFT)

   SIFT is one of the most popular descriptor based point of interest that uses the intensity without color information. For this purpose, the Difference of Gaussian (DoG) is used to identify interest points in image region which are invariant to orientation, scale, illumination, and zoom. This descriptor has been widely used in pattern recognition and classification systems such as face recognition, visual mapping, image stitching. However, when the scale or the size of the image of dataset increases significantly, the disadvantage of the SIFT descriptor is their high computational cost.

   Histogram of Oriented Gradients (HOG)

   The HOG descriptor has been one of the best process accepted to describe image local texture and it is one of the best features of shape and edge information. The HOG descriptor can be describe the shape of the face by distribution of edge direction or light intensity gradient. The process of this technique done by sharing the whole face image into cells (small region or area), a histogram of pixel edge direction or direction gradients is generated of each cell, and finally, the histograms of the whole cells are combined to extract the feature of the face image. It is displays great success in face recognition.

   Local Binary Patterns (LBP)

   LBP is very great general texture descriptor used for feature extraction from image, and has been widely used in a lot of applications, such as face recognition, facial expression recognition, texture segmentation, and texture classification. The most important advantages of this descriptor are its low computational complexity for describing local scene regions (good texture structure description), its invariance to monotonic gray-scale value changes and convenient multi-scale extension. At each pixel, the LBP descriptor presented as binary comparisons of pixel intensities between its eight surrounding pixels and the center pixel. Also, this descriptor has been successfully used in pattern recognition.

(b) Classification

Once features are extracted, a machine learning classifier is trained to recognize specific objects based on these features. Support Vector Machines (SVMs) or decision trees are often used for this task.

K-Nearest Neighbors (KNN)

This is a non-parametric algorithm that classifies data points based on the majority vote of their k nearest neighbors in the training data. The value of k is a parameter that can be tuned to improve the performance of the algorithm.

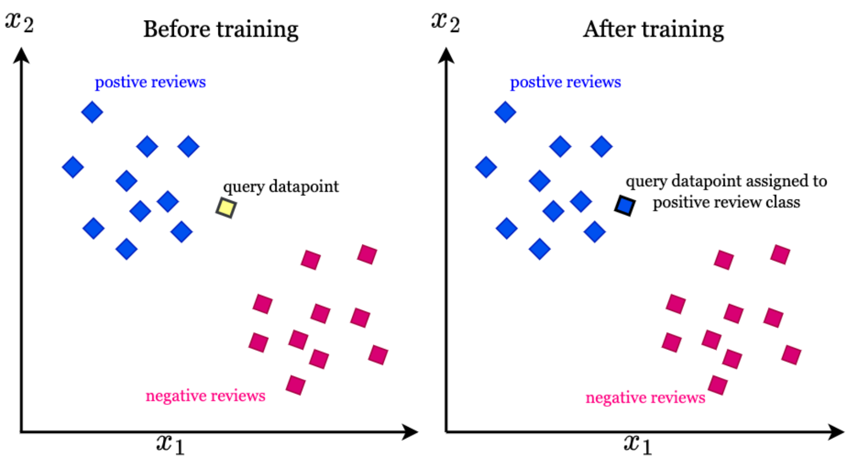

Figure 1.7: K-Nearest Neighbors (KNN) Algorithm for Classification: Real-world Applications and Examples [6].

Decision Trees

This is a tree-like structure that classifies data points by asking a series of questions about their features. Each question in the tree splits the data into two or more subsets, based on the answer to the question. The process continues until the data points in each leaf node of the tree belong to the same class.

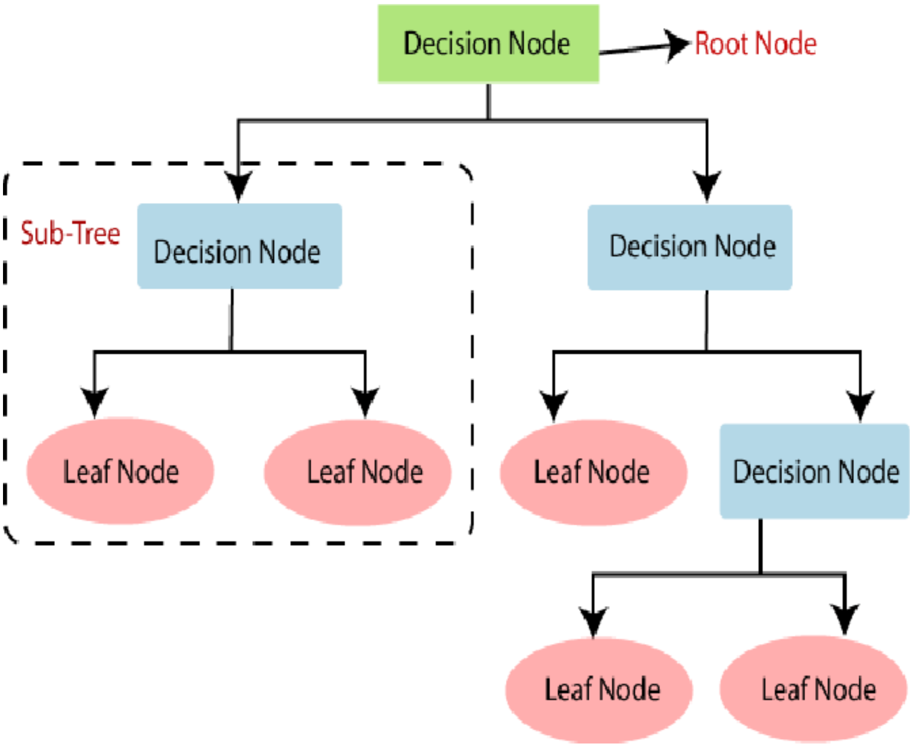

Figure 1.8: Decision Trees [7].

Support Vector Machines (SVMs)

This is a powerful algorithm that can be used for both classification and regression problems. SVMs work by finding a hyperplane that separates the data points of different classes with the maximum margin.

1.5 Deep Learning

Deep learning is a subset of machine learning that uses multi-layered neural networks, called deep neural networks DNN, to simulate the complex decision-making power of the human brain. Some form of deep learning powers most of the artificial intelligence (AI) in our lives today.

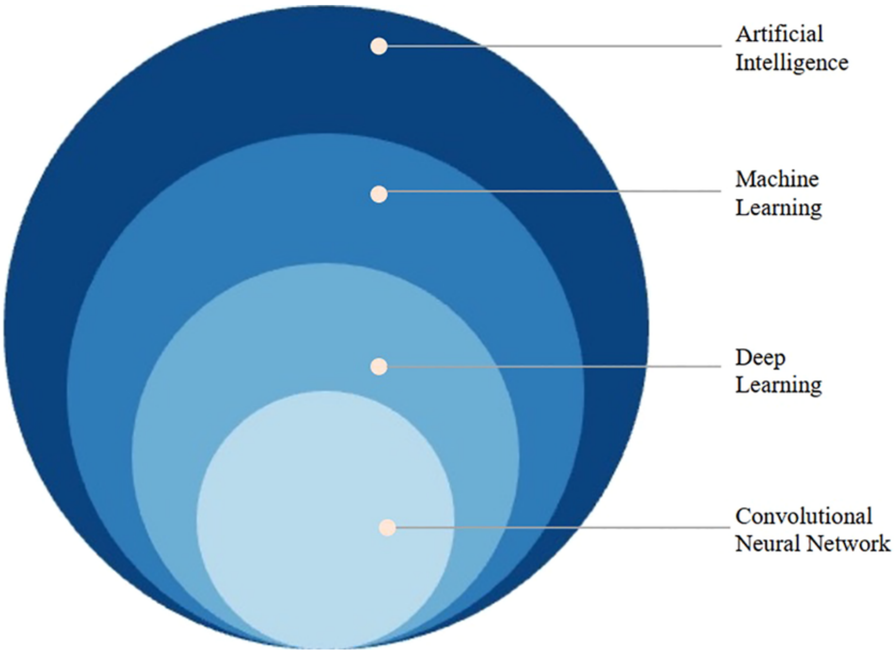

Figure 1.9: The Venn diagram of the artificial intelligence hierarchic terminology [8].

1.5.1 Artificial Neural Network

Artificial Neural Network (ANN) is a type of neural network that is based on a Feed-Forward strategy. It is called this because they pass information through the nodes continuously till it reaches the output node [9]. This is also known as the simplest type of neural network. Some advantages of ANN :

- **Ability to learn irrespective of the type of data (Linear or Non-Linear).**
- **ANN is highly volatile and serves best in financial time series forecasting.**

1.5.2 Deep neural network

A deep neural network, or DNN, is a neural network with three or more layers. In practice, most DNNs have many more layers. DNNs are trained on large amounts of data to identify and classify phenomena, recognize patterns and relationships, evaluate posssibilities, and make predictions and decisions. While a single-layer neural network can make useful, approximate predictions and decisions, the additional layers in a deep neural network help refine and optimize those outcomes for greater accuracy. DNNs consist of multiple layers of interconnected nodes, each building upon the previous layer to refine and optimize the prediction or categorization. This progression of computations through the network is called forward propagation.

The input and output layers of a deep neural network are called visible layers. The input layer is where the deep learning model ingests the data for processing, and the

output layer is where the final prediction or classification is made.

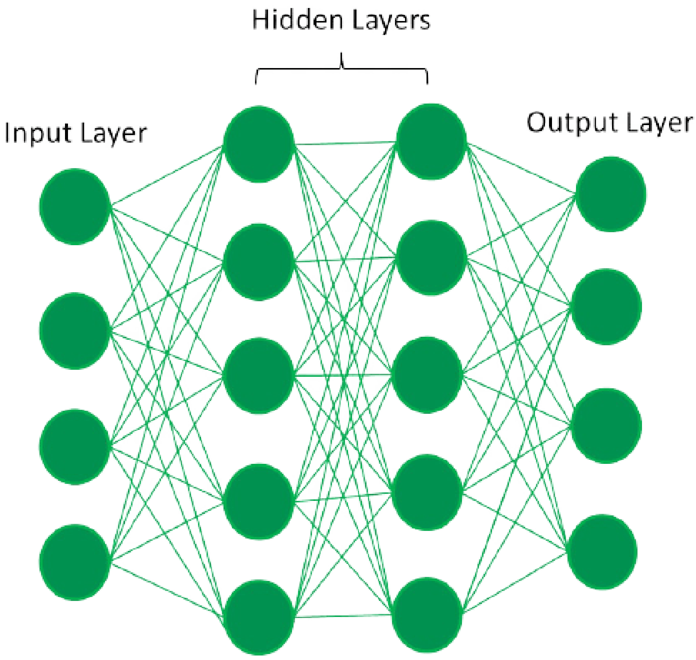

Figure 1.10: DNN Layers [10].

Another process called backpropagation uses algorithms, like gradient descent, to calculate errors in predictions and then adjusts the weights and biases of the function by moving backwards through the layers in an effort to train the model. Together, forward propagation and backpropagation allow a neural network to make predictions and correct for any errors accordingly. Over time, the algorithm becomes gradually more accurate [11].

1.5.3 How Deep Learning for Object Detection Works

Data Preparation

Images are labelled with bounding boxes around the objects to be detected and their corresponding class labels for example: car, person, sheep,bird , ect..

Network Architecture

A deep learning model like YOLO or Faster R-CNN.. is chosen and fine-tuned for the specific object detection task.

Training

The model is trained on a large dataset of labeled images. During training, the model learns to extract features from the images that are relevant for object detection and classification.

Making Detections

Once trained, the model can process new images. It identifies potential objects in the image, predicts bounding boxes around them, and classifies the objects within those boxes.

1.5.4 Advantages of Deep Learning for Object Detection

High Accuracy

Deep learning models can achieve higher accuracy compared to traditional techniques, especially for complex object detection tasks with cluttered backgrounds or variations in object appearance.

Feature Learning

Deep learning eliminates the need for manual feature engineering. The model automatically learns discriminative features from the data during training.

Adaptability

Deep learning models can be adapted to detect a wide range of objects by modifying the network architecture and training data.

1.5.5 Deep Learning Frameworks

TensorFlow

TensorFlow is an open source software library for numerical computation using data flow graphs developed by Google. Nodes in the graph represent mathematical operations, while the graph edges represent the multidimensional data arrays (tensors) that flow between them. This flexible architecture allows you to deploy computation to one or more CPUs or GPUs in a desktop, server, or mobile device without rewriting code. For visualizing TensorFlow results, TensorFlow offers TensorBoard, a suite of visualization tools [12].

Figure 1.11: TensorFlow [12].

PyTorch

Developed by Facebook, PyTorch is known for its ease of use and dynamic computational graph. It's often favored for rapid prototyping and research due to its intuitive interface.

Figure 1.12: PyTorch [13].

Keras

While not a standalone framework, Keras is a high-level API that can be used on top of TensorFlow or other frameworks. It provides a simpler interface for building and training models, making it easier for beginners to learn.

Figure 1.13: Keras [14].

JAX

Developed by Google, JAX is a high-performance framework based on the NumPy library. It offers strong numerical computing capabilities and is well-suited for research involving scientific computing.

Figure 1.14: JAX [15].

1.5.6 Deep learning hardware requirements

Deep learning requires a tremendous amount of computing power. High performance graphical processing units (GPUs) are ideal because they can handle a large volume of calculations in multiple cores with copious memory available. However, managing multiple GPUs on-premises can create a large demand on internal resources and be incredibly costly to scale.

1.5.7 Convolutional Neural Network (CNN or ConvNet)

Convolutional neural networks (CNNs) are artificial intelligence systems based on multi-layer neural networks that can identify, recognize, and classify objects as well as detect and segment objects in images. In fact, CNN or ConvNet is a popular discriminative deep learning architecture that could be learned directly from the input object. This network is frequently used in visual identification, medical image analysis, image segmentation, NLP, and many other applications since it is specifically designed to deal with a range of 2D shapes. It is more effective than a regular network since it can automatically identify key elements from the input without the need for human participation.

1.5.8 CNNs for Object Detection

Convolutional Neural Networks (CNNs) are widely used for object detection due to their ability to effectively learn and represent spatial hierarchies in images. Here are several reasons why:

Learning Spatial Features

Traditional neural networks struggle with spatial information in images. CNNs address this with convolutional layers that use filters to learn patterns and features directly from the image's pixel grid. These features often capture edges, shapes, and textures crucial for object recognition.

Parameter Sharing

CNNs share weights across filters, reducing the number of parameters to learn compared to fully connected layers. This improves efficiency and helps prevent overfitting, a common problem in machine learning.

Local Connectivity

Neurons in a convolutional layer are only connected to a small region of the previous layer, mimicking the receptive field of biological neurons in the visual cortex. This local processing allows CNNs to capture local features effectively.

1.5.9 CNN Architectures

Convolutional Layers

The core building blocks of a CNN are convolutional layers. These layers use filters, also known as kernels, to perform convolution operations on the input image. Each filter slides over the input image, computing dot products with local patches to produce feature maps. Feature maps capture information about different aspects of the input, such as edges, textures, and shapes. Multiple convolutional layers with increasing complexity are stacked to learn hierarchical representations of the input image.

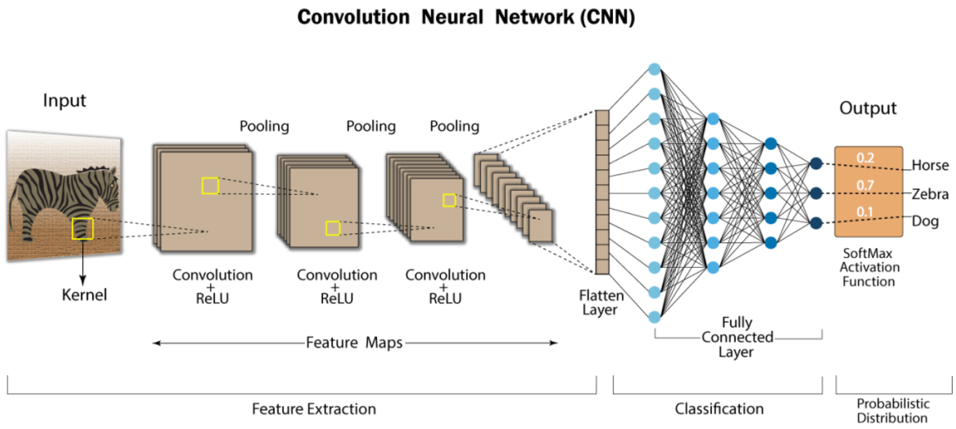

Figure 1.15: CNN Architectures [16].

Pooling Layers

This layer is periodically inserted in the covnets and its main function is to reduce the size of volume which makes the computation fast reduces memory and also prevents overfitting. Two common types of pooling layers are max pooling and average pooling. If we use a max pool with 2 x 2 filters and stride 2, the resultant volume will be of dimension 16x16x12 [17].

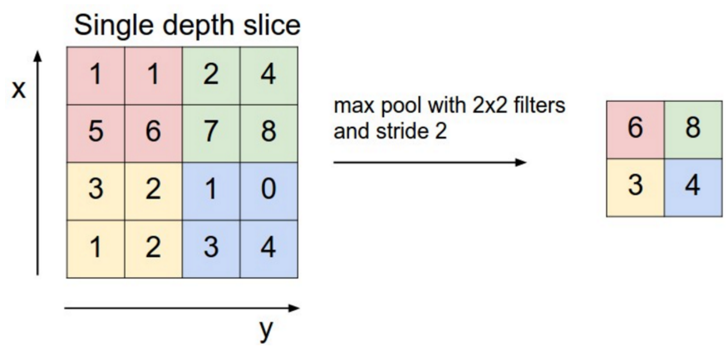

Figure 1.16: Max Pooling [17].

Activation Layers

Activation functions introduce non-linearity into the network, enabling it to learn complex mappings between input and output. ReLU (Rectified Linear Units) is a commonly used activation layers in CNNs due to its simplicity and effectiveness. ReLU sets negative values to zero and preserves positive values, promoting faster convergence and better gradient propagation during training.

Fully Connected Layers

Fully connected layers, also known as dense layers, are typically employed at the end of the CNN architecture to perform high-level reasoning and classification based

on the learned features. Each neuron in a fully connected layer is connected to every neuron in the previous layer, allowing for complex interactions and decision-making. These layers take the high-level features extracted earlier and use them to perform the final object detection tasks. This can involve:

- Classification

  Classifying the object within the image (e.g., car, cat, person).

- Bounding Box Regression

  Predicting the exact location of the object by defining a bounding box around it.

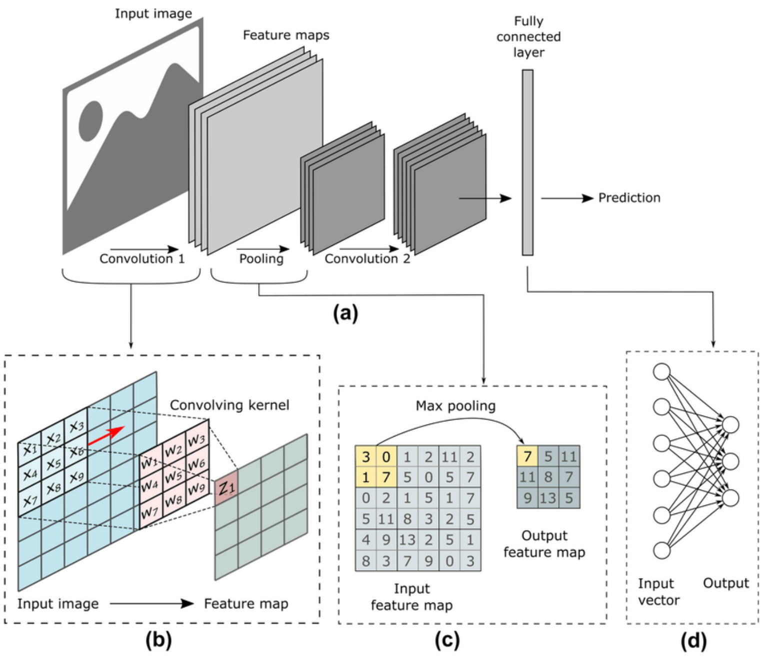

Figure 1.17: Fully Connected Layer [18].

1.6 Object detection

Object detection is a profound computer vision technique that identifies and labels objects within images, videos, and even live footage. Models that perform object detection are trained with a surplus of annotated visuals in order to carry out this process with new data. It becomes as simple as feeding input visuals and receiving a fully marked-up output visual. A key component is the object detection-bounding box, which identifies the edges of the object tagged with a clear-cut quadrilateral — typically a square or rectangle. They are accompanied by a label of the object, whether it be a person, a car, or a dog to describe the target object. Bounding boxes can overlap to showcase multiple objects in a given shot as long as the model has prior knowledge of the items it is tagging [19].

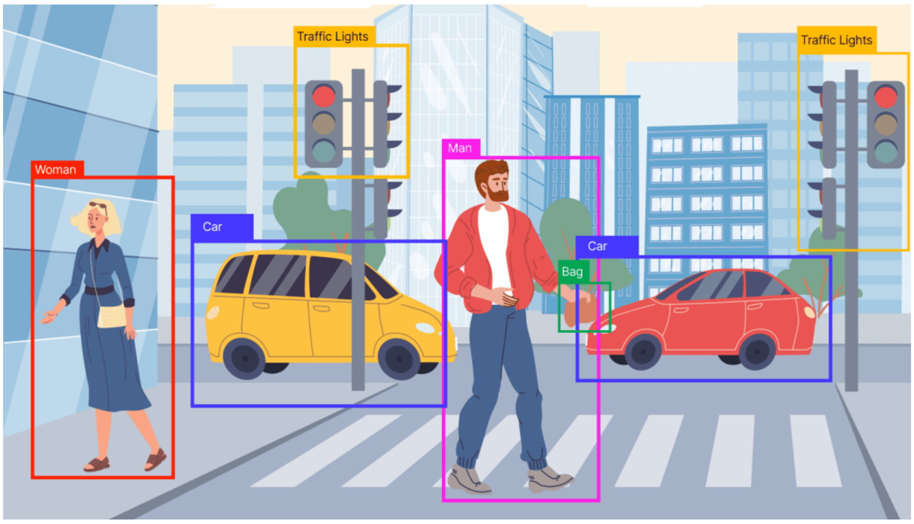

Figure 1.18: Object Detection [20].

1.6.1 Object detection datasets

Object detection datasets are crucial for training and evaluating object detection models. They consist of images or videos that are annotated with bounding boxes around the objects of interest. These annotations allow the model to learn the relationship between pixels in an image and the presence and location of objects.

1 MS COCO

- Description: COCO is one of the most popular datasets for object detection, segmentation, and captioning. It contains images of complex everyday scenes with common objects in their natural contexts.
- Size: Over 330,000 images with 80 object categories.
- Annotations: Object segmentation masks, bounding boxes, keypoints, and image captions.

2 PASCAL VOC

- Description: PASCAL VOC is a benchmark in visual object category recognition and detection.
- Size: Approximately 20,000 images with 20 object categories.
- Annotations: Bounding boxes, object segmentation, and object part annotations.

3 ImageNet

- **Description:** ImageNet, known for its large scale and diversity, is often used for image classification but also contains bounding box annotations for object detection.
- **Size:** Millions of images with thousands of object categories.
- **Annotations:** Bounding boxes for object detection.

4 Open Images

- **Description:** Open Images is a vast dataset containing a wide variety of object categories with a high level of annotation detail.scenes with common objects in their natural contexts.
- **Size:** About 9 million images with 600 object categories.
- **Annotations:** Bounding boxes, object segmentation masks, and visual relationships.

1.6.2 Evaluation of a model

Accuracy

Accuracy is a measure of how well the model's predictions match the ground truth annotations. In object detection, this involves correctly predicting both the presence and location (bounding box) of each object, as well as its class.

Calculating Accuracy in Object Detection

To calculate accuracy in object detection, you need to consider both detection (correctly locating the object) and classification (correctly identifying the object category). Here are the steps and metrics involved:

1. **Intersection over Union (IoU) :** Intersection over Union (IoU) is a measure that shows how well the prediction bounding box aligns with the ground truth box. It's one of the main metrics for evaluating the accuracy of object detection algorithms and helps distinguish between "correct detection" and "incorrect detection". By measuring how well the model's prediction describes the actual region of interest, the IoU score, alongside other evaluation measures, helps researchers gauge the effectiveness and reliability of their models and make informed decisions about algorithm performance.

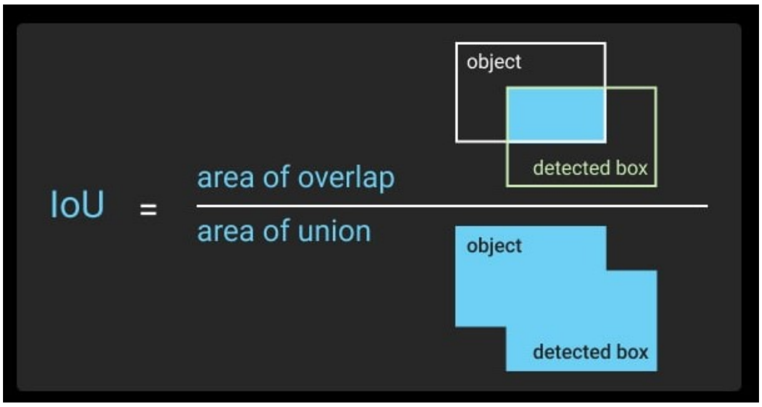

Figure 1.19: Intersection over Union (IoU) [21].

2. **Precision and Recall : While precision focuses on accurately identifying relevant objects, recall emphasizes the model's capability to find all ground truth bounding boxes. Together, precision and recall weigh the balance between prediction quality and quantity.**

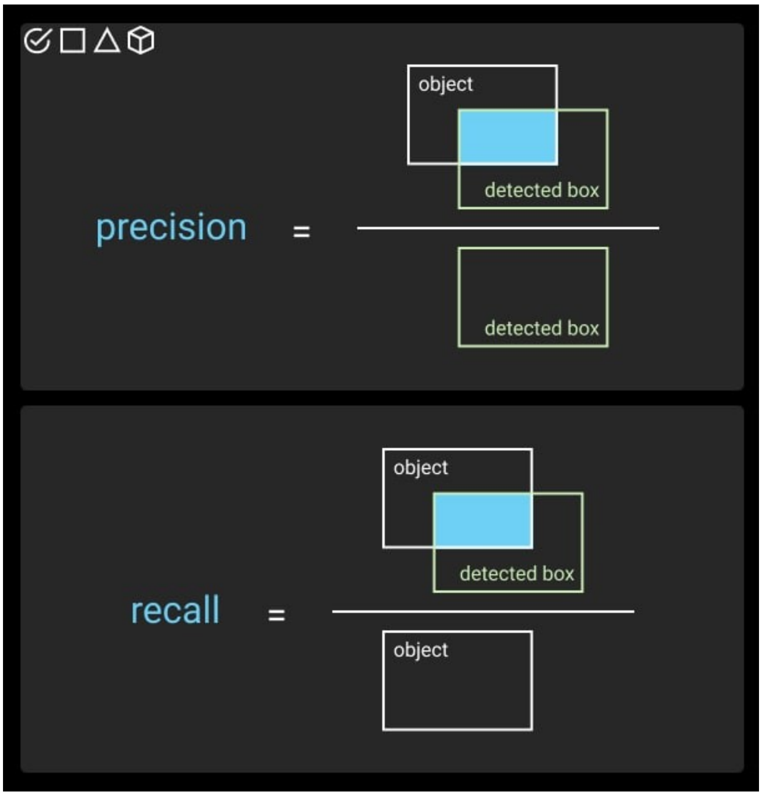

Figure 1.20: precision and recall [22].

3. **Average Precision (AP) : AP stands as the fundamental metric for object detection, which integrates precision, recall, and the model's confidence in each detection. Calculated separately for each class, average precision object detection condenses the Precision x Recall curve into a single numerical summary [22].**

4. **Mean Average Precision (mAP):** Mean Average Precision (mAP) builds on the idea of AP, specifically in multi-class scenarios. It is computed by averaging the AP across all classes. The metric considers precision and recall for various IoU thresholds and object classes, with a higher mAP indicating superior overall model performance[22]. mAP is a popular metric for evaluating object detection models because it is easy to understand and interpret. It is also relatively insensitive to the number of objects in the image. A high mAP score indicates that the model can detect objects with both high precision and recall, which is critical in applications like autonomous driving where reliable object detection is pivotal to avoiding collisions. A perfect mAP score of 1.0 suggests that the model has achieved flawless detection across all classes and recall thresholds. Conversely, a lower mAP score signifies potential areas of improvement in the model's precision and/or recall [21].

5. **F1 Score :** F1 represents a trade-off between precision and recall, calculated as their harmonic mean.

   - **Receiver operating characteristic (ROC) curve:** plots the true positive rate (TPR) versus the false positive rate (FPR) as a function of the model's threshold for classifying a positive data point
   - **Area under the curve (AUC):** metric to calculate the overall performance of a classification model based on area under the ROC curve

$$F_1 = \frac{2 \cdot (\text{precision} \cdot \text{recall})}{(\text{precision} + \text{recall})}$$

Figure 1.21: F1 score [22].

6. **Confusion Matrix :** A confusion matrix is a matrix that summarizes the performance of a machine learning model on a set of test data. It is a means of displaying the number of accurate and inaccurate instances based on the model's predictions. It is often used to measure the performance of classification models, which aim to predict a categorical label for each input instance [23].

ACTUAL VALUES

PREDICTED VALUES	Positive	Negative	
Positive	True Positive (TP)	False Positive (FP)	Precision= TP/TP+FP
Negative	False Negative (FN)	True Negative (TN)	NPV= TN/TN+FN
	Recall/Sensitivity= TP/TP+FN	Specificity= TN/TN+FP	ACCURACY= $\frac{TP+TN}{TP+TN+FN+FP}$

Figure 1.22: Confusion matrix and evaluation metrics [24].

The matrix displays the number of instances produced by the model on the test data.

1. **True positives: data points labeled as positive that are actually positive [23].**
2. **False positives: data points labeled as positive that are actually negative [23].**
3. **True negatives: data points labeled as negative that are actually negative [23].**
4. **False negatives: data points labeled as negative that are actually positive [23].**

1.6.3 Popular CNN Architectures for Object Detection

Several CNN architectures have been specifically designed for object detection:

1. **Region-based convolutional neural network (RCNN) : The region-based convolutional neural network (RCNN) was proposed by Ross Girshick et al. in 2014 and was one of the first successful deep learning approaches to detect objects. RCNN operates by generating a set of region proposals using an external algorithm, applying a pre-trained convolutional neural network (CNN) to each proposal to extract features, and finally classifying each proposal using a support vector machine (SVM).**

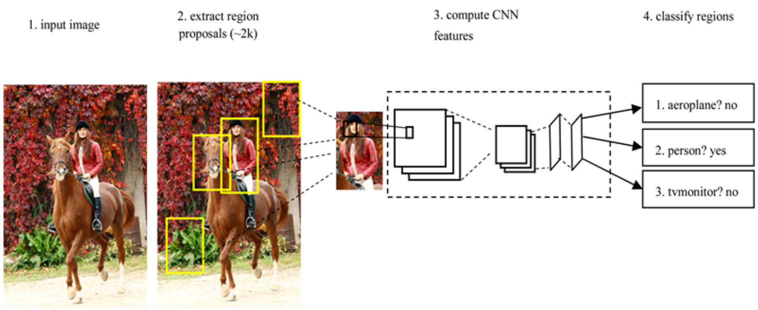

Figure 1.23: The architecture of the R-CNN framework [25].

2. **You Only Look Once (YOLO) : YOLO divides an image into a grid and predicts bounding boxes and class probabilities for each grid cell. Unlike sliding window and region proposal-based techniques, YOLO looks at the whole image only once, hence the name. The original YOLO has seen multiple versions, with YOLOv2 (YOLO9000), YOLOv3, and YOLOv4 introducing various improvements in accuracy, speed, and the ability to detect a wider range of object sizes. Its tiny versions are extremely fast, making it suitable for real-time object detection and even deployment on edge devices. Usually, it has a good balance between precision and recall.**

3. **Single Shot Detector (SSD): SSD predicts multiple bounding boxes and class scores for those boxes in one pass. It does this by using multiple feature maps from different network layers to predict detections at various scales. It uses a base network (often a VGG16 trained on ImageNet) for feature extraction. This network is truncated before the fully connected layers, allowing the model to take inputs of any size. One of SSD's key contributions is using feature maps from different layers in the network to predict detections at multiple scales. This makes it capable of detecting objects of varying sizes, addressing one of the significant challenges in object detection. Smaller feature maps are used to detect larger objects, and larger feature maps detect smaller objects.**

1.7 Conclusion

In summary, this chapter offered a clear overview of artificial intelligence and its subsets, particularly focusing on machine learning. We distinguished between traditional and deep learning approaches, setting the stage for deeper exploration into the evolving landscape of AI technologies.

CHAPTER 2
ARTIFICIAL INTELLIGENCE IN AGRICULTURE

2.1 Introduction

Agriculture is the backbone of civilization, providing food, fiber, and fuel for billions of people. However, the industry faces numerous challenges, including a growing population, climate change, and resource scarcity. Artificial intelligence is emerging as a powerful tool to address these challenges and revolutionize the way we farm. In this chaptre will explore the transformative potential of AI in agriculture. We will discuss its importance in increasing efficiency, optimizing resource use, and ensuring food security.

2.2 Importance

Artificial intelligence is having a major impact on the world of agriculture, and is seen as a key tool to address future challenges of food production.Here is why AI is important in agriculture :

2.2.1 Optimizing Resource Utilization

Traditional farming methods often rely on inefficient practices that waste resources such as water, fertilizers, and pesticides. AI enables precision agriculture, where farmers can precisely tailor inputs based on real-time data, optimizing resource utilization and minimizing waste [1].

2.2.2 Adapting to Climate Change

Climate change poses unprecedented challenges to agriculture, including extreme weather events, shifting growing seasons, and unpredictable rainfall patterns. AI helps farmers adapt to these changes by providing predictive analytics and decision support tools that enable proactive management of climate-related risks [1].

2.2.3 Remote Monitoring and Management

One of the significant advantages of AI in agriculture is its ability to enable remote monitoring and management of agricultural operations. Through the use of (IoT) devices, drones, and satellite imagery, farmers can monitor crop conditions, detect anomalies, and manage resources from anywhere, reducing the need for physical presence in the field and improving operational efficiency [1].

2.2.4 Genetic Improvement and Crop Breeding

AI accelerates genetic improvement and crop breeding processes by facilitating the analysis of genomic data and trait prediction. Machine learning algorithms analyze vast genomic datasets to identify genes associated with desirable traits such as drought tolerance, disease resistance, and yield potential. This enables breeders to develop new crop varieties with improved traits more efficiently, addressing specific challenges faced by farmers in different agroecological regions [1].

2.2.5 Health Monitoring and Disease Detection

AI-powered systems are revolutionizing health monitoring and disease detection in livestock. By utilizing sensors and data analysis, these systems can detect early signs of illness, allowing for timely intervention. This early detection is crucial as it helps in reducing the spread of diseases among livestock, ensuring healthier herds and more efficient farm management [1].

2.2.6 Automated Feed Systems

AI controls automated feeding systems to distribute the right amount of feed at the right time, significantly improving growth rates and feed efficiency. This precise management ensures that each animal receives optimal nutrition, reducing waste and enhancing overall productivity[1].

2.2.7 Productivity and Yield

- Milk Production

  In dairy farming, AI monitors milk yield and quality, helping farmers optimize production processes and maintain herd health [1].

- Growth Tracking

  AI systems track the growth and development of livestock, providing insights into health and productivity[1].

2.3 Some applications

The agriculture sector is experiencing rapid adoption of artificial intelligence in terms of farming techniques and agricultural products. With each passing day, technology expands its wings, providing farmers with solutions to even the most minor field problems. AI-based technical advancements have made it possible for farmers to produce more goods with less resources and even to improve the quality of those goods, ensuring a faster time to market for the harvested crops. Following are some of the applications of artificial intelligence in agriculture[26].

2.3.1 Soil testing and monitoring

IBM (2018) developed a mini soil testing system, ‘Agropad’ that can successfully test five soil indicators based on colorimetric tests. It is done by farmers by putting a drop of soil or water on the test strip. The five indicators change colour based on the levels of pH, nitrogen dioxide, aluminum, magnesium, and chlorine that are present in the sample. The application makes a recommendation to the farmer for fertiliser adjustments that will help optimise the crop’s growth [26].

Figure 2.1: Agropad [27].

2.3.2 Robotics in agriculture

Robotics is widely employed in many industries, mostly in manufacturing, to carry out complex jobs. Currently, several AI companies are creating robots for use in the agriculture industry. These AI robots are designed so that they can do a variety of farming activities. AI robots are also taught to inspect the quality of crops, find and eradicate weeds, and harvest harvests more quickly than a person[26].

Figure 2.2: Agricultural robots [26].

2.3.3 Automated weed eradication

Weeds can be quickly and readily recognised with AI sensors, which can also identify weed-affected locations. Herbicides may be accurately applied in these locations after locating them, which reduces the need for herbicides while also saving time and crop. Various AI start-ups are developing weed-spraying robots that are accurately guided by computer vision and AI. AI sprayers can significantly reduce the amount of pesticides that must be applied to fields, improving crop quality and lowering costs[26].

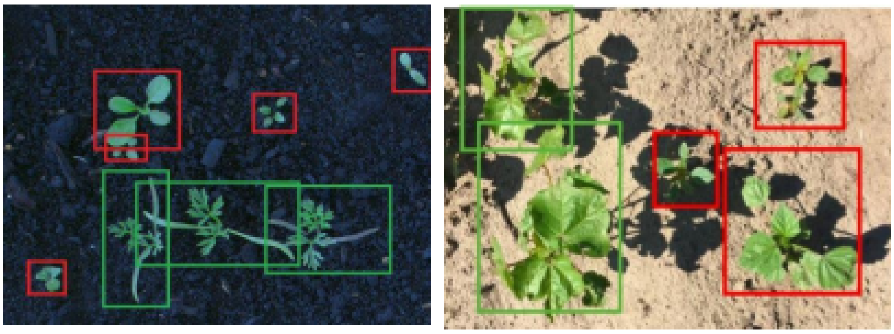

Figure 2.3: Detecting weeds through Computer Vision by the Start-Ups Weedbot and Blue River Technologies [26].

2.3.4 Intelligent Dairy Farmer's Assistant

One notable application of AI in livestock farming system developed by Connecterra. IDA uses machine learning to analyze data from sensors on cows, monitoring health and activity parameters. It provides farmers with real-time insights and recommendations, enabling early detection of health issues, optimized feeding, and improved efficiency and productivity in dairy operations[28].

Figure 2.4: IDA application [28].

2.3.5 Chicken Robot

The "ChickenBoy" system by Faromatics utilizes AI to revolutionize poultry farming. Equipped with sensors and cameras.ChickenBoy monitors environmental and health factors in poultry houses, providing real-time data on temperature, humidity, air quality, and bird behavior. Its AI algorithms process this data to offer actionable insights to farmers, enabling prompt resolution of issues and enhancing bird welfare and productivity[29].

Figure 2.5: Chicken Boy Robot [29].

2.4 Our applications: sheep face detection state of the art

Sheep face detection is a cutting-edge area in animal biometrics, aimed at improving livestock management through advanced computer vision techniques. Recent studies have explored various algorithms to accurately detect and recognize individual sheep based on their facial features. This technology holds promise for enhancing the efficiency and welfare in sheep farming. This review highlights significant research and applications in the state of the art of sheep face detection.

2.4.1 Related Work

The paper ”Sheep Face Detection Based on an Improved RetinaFace Algorithm” by Jinye Hao et al [30],proposes an effective and lightweight sheep face detection method based on an improved RetinaFace algorithm, we will go over the most critical information:.

- Data collection

  The data used in this study is a total of 3749 images of 119 sheep. It includes 1415 images of 62 Ningxia Tan sheep and 2334 images of 57 dairy goats [30].

- Key Improvements

  1. Enhanced Backbone Network: The authors employ an enhanced MobileNetV3-Large network as the backbone. [30].
  2. Attention Module: They incorporate a Shuffle Attention Channel-Spatial Attention Module (SAC-CBAM) to enhance feature extraction[30].

- Performance Metrics

  The proposed method achieved :

  - An F1-score of 95.25% [30].
  - An average precision of 96.00% [30].
  - A model size of 13.20 MB [30].
  - An average processing time of 26.83 ms [30].
  - A parameter count of 3.20 million [30].

Evaluation

1. Comparison: Qualitative and quantitative evaluations demonstrated the superior performance of the proposed method compared to other face detectors like YOLO and RetinaFace [30].
2. Robustness: The method showed robust sheep face detection under different lighting conditions [30].
3. Ablation Studies: These studies confirmed the effectiveness of the key components, including the SAC module and the combination of MobileNetV3-Large with the attention mechanism [30].

In summary, the paper presents a state-of-the-art sheep face detection approach that achieves high accuracy and efficiency, making it suitable for various applications in animal biometrics and precision livestock farming.

In the the work of Song et al,entitled " Using Pruning-Based YOLOv3 Deep Learning Algorithm for Accurate Detection of Sheep Face " [31], the authors utilized a method based on YOLOv3 model, we will go over the most critical information :

- Data collection

  Sheep face data were collected for each sheep in turn. The dataset collected video data from 20 adult Sunit sheep, In order to make the collected dataset with a certain complexity, different lighting and different angles are used for the shooting , a total of 20 sheep were captured, with 1–3 min of video captured for each sheep[31].

- Method

  In this study, a sheep face detection method based on YOLOv3 model pruning is proposed, abbreviated as YOLOv3-P [31].

- Results

  - The mean Average Precision increased from 95.3 % to 96.4% after clustering the anchor frames[31].
  - Further compression of the model improved the mAP from 96.4% to 97.2% [31].
  - The model size was reduced to one-quarter of the original[31].
  - A 10-fold cross-validation experiment yielded an mAP of 96.84% [31].

- Model comparison

  YOLOv3 showed superior performance compared to YOLOv4 and Faster-RCNN, with higher mAP and F1-score, while maintaining faster detection speed. Despite slightly lower metrics compared to SSD, YOLOv3's faster speed made it the preferred choice. Optimization via K-means clustering further improved YOLOv3's mAP by 1.1% [31].

Finly ,this method is effective for identifying sheep and has significant applications in precision animal management and good farming.

In the work of Wenhao Hong et al,entitled "Goat and Sheep Face Detection Algorithm Based on YOLOv5-Swin-Transformer-BiFPN" [31], the authors utilized a combination of YOLOv5, Swin Transformer, and BiFPN architectures [32] . Here are the key points about the algorithm :

- Data collection
  1. Initial Dataset
     250 images were initially selected for the experiment [32].
  2. Data Augmentation
     * Applied image rotation, brightness, and contrast adjustments to expand the dataset to 2500 images [32].
     * Final dataset composition: 1500 goat images and 1000 sheep images [32].
  3. Dataset Splitting
     Divided the dataset into a training set and a test set in a 7:1 ratio [32].
  4. Annotation Process
     * Used Labelme tool to mark goat and sheep faces [32].
     * Converted the obtained JSON files to TXT format suitable for the YOLOv5 model [32].

- Method
  - Base Model : Start with the YOLOv5 model as the baseline for face detection [32].
  - Swin-Transformer Integration : Integrate the Swin-Transformer module into the YOLOv5 backbone network to enhance the model's capability to capture global information and reduce interference from complex backgrounds [32].
  - BiFPN Integration : Introduce the BiFPN to the model for effective multi-scale feature fusion, improving the detection of small and occluded targets [32].

- **Results**
  - **Precision: 93.4% [32] .**
  - **Recall: 78.7% [32].**
  - **mAP@0.5: 87.4% [32].**
  - **A FPS: 61 frames per second [32].**
- **Comparison with other methods**

  The proposed model outperforms other target detection algorithms such as Faster-RCNN-resnet50, Faster-RCNN-vgg, YOLOv4, and standard YOLOv5 [32].

In summary, this paper introduces a cutting-edge approach for sheep face detection, boasting high accuracy and efficiency. Its versatility makes it ideal for applications in animal biometrics and precision livestock farming.

2.5 Conclusion

Chapter 2 highlights the transformative role of AI in agriculture, enhancing efficiency and productivity. It covers various applications like crop monitoring and precision farming, showcasing AI's potential to tackle agricultural challenges. Specifically, it discusses the advanced application of sheep face detection, which improves animal management and welfare. Overall, AI is pivotal for a sustainable and efficient future in agriculture .In the next chapter will delve into the methods and results of our specific project.

CHAPTER 3 METHODS RESULTS AND DISCUSSION

3.1 Introduction

Building upon the foundation established in Chapter 1 on object detection techniques, this chapter focuses on the application of deep learning for automatic sheep face detection. Traditional methods for sheep detection can be time-consuming and prone to error. This chapter explores the potential of deep learning to address these limitations. We dig into the implementation of two specific deep learning models, the SSD MobileNet V2 320x320 and CenterNet Hourglass 512x512, utilizing Python, and TensorFlow2 libraries. By comparing these models on a CPU platform, we aim to identify the optimal approach for accurate and efficient sheep face detection.

3.2 Method

This section explains methods and experiments used in this thesis to produce the final results.

3.2.1 Development environment -Hardwre-

Various hardware components were employed in this study to support different stages of the research process. A PC with a GPU GeForce GTX1070 8GB facilitated computational tasks, with the GPU specifically used for model training. Smartphones were utilized for image data collection, while a dedicated camera enabled precise real-time detection. This hardware integration optimized workflow efficiency and contributed to the study's success.

3.2.2 Development environment -Software-

In this study, a range of software tools was employed to support various research tasks. LabelImg ensured precise dataset labeling, Python facilitated algorithm implementation, and TensorFlow enabled efficient deep learning model development and training. Together, these tools streamlined research processes, enhancing data handling and model development efficiency.

1. Labelimg The open-source LabelImg software empowers efficient annotation of the sheep face detection dataset. It allows to create precise bounding boxes for sheep faces in the images. LabelImg automatically saves annotations in the XML. solution simplifies dataset preparation for sheep face detection .

2. Python Python is a high-level, interpreted programming language known for its readability, simplicity, and versatility. It supports multiple programming paradigms, including procedural, object-oriented, and functional programming, making it suitable for a wide range of applications from web development to data analysis and artificial intelligence.

3. TensorFlow Model Zoo The TensorFlow Model Zoo is a repository where you can find pre-trained models for various machine learning tasks, including object detection. These models are trained on large datasets like COCO, Kitti, and Open Images. and can be fine-tuned or used directly for specific tasks. TensorFlow Model Zoo offers a rich selection of pre-trained models and tools for object detection. Here are some prominent options [33]:

   - CenterNet
   - EfficientDet
   - SSD MobileNet
   - SSD ResNet
   - Faster R-CNN
   - ExtremeNet
   - Mask RCNN And many more.

3.3 Methods and training steps

3.3.1 Collecting Data

The dataset used in this study consists of 3818 high-resolution images of 35 sheep, including 2510 images of 24 Rambi sheep and 1308 images of 9 Saanen goats, obtained from farms in El Assafia and Bennasser Benchohra in Laghouat, Algeria. These images, captured with high-quality phone cameras, serve as the foundational dataset for training the AI model. To ensure accuracy and reliability, the dataset includes a diverse range of images, covering variations in lighting conditions, angles, and facial expressions.

Example from our dataset

Figure 3.1: dataset.

3.3.2 Data labelling

Creating high-quality labels is crucial for training a robust sheep face detection model. Use tools like LabelImg or Labelbox to define and accurately draw labels, such as bounding boxes, on your images. Ensure accuracy through double-checking and data augmentation. Labelled data includes images with coordinates in either (x, y, w, h) or (xmin, ymin, xmax, ymax) formats. This study manually labelled images using LabelImg, saving outputs as XML files in PASCAL VOC format for easy conversion to TF-records, compatible with TensorFlow 2. Well-labelled data is key to a high-performing model.

Figure 3.2: Labelled image.


```
<?xml version="1.0"?>
- <annotation>
    <folder>0-2122</folder>
    <filename>39.jpg</filename>
    <path>C:\Users\SISSI\Documents\PFE\Sheep pics\0-2122\39.jpg</path>
  - <source>
      <database>Unknown</database>
    </source>
  - <size>
      <width>1800</width>
      <height>4000</height>
      <depth>3</depth>
    </size>
    <segmented>0</segmented>
  - <object>
      <name>sheep</name>
      <pose>Unspecified</pose>
      <truncated>0</truncated>
      <difficult>0</difficult>
    - <bndbox>
        <xmin>764</xmin>
        <ymin>1235</ymin>
        <xmax>1083</xmax>
        <ymax>1758</ymax>
      </bndbox>
    </object>
</annotation>
```


Figure 3.3: Resulted XML file .

3.3.3 Object detection models

SSD MobileNet V2 320x320

SSD MobileNet V2 320x320 is a specific pre-trained version of the SSD MobileNet model architecture designed for real-time object detection on devices with limited computational resources [34].

- SSD : is a real-time object detection framework that combines the speed of single-shot detection and accuracy of multi-stage detection [34].
- MobileNet V2 : is a lightweight convolutional neural network architecture designed for mobile and embedded devices.
- 320x320 refers to the input image size that the model expects.
- SSD MobileNet V2 320x320 : provides a good balance between speed and accuracy, making it suitable for real-time applications on resource-constrained devices.

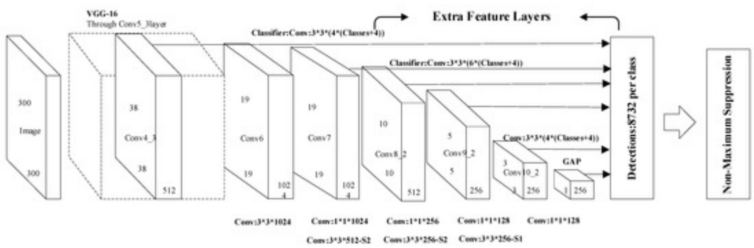

Figure 3.4: Architecture of SSD MobileNet model [35].

SSD MobileNet V2 320x320 is a streamlined object detection model that combines the Single Shot MultiBox Detector (SSD) framework with the MobileNet V2 backbone to achieve efficient and accurate detection on mobile and embedded devices. The model takes a 320x320 pixel input image and processes it through the MobileNet V2 feature extractor, which is optimized for low computational cost and high efficiency using depthwise separable convolutions and inverted residuals. The extracted features are then fed into multiple convolutional layers that predict bounding boxes and class scores for objects at different scales, enabling the detection of objects of varying sizes. By leveraging the lightweight MobileNet V2 architecture and the efficient SSD detection framework, SSD MobileNet V2 320x320 achieves a balance between speed and accuracy, making it suitable for real-time applications on resource-constrained devices.

CenterNet Hourglass 512x512

Hourglass 512x512 is a pre-trained deep learning model designed for object detection. Core Concept:

- **CenterNet:** This framework differs from traditional object detection approaches. Instead of predicting bounding boxes, it directly predicts the center point, size, and class of an object in an image. This simplification leads to faster inference speeds.

- **Hourglass Network:** This is a specific neural network architecture known for its ability to capture long-range dependencies within an image, potentially improving object detection accuracy.

- **512x512:** Images fed into the model are resized to 512x512 pixels. This resolution offers a balance between speed and accuracy compared to lower resolutions typically used for mobile applications (e.g., 320x320 in SSD MobileNet V2)

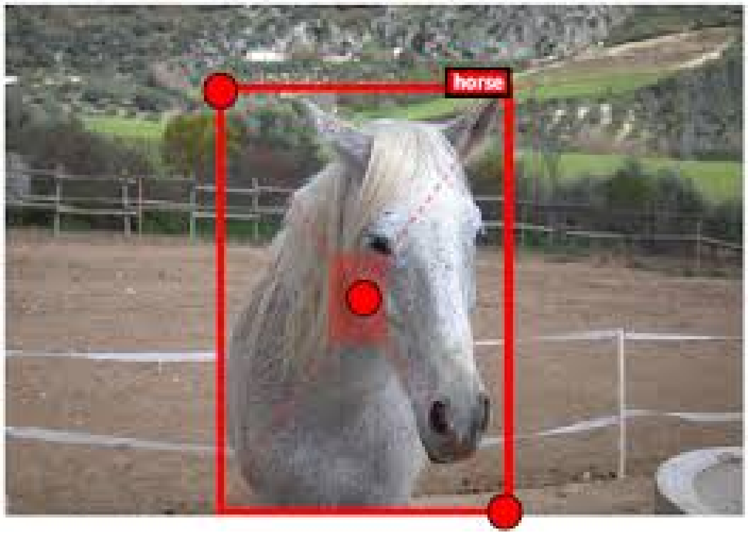

Figure 3.5: CenterNet Keypoint Triplets for Object Detection [36].

CenterNet Hourglass 512x512 is a deep learning model designed for object detection and keypoint estimation. It utilizes a fully convolutional architecture based on the hourglass network, which captures and processes information at multiple scales to detect objects with high accuracy. The model takes a 512x512 pixel input image and outputs a heatmap where each peak corresponds to the center of an object. CenterNet employs keypoint detection to locate object centers and other relevant points, simplifying the detection pipeline by eliminating the need for anchor boxes or region proposal networks. This approach enhances both the efficiency and precision of detecting various objects within an image.

3.3.4 Pre-processing

Partitioning the Dataset

The dataset underwent division into training and testing sets, maintaining 90% of the dataset for the training and 10% for the testing.

Creating the Label Map

TensorFlow 2 necessitates a label map, which associates each label with an integer value. This map serves both the training and detection processes.

```
item{
    id: 1
    name: 'sheep'
}
```

Figure 3.6: Label map.

Creating TensorFlow 2 Records

Following annotation generation and dataset partitioning, the next step involves converting annotations into TFRecord format. This conversion enhances data management within TensorFlow 2.

3.3.5 Model Training

This section elaborates on the procedural steps involves in training and hyperparameter tuning of two models utilizing SSD object detection architectures: MobileNet 320x320 and CenterNet Hourglass 512x512 .

1. **Downloading Pre-Trained Models:** Initially, the latest pre-trained network for the desired model needs to be acquired. This is conveniently achieved by accessing the TensorFlow 2 Detection Model Zoo and selecting the desired model.

2. **Configuring Training Pipeline:** With the pre-trained model downloaded and extracted, the training pipeline file requires meticulous configuration:

    - num classes: This parameter specifies the number of classes in the training dataset, typically set to 2 for binary classification tasks.

- batch size: Determining the number of images fed into the model per step, the batch size is selected considering hardware constraints, often trading training time for computational power. In this instance, a batch size of 8 was chosen.
- fine tune checkpoint: Utilized for transfer learning, the fine-tune checkpoint involves applying a pre-trained model's base weights to our classification problem, mitigating overfitting in scenarios with limited data. The path to the "checkpoint/ckpt-0" in the extracted model is specified.
- fine tune checkpoint type: This setting dictates how variables are restored from the pretrained model, offering options like "classification," "detection," or "full." Here, "detection" is selected.
- label map path: Pointing to the "label_map. pbtxt" file containing class IDs and names, this path is set under both "train_input_reader" and "eval_input_reader."
- input path: To ensure proper data input, the input_path directs to the paths of train.records/test.records. "train.records" and "test.records" are assigned under "train_input_reader" and "eval_input_reader," respectively.

3. Training the Model: Initiating the training process entails specifying the path for saving the trained model and the 'pipeline.
config' file. This step is crucial for initiating the training job and ensuring the model's progress and performance.

4. Model Evaluation : During the training process, periodic creation of checkpoint files occurs within the training directory, representing snapshots of the model at specific steps. Upon generation of a set of new checkpoint files, the evaluation process commences, utilizing these files to assess the model's efficacy in object detection across the test dataset. Evaluation results are consolidated into metrics, providing insights into the model's performance. These metrics can be monitored over time through TensorBoard, facilitating comprehensive analysis of the model's evolution and effectiveness.

5. Training Progress using TensorBoard : TensorBoard, an extension created by the TensorFlow team, simplifies the complexity of neural networks by allowing continuous monitoring and visualization of various training and evaluation metrics. As a result, it generates graphs such as Accuracy, Error, and weight distributions. By running the command tensorboard –logdir=../mymodel, a local URL is provided to view the dashboard in the browser, enabling real-time insights into the model's performance.

3.4 Results

After completing the training process, we obtained the following results :

3.4.1 The latest checkpoint

In our scenario, the latest checkpoint yields the following results:

For the SSD MobileNet V2 320x320

```
Average Precision  (AP) @[ IoU=0.50:0.95 | area=   all | maxDets=100 ] = 0.607
 Average Precision  (AP) @[ IoU=0.50      | area=   all | maxDets=100 ] = 0.813
 Average Precision  (AP) @[ IoU=0.75      | area=   all | maxDets=100 ] = 0.695
 Average Precision  (AP) @[ IoU=0.50:0.95 | area= small | maxDets=100 ] = -1.000
 Average Precision  (AP) @[ IoU=0.50:0.95 | area=medium | maxDets=100 ] = 0.000
 Average Precision  (AP) @[ IoU=0.50:0.95 | area= large | maxDets=100 ] = 0.611
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets=  1 ] = 0.457
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets= 10 ] = 0.646
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets=100 ] = 0.668
 Average Recall     (AR) @[ IoU=0.50:0.95 | area= small | maxDets=100 ] = -1.000
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=medium | maxDets=100 ] = 0.000
 Average Recall     (AR) @[ IoU=0.50:0.95 | area= large | maxDets=100 ] = 0.671
```

Figure 3.7: The latest checkpoint of SSD MobileNet V2 320x320

The latest SSD MobileNet V2 320x320 checkpoint shows mixed results: a decent overall AP of 0.607, high precision at IoU=0.50 (0.813), and good precision at IoU=0.75 (0.695). It struggles with small and medium objects (AP of -1.000 and 0.000, respectively) but performs well on large objects (AP of 0.611). Overall AR is 0.646, indicating good recall for large objects (0.671) but poor for smaller ones. Improvements are needed for detecting small and medium-sized objects.

For the CenterNet Hourglass 512x512

```
Average Precision  (AP) @[ IoU=0.50:0.95 | area=   all | maxDets=100 ] = 0.676
Average Precision  (AP) @[ IoU=0.50      | area=   all | maxDets=100 ] = 0.842
Average Precision  (AP) @[ IoU=0.75      | area=   all | maxDets=100 ] = 0.744
Average Precision  (AP) @[ IoU=0.50:0.95 | area= small | maxDets=100 ] = -1.000
Average Precision  (AP) @[ IoU=0.50:0.95 | area=medium | maxDets=100 ] = 0.000
Average Precision  (AP) @[ IoU=0.50:0.95 | area= large | maxDets=100 ] = 0.680
Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets=  1 ] = 0.503
Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets= 10 ] = 0.702
Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets=100 ] = 0.707
Average Recall     (AR) @[ IoU=0.50:0.95 | area= small | maxDets=100 ] = -1.000
Average Recall     (AR) @[ IoU=0.50:0.95 | area=medium | maxDets=100 ] = 0.000
Average Recall     (AR) @[ IoU=0.50:0.95 | area= large | maxDets=100 ] = 0.711
```

Figure 3.8: The latest checkpoint of CenterNet Hourglass 512x512

The latest checkpoint of CenterNet Hourglass 512x512 shows strong performance overall, with an AP of 0.676 across IoU thresholds from 0.50 to 0.95, and high precision at IoU=0.50 (0.842) and IoU=0.75 (0.744). However, it fails to detect small objects (AP = -1.000) and has no detections for medium objects (AP = 0.000). It performs well on large objects with an AP of 0.680. The AR metrics reflect a similar trend, with a reasonable overall AR of 0.503, good recall for larger objects (0.711), but poor recall for small and medium objects. The model excels with larger objects but requires improvements for smaller and medium-sized objects.

3.4.2 Tensorboard Graphs

After the training process, we obtained these graphs from TensorBoard.

The model SSD MobileNet V2 320x320

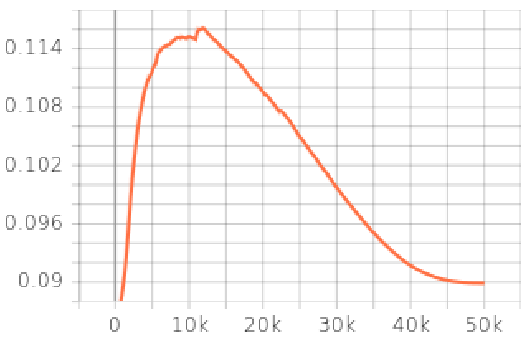

Figure 3.9: Loss regularization.

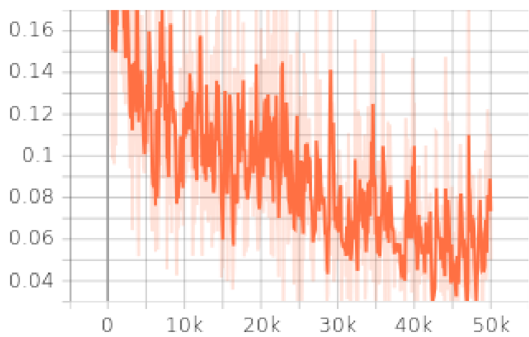

Figure 3.10: loss localization.

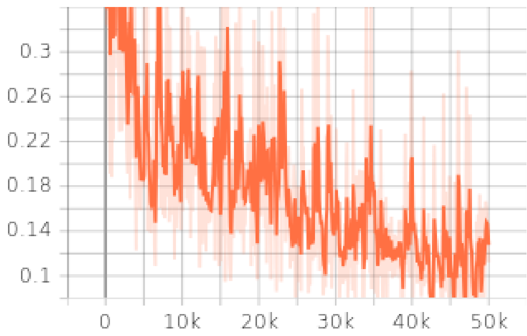

Figure 3.11: Loss classification.

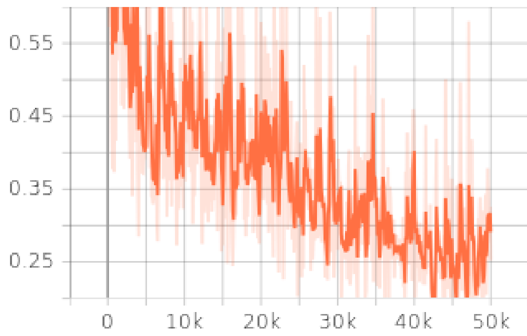

Figure 3.12: Loss total loss.

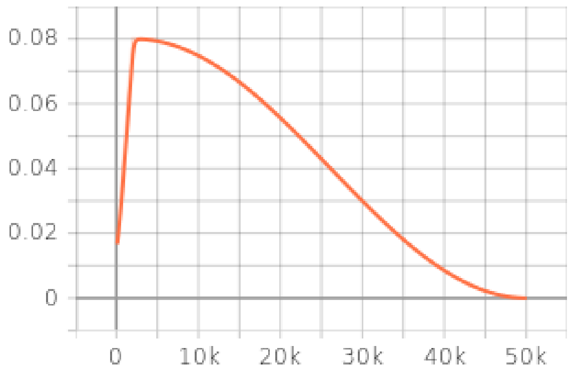

Figure 3.13: Learning rate.

The model CenterNet Hourglass 512x512

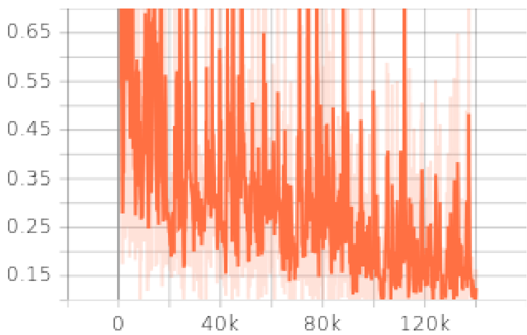

Figure 3.14: Loss box scale.

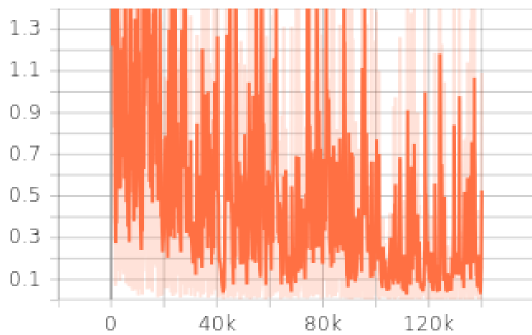

Figure 3.15: Loss object center.

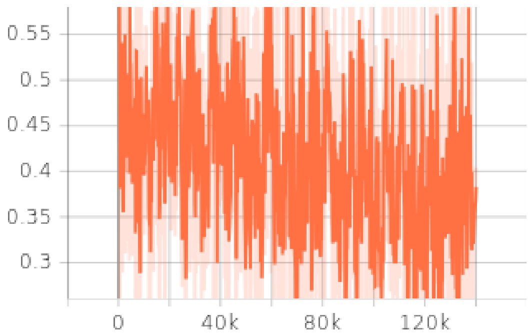

Figure 3.16: Loss box offset.

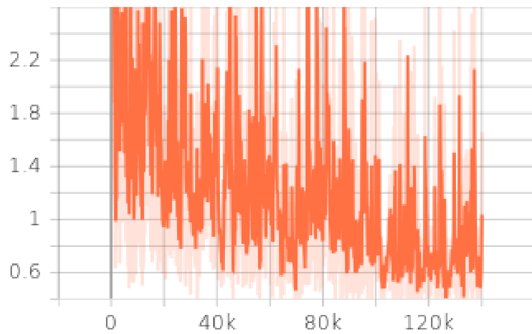

Figure 3.17: Loss total loss.

The SSD-MobileNet took over 11h 54min 27s for the training, meanwhile Center-NetHourglass took over 2d 2h 31 min 16s. However, SSD MobileNet accuracy is less than CenterNet-HourGlass.

3.4.3 The accuracy

Table 3.1: The accuracy table

	SSD MobileNet320x320	CenterNet-HourGlass512x512
DetectionBoxes Precision/mAP	0.695	0.744

3.4.4 Video detection

The model SSD MobileNet V2 320x320

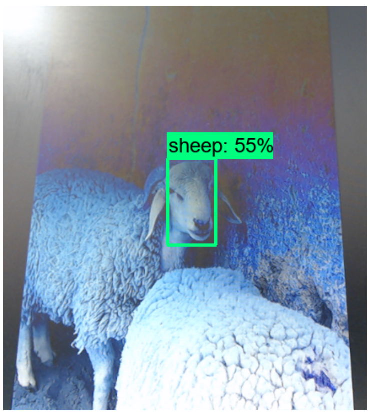

Figure 3.18: Detection 1

Figure 3.19: Detection 2

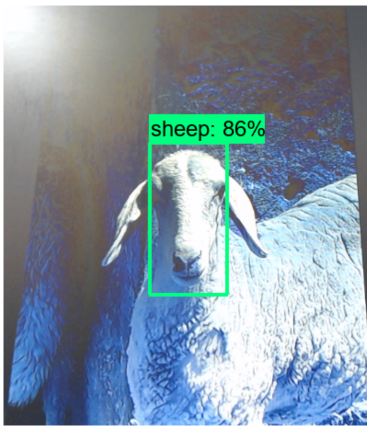

Figure 3.20: Detection 3

- **In Detection 1,2 and 3 we used some images form the test images that we took from our Dataset.**

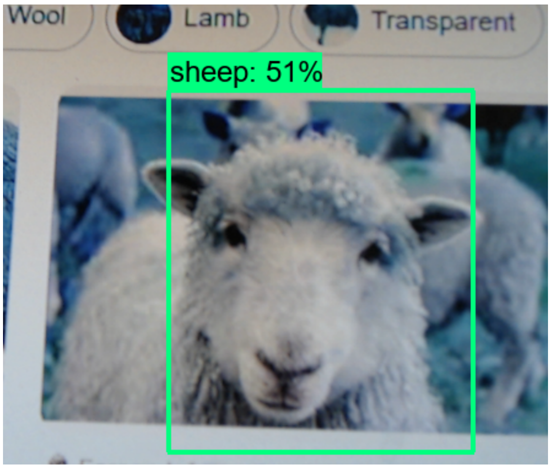

Figure 3.21: Detection 4

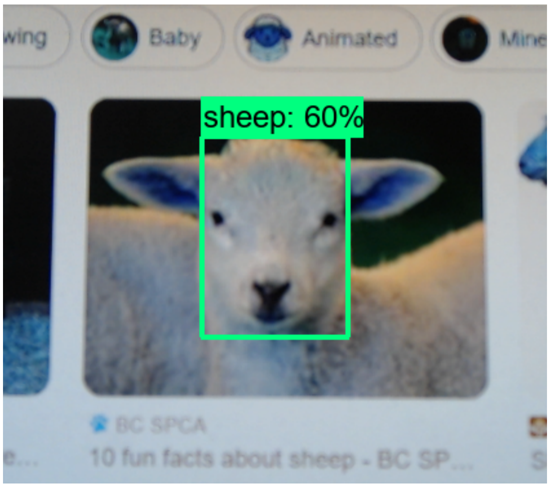

Figure 3.22: Detection 5

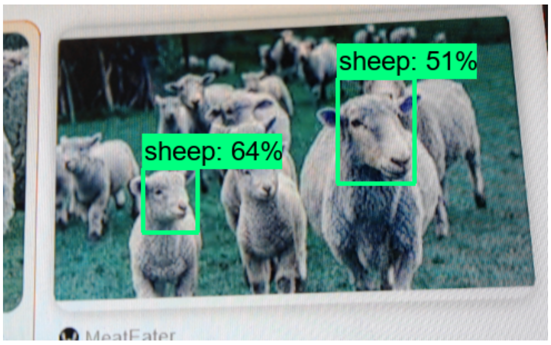

Figure 3.23: Detection 6

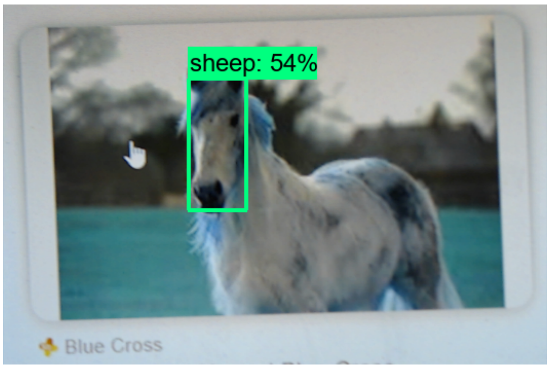

Figure 3.24: Detection 7

- **In Detection 4,5 and 6 we used some sheep images from the internet and SSD MobileNet model has detected all of them.**
- **In Detection 6 the image contains more than one sheep but the SSD MobileNet model has detected only two sheeps.**
- **In Detection 7 is a false detection, the image contains a horse but the model has detected it as a sheep.**

The model CenterNet Hourglass 512x512

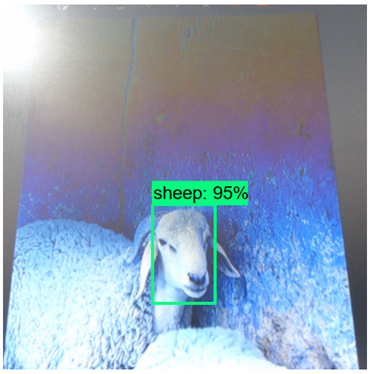

Figure 3.25: Detection 1

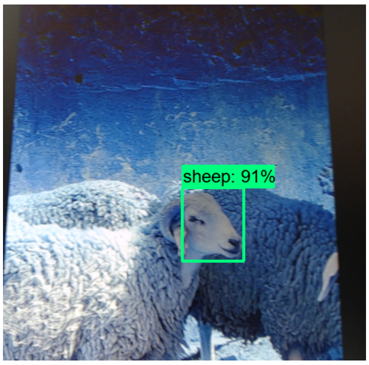

Figure 3.26: Detection 2

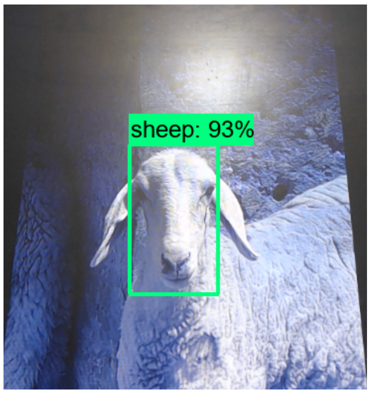

Figure 3.27: Detection 3

- **In Detection 1, 2 and 3 we used some images form the test images that we took from our Dataset, and the model CenterNet Hourglass512x512 detected them.**

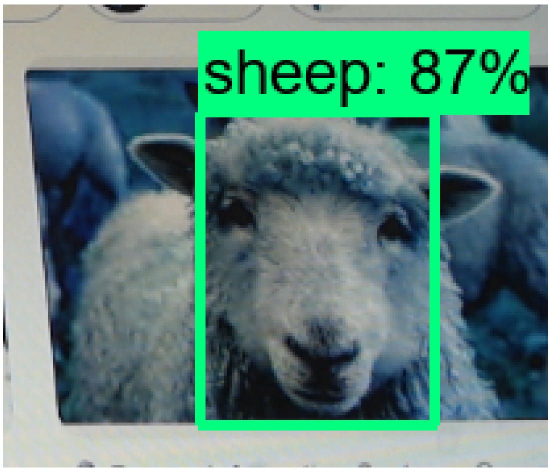

Figure 3.28: Detection 4

Figure 3.29: Detection 5

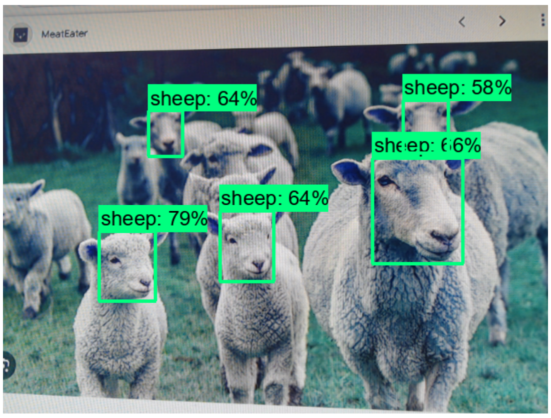

Figure 3.30: Detection 6

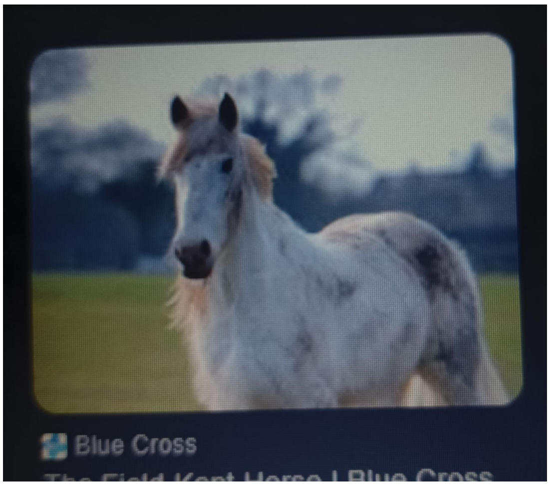

Figure 3.31: Detection 7

- **In Detection 4,5 and 6 we used some sheep images from the internet and CenterNet Hourglass512x512 model has detected all of them.**
- **In Detection 6 the image contains more than one sheep and the CenterNet Hourglass512x512 model has detected almost all of the sheeps.**
- **In Detection 7, the image contains a horse but the model has not detected it as sheep.**

Based on this comparative study, we conclude that while CenterNet achieves superior performance compared to SSD MobileNet, it requires more processing time.

3.5 Conclusion

This chapter we explored deep learning for sheep face detection. We implemented and evaluated two models SSD MobileNet320x320 and CenterNet Hourglass 512×512 using Python and TensorFlow to identify the optimal approach for accurate sheep face detection in images.

GENERAL CONCLUSION

In conclusion, this thesis demonstrates the significant potential of applying deep learning techniques to sheep face detection, showcasing both SSD MobileNet 320x320 and CenterNet Hourglass 512x512 models. The results highlight the balance between accuracy and processing speed, with CenterNet Hourglass 512x512 achieving higher accuracy [74.4%] and SSD MobileNet 320x320 providing faster processing times [69.5%].

The findings underscore the transformative impact of AI on agricultural practices, particularly in livestock management. By enhancing the ability to accurately and efficiently identify individual sheep, these technologies can lead to better health monitoring, breeding management, and overall improved operational efficiency. Additionally, implementing these technologies is crucial for the security and protection of sheep, helping to prevent theft and ensure the well-being of each animal.

This study contributes to the growing body of knowledge in agricultural AI, offering practical solutions for the industry. Future research could further optimize these models and explore additional AI techniques to expand the applicability and effectiveness of sheep face detection systems. As AI continues to evolve, its integration into agriculture promises to drive significant advancements, fostering more intelligent and sustainable farming practices.

REFERENCES

[1] OpenAI, "ChatGPT." GitHub repository, june 2024.

[2] QTravel, "Machine learning: Definitions, types, and practical applications." https://www.qtravel.ai/blog/machine-learning-definitions-types-and-practical-applications/, may 2024.

[3] JavaTpoint, "Types of machine learning." https://www.javatpoint.com/types-of-machine-learning, may 2024.

[4] GeeksforGeeks, "Types of machine learning." https://www.geeksforgeeks.org/types-of-machine-learning/, may 2024.

[5] "Traditional computing process," may 2024.

[6] Mert Sukru Pehlivan, "Exploring the k-nearest neighbors algorithm in machine learning." https://medium.com/@mertsukrupehlivan/exploring-the-k-nearest-neighbors-algorithm-in-machine-learning-e3b90a3c9e0d, may 2024.

[7] KrishnaJadhavResearch, "Machine learning in insurance: Eight applications." https://medium.com/@KrishnaJadhavResearch/machine-learning-in-insurance-eight-applications-e845957be923, may 2024.

[8] "The venn diagram of the artificial intelligence hierarchic terminology." https://www.researchgate.net/figure/The-Venn-diagram-of-the-artificial-intelligence-hierarchic-terminology_fig2_379045983, may 2024.

[9] GeeksforGeeks, “Difference between ann and bnn.” https://www.geeksforgeeks.org/difference-between-ann-and-bnn/, may 2024.

[10] “Deep learning artificial neural network machine learning.” https://www.cleanpng.com/png-deep-learning-artificial-neural-network-machine-le-1356670/, may 2024.

[11] IBM, “Deep learning.” https://www.ibm.com/topics/deep-learning, may 2024.

[12] NVIDIA Developer, “Deep learning frameworks.” https://developer.nvidia.com/deep-learning-frameworks, may 2024.

[13] Weida Gang, “Learning PyTorch: The Basic Program Structure.” Medium, may 2024.

[14] Actu IA, “Keras - Le guide ultime pour débutants.” Actu IA, may 2024.

[15] Sfeir, “Est-ce que vous me connaissez ? Je suis JAX!.” Sfeir, june 2024.

[16] OpenBee, “IA et fraudes à la facturation.” OpenBee, june 2024.

[17] GeeksforGeeks, “Introduction to convolution neural network.” https://www.geeksforgeeks.org/introduction-convolution-neural-network/, may 2024.

[18] ResearchGate, “Generic CNN structure and operation.” ResearchGate, june 2024.

[19] SuperAnnotate, “Object Detection with Deep Learning.” SuperAnnotate Blog, june 2024.

[20] Sharath Hebbar, “Object Detection from Scratch.” Medium, june 2024.

[21] SuperAnnotate, “Intersection over Union for Object Detection.” SuperAnnotate Blog, june 2024.

[22] Label Your Data, “Object Detection Metrics.” Label Your Data, june 2024.

[23] GeeksforGeeks, “Confusion Matrix in Machine Learning.” GeeksforGeeks, june 2024.

[24] Towards Data Science, “Baffling Concept of True Positive and True Negative.” Towards Data Science, june 2024.

[25] H. Zhang, L. Wang, Z. Zhang, and Y. Chen, “Automatic license plate recognition based on deep learning,” *Journal of Computer and Communications*, vol. 9, pp. 21–35, june 2024.

[26] "Artificial intelligence: A multi-dimensional approach to smart farming," june 2024.

[27] "Artificial intelligence enables mobile soil analysis for sustainable agriculture," june 2024.

[28] "New cow monitoring system promises artificial intelligence in the dairy parlor," *Hoard's Dairyman*, june 2024.

[29] "Chicken boy robot broiler puppet for poultry farms," *Robots Science*, june 2024.

[30] Y. Z. W. P. Z. Z. Zhenchao Chen, Lin Ma, "Sheep face detection based on an improved retinaface algorithm," *Animals*, vol. 13, p. 2458, june 2024.

[31] S. Song, T. Liu, H. Wang, B. Hasi, C. Yuan, F. Gao, and H. Shi, "Using pruning-based yolov3 deep learning algorithm for accurate detection of sheep face," *Animals*, vol. 12, p. 1465, june 2024.

[32] Authors, "Goat and Sheep Face Detection Algorithm Based on YOLOv5-Swin-Transformer-BiFPN." SSRN Working Paper, june 2024. SSRN Working Paper.

[33] TensorFlow, "TensorFlow 2 Detection Model Zoo." GitHub. Accessed: 2024-06-13.

[34] Yonas650, "Real-Time-Object-Detection." GitHub. Accessed: 2024-06-13.

[35] ResearchGate, "Architecture of SSD MobileNet model." ResearchGate. Accessed: 2024-06-13.

[36] N. F. Tech, "Review: CenterNet: Keypoint Triplets for Object Detection." Medium. Accessed: 2024-06-13.