RÉPUBLIQUE ALGÉRIENNE DÉMOCRATIQUE ET POPULAIRE
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

UNIVERSITE ECHAHID HAMMA LAKHDAR - EL OUED
FACULTÉ DES SCIENCES EXACTES
Département D'Informatique

Mémoire de Fin D'étude
Présenté pour l'obtention du Diplôme de

MASTER ACADEMIQUE

Domaine : **Mathématique et Informatique**
Filière : **Informatique**
Spécialité : **Systèmes Distribués et Intelligence Artificielle**

Présenté par :

- **HALOUAT** Hakim Seif Eddine
- **YOUMBAI** Mouawia

Thème

Edge-AI Approach for Scorpion Detection

Soutenue le **22 - juin - 2021** Devant le jury :

Dr. EL-ZAIZ Fouzi	MCA	Président
Dr. YAAQOUB Mohamed Amin	MAA	Rapporteur
Dr. KHOULLADI Nedjoua Houda	MAA	Encadreur

Année Universitaire: 2020/2021

PEOPLE'S DEMOCRATIC REPUBLIC OF ALGERIA
Ministry of Higher Education and Scientific Research

ECHAHID HAMMA LAKHDAR UNIVERSITY - EL OUED
FACULTY OF EXACT SCIENCES
Computer Science department

End of Study Memory
Presented for the Diploma of

ACADEMIC MASTER

Domain: **Mathematics and Computer Science**
Spinneret: **Computer Science**
Specialty: **Artificial Intelligence and Distributed Systems**

Presented by:

- **HALOUAT** Hakim Seif Eddine
- **YOUMBAI** Mouawia

THEME

Edge-AI Approach for Scorpion Detection

In front of the committee composed of:

Dr. EL-ZAIZ Fouzi MCA President

Dr. YAAQOUB Mohamed Amin MAA Examiner

Dr. KHOULLADI Nedjoua Houda MAA Supervisor

Academic year: 2020/2021

Acknowledge

My thanks go first to Almighty GOD who gave us patience, courage and health to complete this work. Research work can by no means be brought to light without the help of people around us. For this, we would like to thank our Supervisor Mrs.KHOLLADI Nedjoua Houda for her commitment and persistence. Similar gratitude addressed to our teachers for their academic generosity throughout five years of devotion.

Finally, our thanks go to the members of the jury for accepting to discuss and evaluate our work.

Dedication

It gives me great pleasure to reserve this page as a witness of gratitude to my beloved ones.
To those who breathed into me courage and perseverance, who instilled in me a taste for effort, who rained me with hope and blessed me with their prayers: my dear parents.
Whatever I do or say, I cannot thank you properly.
To my dear sisters and my dear twin brother who have always supported and motivated me to go forward.
May God fill you with health, happiness and success.

Hakim Seif Eddine

Dedication

It gives me great pleasure to reserve this page as a witness of gratitude to my beloved ones.
To those who breathed into me courage and perseverance, who instilled in me a taste for effort, who rained me with hope and blessed me with their prayers: my dear parents.
Whatever I do or say, I cannot thank you properly.
To my dear brothers and sisters who have always supported and motivated me to go forward.
May God fill you with health, happiness and success.

Mouawia

Abstract

Regardless of the final target aim with the use of NVIDIA® toolkit, which offers available resources with typically restricted embedded models. NVIDIA® Jetson Nano™ Toolkit permits users carry on incredible new capacities to millions of small, power-efficient AI systems. It launches new fields of IOT embedded applications, comprising entry-level Network Video Recorders (NVRs), robotics and home robots, and intelligent world using entirely analysis and analytics

Thus, a proposed architecture and pipeline design is highly required to clarify all steps implemented. Meanwhile integration of Deep learning based AI approaches into embedded projects.

The proposed project covers the Scorpion detection AI workflow, incorporating sensors, cables, transistors, NVIDIA® Jetson Nano™ Toolkit, buzzer and RaspberryPI4 camera with using inference employing transfer learning techniques as Deep Learning based AI approaches to the Edge-AI used technology. Here, a mobile application is demonstrated for an object detection task running on Jetson Nano™ technology using TensorRT.

The mobile application receives signal from Jetson Nano device, a user interface is implemented on mobile, coded by using Flutter and Dart, which is Google's UI toolkit for constructing stunning, natively running applications for mobile computing.

The goal of the realized system has the ability of safer life and helps reducing scorpion stings by tracking scorpions and alerting space holders accurately. Additional option added is the ability to alert with buzzer once scorpion detected.

This project can be deployed in the known research field Smart City and its Applications, in the future this project can widely explore for many reasons and requirements.

Keywords: Scorpion, Artificial Intelligence, Edge AI, Mobile Computing, Object Recognition, Scorpion Detection, Neural Network, Flutter.

الملخص

بغض النظر عن الهدف النهائي مع استخدام مجموعة أدواتNVIDIA ، التي تقدم الموارد المتاحة مع نماذج مدمجة محدودة عادة .وتسمح مجموعة أدوات Nano™ NVIDIA Jetson للمستخدمين بالاستفادة من قدرات جديدة لا تصدق لملايين الأنظمة الصغيرة ذات الكفاءة في استخدام الطاقة .وهو يطلق حقول جديدة من التطبيقات المدمجة في نظام المعلومات والاتصالات (IOT) ، وتشمل مسجلات الفيديو الشبكية المبتدئة (NVR) ، والروبوتات المنزلية ، والعالم الذكي باستخدام التحليلات والتحليلات بالكامل.

ومن ثم ، فإن هناك حاجة شديدة إلى تصميم مقترح للهياكل الأساسية وخطوط الأنابيب لتوضيح جميع الخطوات المنفذة .وفي الوقت نفسه، إدماج نهج منظمة العفو الدولية القائمة على التعلم العميق في المشاريع المدمجة.

ويغطي المشروع المقترح تدفق العمل الخاص بالكشف عن العقرب، الذي يضم أجهزة استشعار، وكابلات، وترانزستورات، ومجموعة أدوات NVIDIA Jetson Nano™ وجرس، وكاميرا RaspberryPI4 ، مع استخدام الاستدلال باستخدام تقنيات تعلم نقل مع نهج منظمة العفو الدولية القائمة على التعلم العميق للتكنولوجيا المستخدمة في Edge-AI وهنا يظهر تطبيق متنقل لمهمة كشف الأجسام التي تجري على تكنولوجيا Jetson Nano™باستخدام TensorRT .

ويستقبل التطبيق المحمول إشارة من جهاز (Jetson Nano)، وتنفذ واجهة مستخدم على الهاتف المحمول، ويرمز لها باستخدام Flutter and Dart ، وهي مجموعة أدوات UI الخاصة بشركة Google لبناء التطبيقات المذهلة التي تدير بشكل ناقص للحساب المحمول.

ويتمتع هدف النظام المحقق بالقدرة على الحياة الآمنة ويساعد على الحد من لسعات العقارب عن طريق تتبع العقارب وتنبيه أصحاب الفضاء بدقة .الخيار الإضافي المضاف هو القدرة على التنبيه مع الجرس بمجرد اكتشاف العقرب. ويمكن نشر هذا المشروع في ميدان البحوث المعروف "المدينة الذكية وتطبيقاتها"، ويمكن لهذا المشروع في المستقبل أن يستكشف على نطاق واسع لأسباب ومتطلبات عديدة.

الكلمات الرئيسية: العقرب، الذكاء الاصطناعي، الحوسبة النقالة، التعرف على الأجسام، الكشف عن العقارب، الشبكة العصبية، كاشف الطلقة الواحدة، فلاتر.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

GLOSSARY

AI: Artificial Intelligence.

CNN: Convolutional Neural Network.

DL: Deep Learning.

HCA: Hierarchical Cluster Analysis.

LD50: Lethal Dose.

LLE: Locally Linear Embedding.

MAP: Mean Average Precision.

ML: Machine Learning.

NN: Neural Network.

NVR: Network Video Recorder.

PCA: Principal Component Analysis.

R-CNN: Regions with CNN Features

SSD: Single Shot Detector.

SVM: Support Vector Machines.

T-SNE: T-distributed Stochastic Neighbor Embedding.

YOLO: You Only see it Once.

GENERAL INTRODUCTION

El-Oued and Sahara are the largest area in Algeria. Although all dissert region has the danger of scorpions sting, there are relatively safe cases as there are death cases. Most of scorpions stingers usually cause only local pain, minimal edema, lymphangitis with satellite lymphadenopathy, an increase in local skin temperature, and tenderness around the injury [16].

Onset symptoms include immediate pain and sometimes numbness or tingling in the affected area. There is usually no edema and few skin signs. The most common serious symptoms in children include: {Agitation, Muscle spasms, Abnormal and random movements of the head, neck and eyes, anxiety and restlessness, Sialorrhea and sweating}. In adults, tachycardia, hypertension, hyperventilation, weakness, muscle spasms and fasciculations may be prominent. Breathing difficulties are rare, whether in children or adults. Stings of such species like C. sculpturatus have been shown to be fatal in children <6 years of age and in sensitized individuals.

A significant exception in Algeria dissert, a variety of scorpion species (Centruroides sculpturatus, also known as C.exilicauda), which occurs in different regions: El-Oued, Ghardaya, Bordj-Baji Mokhtar, Addrar, Naama,etc. This species is poisonous and causes more serious lesions and disorders, which is classified as the most dangerous scorpion in the world.

Despite the decline in the number of deaths, from 106 cases in 1991 to 46 cases in 2019, the number of scorpion envenomations "continues to increase year by year, from nearly 23,000 cases in the 1990s to nearly 45,000 cases in recent years, according to statistics provided by the Department of Prevention and Health Promotion. The Algerian ministry reports 42 wilayas exposed in 2018 to scorpion envenomations, including at least 15 having recorded cases of death [31].

Geographic data shows that the scorpion endemic to the Sahara now spreads quite easily in different areas with transport and economic business, but also thanks to its ability to adapt. Now, the scorpion is spotted even in some parts of the North as Sidi Belabbs [16].
The increase in cases of scorpion envenomations "is due to the non-respect of the cleanliness of the environment and the proliferation of precarious dwellings and household refuse, which constitute favorable conditions for the multiplication of the nests of scorpions, explained to APS Dr. Mohamed Lamine Saïdani, member of the National Commission for the Prevention of Scorpion stings and Head of the Anti-Scorpion Serum Production Unit at the Institut Pasteur [31].
Scorpion envenomation is a public health issue in Algeria according to WHO. In fact, each year there are on average 25,000: to 50,000 bites/ year, 100 to 200 deaths/ year [32].
With the progress of AI technology, several systems and applications are proposed for many reasons as obstacle detection, fraud detection, anomaly detection, threat detection and danger detection. Toward smart life vs smart world with exploring Internet technology, variety of projects have adopted in advanced domains. In fact, the proposed project tackle as first position to recognize the scorpion as the wanted object using object detection
The proposed system based Jeston-Nano processing, has the following characteristics: integrated model of object detection in Edge AI device, integrated a notification and alert system as mobile computing, high-tech automatic for processing both edge computing with different sensors belonging to IOT field, real-time perception as data driven, high efficiency and safety achieved real-time responding. In the whole real time detection system, the identification of dangerous object has always been the core process. From many objects, the proposed AI based model which is implemented by Jeston-Nano, can monitor the defined dangerous object in real time , the consequences will be unimaginable in any axes: healthcare, biology, ecology… etc. using various I/O support. Through an AI SDK, NVIDIA® Jetson Nano™ Toolkit offers GPU also provides fast AI deployment with AI function processing.
In order to fine tune the convolution layer and the full connection layer a transfer learning method is used, as a result, it can be addressed to multiplicity of research axes as danger detection or risk detection. Even though this entirely work determines the role of deep learning in this research area, it does not point out some issues that necessitate to be measured in practical application, such as risk detection. The scorpions tracking algorithm and object detections AI models based on Edge AI devices implemented by NVIDIA® Jetson Nano™ Toolkit has a persuasive and real leading project which offers

prominent impact on daily life. The operating technique that is a deep learning detection network based on improved SSD-MobileNet method, which permits the detector to cover high accuracy detection regarding the condition of real-time streaming; with different circumstances, the detection results based on tracking the scorpion is realized, and the bounding box has been designed to visualize visibly and precisely the scorpion, here the detection and tracking task in same angle in real time of sequence of video [33].

This work aims to track a dangerous animal, to manifest saving people life, monitor a real-time AI experience and processing, sending data{notifications} from devices to mobile application, scalability resulted from distributing AI services and make it easier to propagate IOT ecosystem.

This dissertation investigates a serious problem, regarding to explain thoroughly and concisely this problematic, it has outlined the structure of the proposed work on the following chapters:

Chapter 1: STATE OF THE ART

This chapter has clarified a start point and the main raison that conduct this project to be place, the recent research field literally attempt to solicit the problematic which is detecting scorpions using AI models, this chapter has exposed a set of fields with showing detail of corresponding domains in order to accomplish the aim of project, these fields are Artificial Intelligence and its introduction, Edge AI, Computer Vision and Machine Learning and Deep Learning Components.

Chapter 2: MOTIVATION OF THE STUDY

Related work appear in this field that lead us to show the strength of given contributions.

This chapter has focalized to present several point initiating by Motivation of this work and projection their interest in daily-life, then a highlighting State of the art of Scorpion in Algeria, as result this project is belonging to Danger Detection using Artificial Intelligence, trying to show main features.

The main realized projects in it. First, a brief summary presented for Internet of Things, then succinct review of Edge Computing. Next, a detailed paragraphs for Computer Vision. Passing by a concise Feed Forward Neural Networks technique specially those used and implemented on the Edge-AI with illustrating main algorithms in Deep Learning. Finally, an Evaluation Metrics is computed to assess the proposed model.

Chapter 3: SUPPORTED TOOLS IN DEVELOPEMENT (NVIDIA® Jetson Nano™)

This section introduce this tools supported by NVIDIA® Jetson Nano™, then identify The Jetson Nano, explains well all steps:

- Setting Up and Running.
- Required Components.
- Items for Getting Started
- Write Image to the microSD Card
- Initial Setup with Display Attached Connecting the Camera to the Jetson Nano, by the end clarifies Setting up Developing Environment.

Chapter 4: OBJECT DETECTION WITH JETSON NANO

This chapter demonstrate Object Detection Top-performing deep learning models for Object Detection, MOBILENET-SSD, R-CNN and YOLO in The Edge-AI.

Chapter 5: REALIZATION OF THE PROJECT

This section situate the Setting Up of Jetson Inference Library then give the way of Installing PyTorch at the edge to compile this Project. A required step are needed for running this project starting with Collecting Detection Dataset, then exposing the Training The Model, all this part should be done on Jetson-Nano.

A second half of project is the mobile computing task, it should display Alerts and Notifications using mobile application and a connection realized between Edge AI and mobile application.

Finally attempt to show result and explain metrics that perform it out.

CHAPTER 1

STATE OF THE ART

1.1 Introduction

The emergence of the Visual Automation and recognition as a major axis for digital data: text, image, speech and video. Computer vision leverages high position for using machine learning and deep learning algorithms to optimize and scale up applications and their use in any field, also to perform a competitive advantages.

In this chapter, the dissertation has tackled several important titles as: Artificial Intelligence, Edge AI, Computer Vision, Machine learning and Deep learning.

1.2 Artificial Intelligence

1.2.1 Definition of Artificial Intelligence

To begin, let's define what Artificial Intelligence (AI) means, The Oxford English dictionary defines AI as "The theory and development of computer systems able to perform tasks normally requiring human intelligence, such as visual perception, speech recognition, decision-making, and translation between languages.''.

Therefore, Artificial intelligence (AI) is a branch of computer engineering designed to create machines that act like humans. Although AI has come this far in recent years, it is still missing some essential parts of human behavior, such as emotional behavior, object identification, and smooth human interaction. Such would mean computers or machines with programs capable of performance similar to human intellectual capacity or even amplified by technology.

These machines can:

- Reason
- Process large amounts of data
- Discern patterns undetectable by a human eye
- Understand and analyze these models
- Interacting with People
- Learn gradually
- Continuously improve performance

"Artificial intelligence" thus covers an enormous subject permanently changing. Moreover, with fast progress since 1950, the foundation year of AI [1].

1.2.2 1950-The Founding Year of Artificial Intelligence

It was 1950 when Alan Turing, British mathematician, and cryptologist, published Computing Machinery and Intelligence and introduced the Turing test, which is the purpose of which (still used today) is to determine whether a computer is behaving like a human. The test consists of allowing one person to speak blindly to another human and a computer, and then the person must determine which of their interlocutors is a machine [2].

In 2017, artificial intelligence took a decisive step by successfully identifying words in a conversation and also in human beings and opening up new perspectives for speech recognition and machine translation in everyday life [1].

In January 2018, Artificial Intelligence surpassed humans in various reading and comprehension exercises in Stanford University's famous reading test. That will enable AI to interact with humans easily in the future and to provide them with information in a more natural way [1].

1.2.3 Artificial Intelligence Performance

According to Harry Shum, the past Executive Vice President of Artificial Intelligence & Research at Microsoft "AI only works when there is a large amount of data, exceptional computing power, exclusively thanks to the cloud and revolutionary algorithms based on deep learning."

Today, AI used in a variety of areas such as:

- Health care and Medical Imaging Analysis
- Games
- Farming and Agriculture
- Sports analytics and Activities
- Robotics
- Machine Translation

- Transport
- Public administration
- Web search
- Finance

1.3 Edge AI

1.3.1 Definition of Edge Artificial Intelligence

Edge Artificial intelligence is a system that uses machine-learning algorithms to process and generate data (data from sensors or signals). Edge moves AI to where it is required, on the device itself, not in the cloud. The device does not need to be connected to the Internet to process data, provide real-time information and decision making in few milliseconds.
To use Edge AI, you need a device with a microprocessor and sensors [3].

1.3.2 The existence of Edge Artificial Intelligence

Some famous real-time instances use complex algorithms to process the data right on your device instead of sending it to the cloud for results.

Examples: [3]

- The iPhone registers and recognizes your face to unlock the phone in milliseconds
- Google Maps, which trigger an alert for bad traffic.
- Autonomous vehicles: an intelligent car camera can locally recognize vehicles, traffic signs, pedestrians, streets, and objects and only send the information, which is necessary for the realization of autonomous driving.
- Similar concepts apply to robots and drones.

1.3.3 Functions of Edge Artificial Intelligence

In a typical machine-learning environment, we first train a model for a specific problem on a suitable data set. Fundamentally, training a model means programming it to find patterns in the training data set and then evaluating it in the test data set to test its performance on other invisible data sets. The type should be similar to the sort of training data set.

After training the model, if the model is in the cloud, it must first transmit the necessary data (input data) from the terminal machine, and then use it to predict the results, which requires a reliable connection, and due to a large amount of data, the transmission speed may be tardy, or even impossible in some cases. If the data transfer fails, the model is not useless.

In data transmission, we still have to deal with delays. Unlike traditional configurations where inference runs on a cloud-computing platform, the Edge AI model runs on peripheral devices without having to be permanently connected to the output. This process transfers the decision-making power to the device and enables it. To achieve autonomy, this is called Edge's artificial intelligence [4].

1.3.4 The importance of Edge Artificial Intelligence

Edge Artificial Intelligence has many advantages: [5]

- It supports real-time operations, including data creation, decision making, and processes involving milliseconds.
- Real-time performance is essential for self-driving cars, robotics, and many other fields.
- For portable devices, Reducing power consumption and therefore extending battery life is very important.
- As the data transmission reduced, the data transmission cost decreased. By processing data locally, you can avoid the problem of transferring and storing large amounts of data in the cloud, making you vulnerable to attacks Data protection perspective.

1.4 Computer Vision

1.4.1 Definition of Computer Vision

Computer imaginative and prescient is the sphere of Artificial Intelligence (AI), which permits computer systems and structures to extract significant information from digital images, videos, and other visual materials, and take action or make recommendations based on this information. It focuses on creating digital systems that can process, analyze, and understand visual data (images or videos) similar to humans.

The computer vision concept focuses on teaching computers to process images at the pixel level and find out. Technically, machines try to use specific software algorithms to extract visual information, process it, and interpret the results. Artificial Intelligence enables computers to think, although computer vision permits them to see, observe, and understand [6].

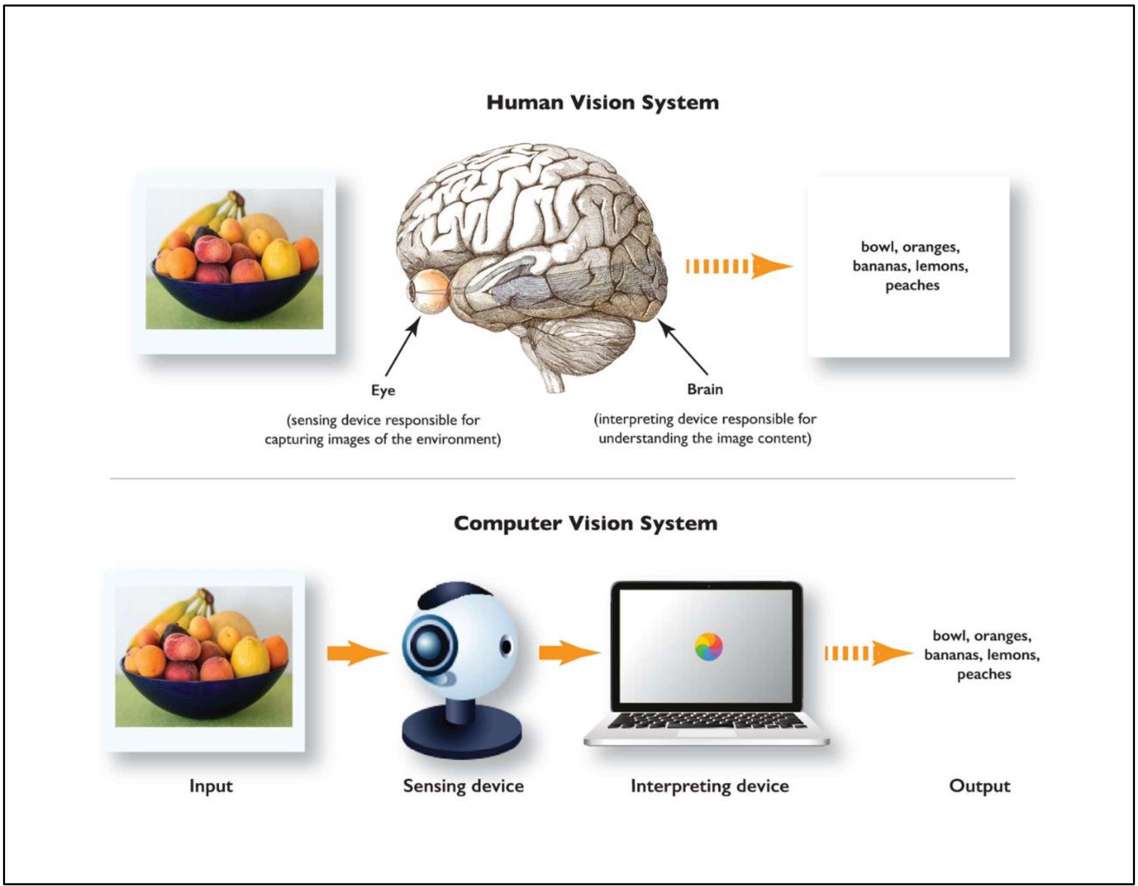

Figure 1. 1: Human vision system vs computer vision system.

The following are some common tasks used by computer vision systems:

- **Object classification:** the wide class that the item lies in.
- **Object localization:** Object position.
- **Object detection:** Object and its position in the image.
- **Object segmentation:** The pixel items that make up the object.

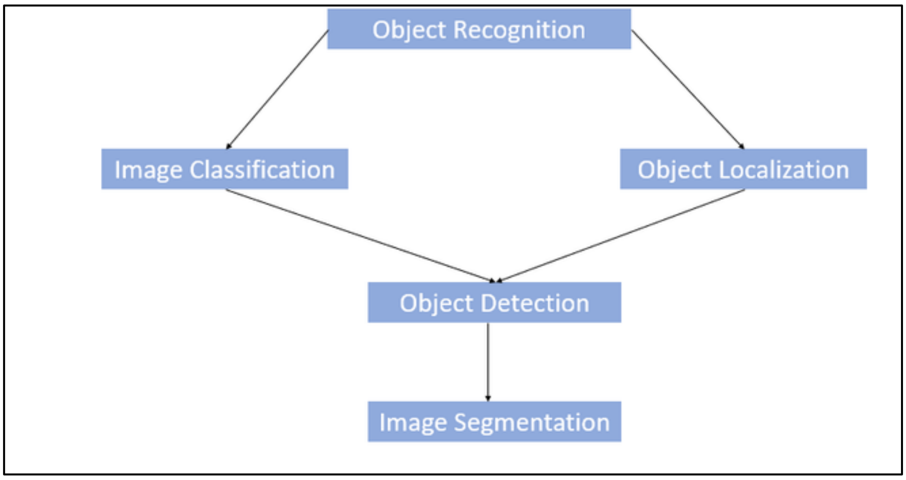

Figure 1. 2: Computer vision common tasks.

Note: Object recognition is a term to describe a collection of computer vision tasks that comprise identifying objects in digital photographs.

1.4.2 Computer Vision History

Computer vision is not a new technology. The first experiments in computer vision began in the 1950s when this invention was used to interpret typed and handwritten text. During that period, computer vision analysis technology was quite simple.

However, operators who must manually provide sample data for analysis require hard work. As you may have guessed, it is exhausting to process large amounts of data manually. Not good enough, so the margin of error for this analysis is very high.

Today, we do not lack computing power. The combination of cloud computing and robotic algorithms can help us solve the most complex problems. However, it is not just the combination of new hardware and complex algorithms that control the development of computer vision technology (we will introduce it in the next part). A sizable amount of publicly available visual data we generate every day is the driving force behind the latest developments in this technology [6].

According to Forbes, there are more than 3 billion images online every day, and these data are used to train computer vision systems.

1.4.3 Operations of computer vision

Computer vision technology mimics the way the human brain works. But how does our brain solve the problem of recognizing visual objects? A common assumption is that our brains rely on patterns to decode individual objects. This concept is used to create a computer vision system. The current use is based on pattern recognition. We train the computer with a large amount of visual data: the computer processes the image, marks objects on the image, and finds patterns on these objects. For example, when we send a million pictures of apples, the computer will analyze them, identify patterns similar to all apples, and create an "apple" model at the end of the process. Therefore, every time a picture is sent to them, the computer can accurately determine whether a particular image is a flower [6].

Example:

Below is a photo of Abraham Lincoln. The brightness of each pixel in this image is represented by an 8-bit number between 0 (black) and 255 (white). When inputting a picture, the program will display these numbers. As the input of the computer vision algorithm, the algorithm is responsible for further analysis and decision-making [6].

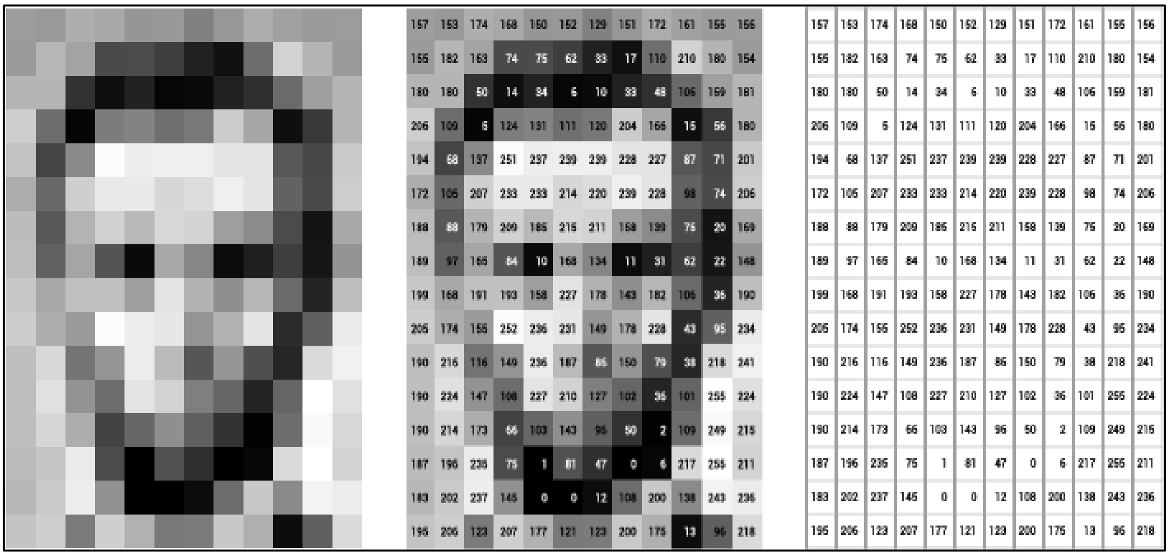

Figure 1. 3: Example how computer vision work.

1.4.4 The fields of computer vision technology application

Some people think that computer vision is the distant future of design. That is not right. Computer vision has been integrated into many areas of our lives.

Here are some well-known examples of how we use this technology today: [6]

- Content organization
- Facial recognition
- Augmented reality
- Self-driving cars

1.5 Machine Learning and Deep Learning Components

Machine Learning is a branch of Artificial Intelligence, which includes Deep Learning, as shown in Figure (1.4).

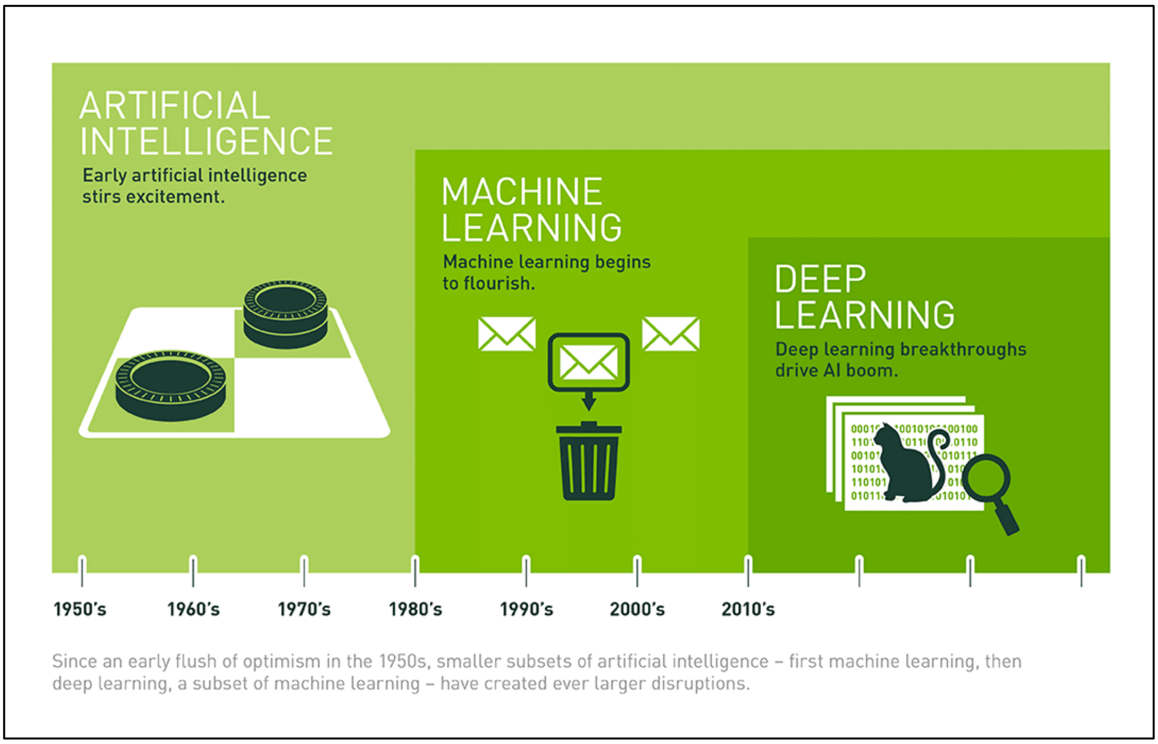

Figure 1. 4: AI, ML and DL

1.5.1 Machine Learning

Machine learning (ML) is a field of study that uses the statistics and computer science principles, to create statistical models, used to perform major tasks like predictions and inference. These models are groups of mathematical relationships through the inputs and outputs of a given system. The learning procedure is the process of guessing the models parameters such that the model can achieve the stated task [7].

1.5.1.1 Fields of machine learning

Machine learning is a common buzzword in Computer vision, Medical, E-commerce, Finance, Transportation and various sectors.

There are many applications of these algorithms in different fields: [8]

Field	Applications
Computer vision	Face Recognition, Object Recognition, Robotics, Self-Driving cars, etc...
Medical	Drug Development, Biomedical Data Analysis, Neurosciences, etc.
Finance and E-commerce	Analytics, Insurance, Fraud detection, Financial Forecasting, Spam Email, etc.
Geography	Earthquake Detection, Weather Detection (google weather), etc.
Signal	Language NLP, Speech Processing, etc.

Table 1. 1: The Usage of Machine Learning

1.5.1.2 Types of machine learning

There some variations of how to define the types of Machine Learning Algorithms but commonly they can be divided into categories according to their purpose and the main categories are the following: [7]

- Supervised learning.
- Unsupervised Learning.
- Semi-supervised Learning.
- Reinforcement Learning.

1.5.1.2.1 Supervised learning

The supervised machine learning algorithms are those algorithms which needs external assistance. The input dataset is divided into train and test dataset. The train dataset has output variable which needs to be predicted or classified. All algorithms learn some kind of patterns from the training dataset and apply them to the test dataset for prediction or classification [9].

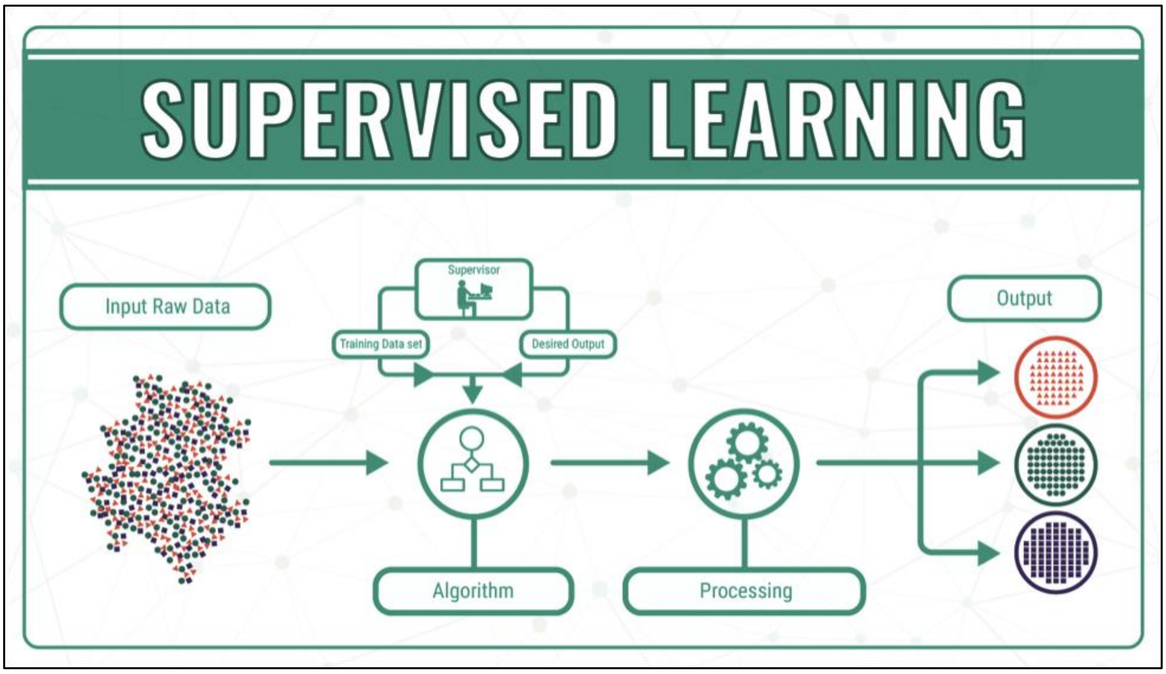

Figure 1. 5: Overview of supervised learning.

List of common algorithms

- Nearest Neighbor
- Naive Bayes

- Decision Trees
- Linear Regression
- Logistic Regression
- Support Vector Machines (SVM)
- Neural Networks

1.5.1.2.2 Unsupervised learning

The unsupervised learning algorithms learns few features from the data. When new data is presented, it uses the earlier learned features to identify the class of the data. It is mostly used for feature reduction and clustering [9].

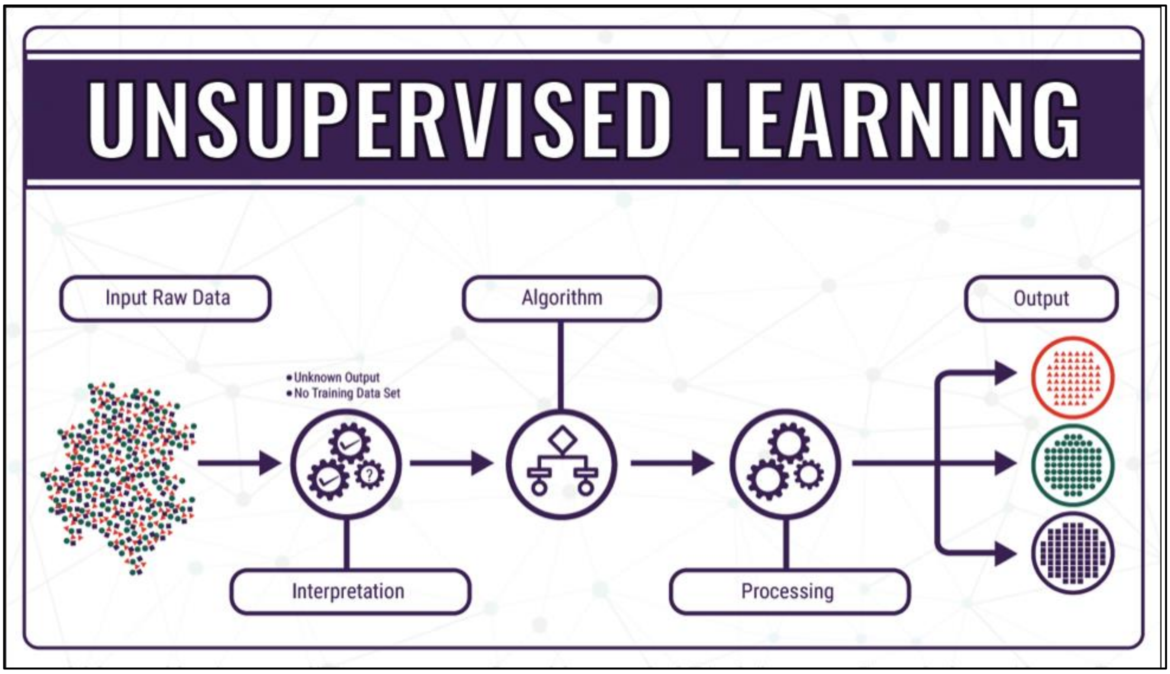

Figure 1. 6: Overview of Unsupervised learning.

Here are some of the most important unsupervised learning algorithms: [9]

- Clustering
  - K-Means.
  - Hierarchical Cluster Analysis (HCA).
  - Expectation Maximization.
- Visualization and dimensionality reduction
  - Principal Component Analysis (PCA).
  - Kernel PCA.

  - Locally Linear Embedding (LLE).
  - T-distributed Stochastic Neighbor Embedding (t-SNE).
- Association rule learning
  - Apriori.
  - Eclat.

1.5.1.2.3 Semi-Supervised learning

Semi-supervised Learning algorithms is a technique, which combines the power of both supervised and unsupervised learning. It can be fruit-full in those areas of machine learning and data mining where the unlabeled data is already present and getting the labeled data is a tedious process [9].

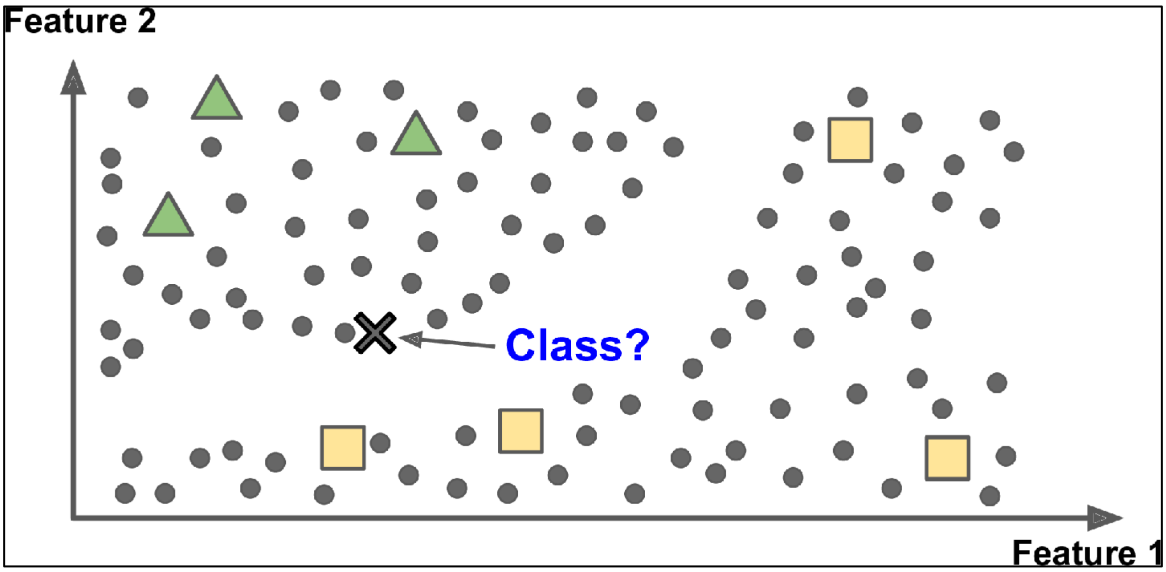

Figure 1. 7: Example of Semi-supervised learning.

1.5.1.3 Techniques of machine learning

Machine learning has various techniques; the above once are the most important: [10]

- Transfer Learning
- Regression
- Classification
- Clustering
- Dimensionality Reduction
- Natural Language Processing

Transfer learning is a machine learning technique in which a model developed for one task is reused as the starting point for a model in a second related task.

It is a popular deep learning method that uses pre-trained models as the starting point for computer vision and natural language processing problems because the development of neural network models requires a lot of computing and time resources [11].

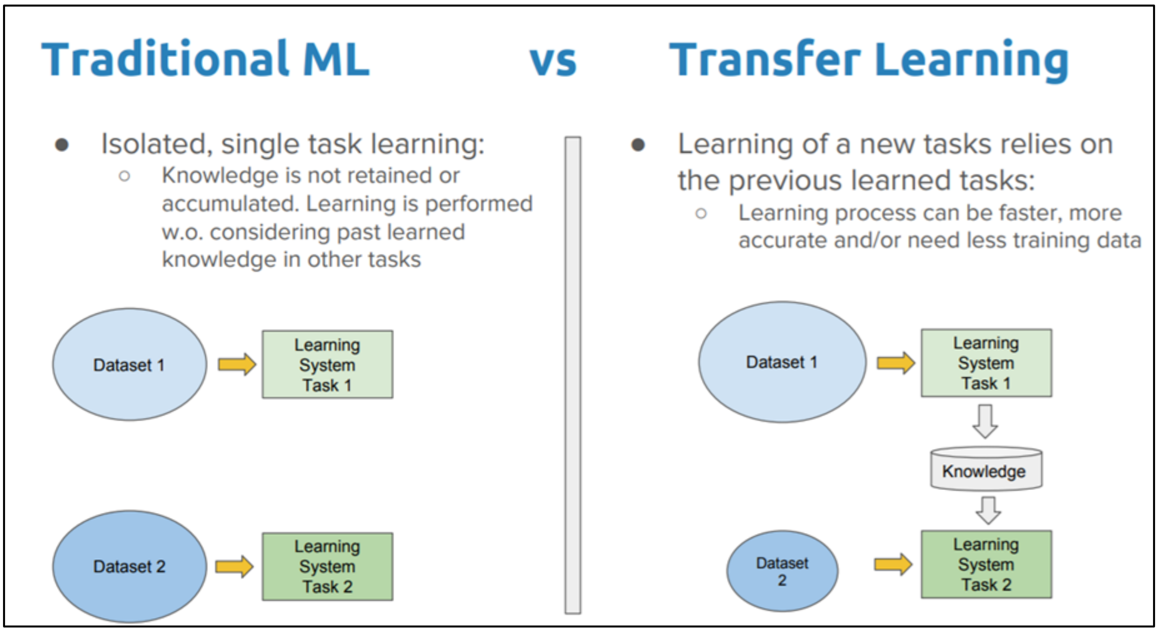

Figure 1. 8: Traditional ML vs Transfer Learning.

Note: In this project, we use this technique due to time constraints and lack of performance, also because there is no available dataset for the scorpion. In addition, one of the main advantages of this technique it is allowed us to work with minimal datasets. Since the collected datasets were small because of the seriousness of approaching it and dealing with it or photographing it.

1.5.1.4 Big challenge in the machine learning process

- Insufficient Quantity of Training Data
- Non representative Training Data
- Poor-Quality Data
- Irrelevant Features
- Overfitting the Training Data
- Under fitting the Training Data [12].

1.5.2 Deep Learning

Deep learning (DL) is a set of learning approaches trying to model data with multifaceted architectures uniting different non-linear transformations. The basic brick of deep learning is the neural network that is combined to form the deep neural network. These techniques have allowed significant progress in the fields of image processing and sound, including facial recognition, computer vision, speech recognition, automated language processing, text classification (for example spam recognition).

There exist some types of architectures for neural networks:

- The multilayer perceptron's, that are the oldest and simplest ones.
- the Convolutional Neural Networks (CNN), particularly adapted for image processing.

The recurrent neural networks, used for sequential data such as text or times series [13].

1.5.2.1 Artificial Neural Networks

An Artificial Neural network are very powerful brain-inspired computational models. Which have been active in several areas such as medicine, economics, engineering, computing and many others [14].

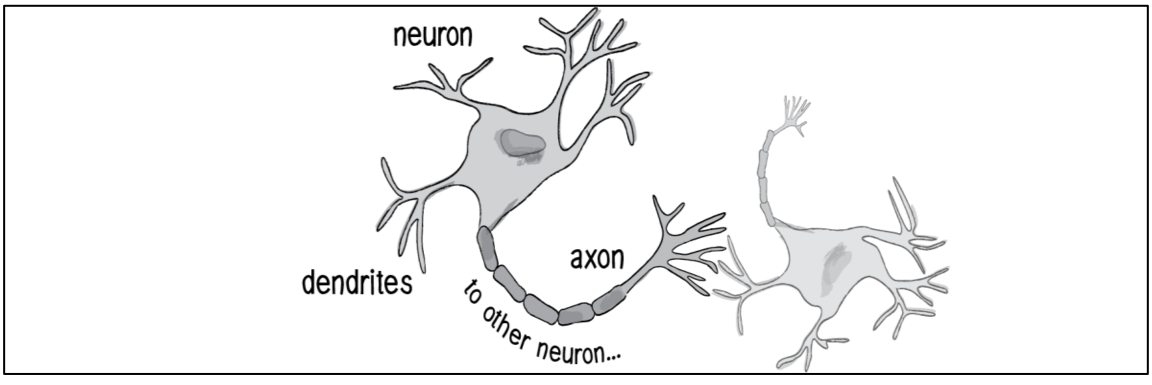

Figure 1. 9: Natural neuron.

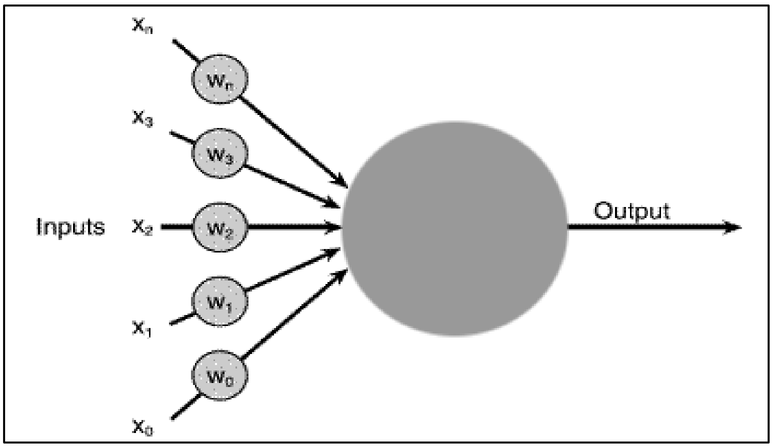

Figure 1. 10: Artificial neuron.

An artificial neural network is built on the optimization theory. It is a computational model inspired in the working of the human brain. It is consisting by a set of artificial neurons that are interconnected with other neuron these neurons rely on weights of the neural network. As The word network in Neural Network mentions to the interconnection between neurons present in several layers of a system. These weights represent the connections between the neurons, which define the impact of one neuron on another [14].

1.5.2.2 Convolutional Neural Network

Convolutional Neural Network (CNN) is a special case of the neural network described above. A CNN contains of one or more convolutional layers, often with a subsampling layer, which are followed by one or more fully connected layers as in a standard neural network. The strategy of a CNN is interested by the discovery of a visual mechanism, the visual cortex, in the brain. The visual cortex covers many cells that are responsible for detecting light in small, overlapping sub-regions of the visual field, which are called receptive fields. These cells work as local filters over the input space, and the more intricate cells have higher receptive fields. The convolution layer in a CNN do the function that is achieved by the cells in the visual cortex.

Each feature of a layer collects inputs from a set of features placed in a small neighborhood in the preceding layer called a local receptive field. With local receptive fields, features can extract elementary visual features, such as end-points, corners, oriented edges, etc., which are then combined by the higher layers.

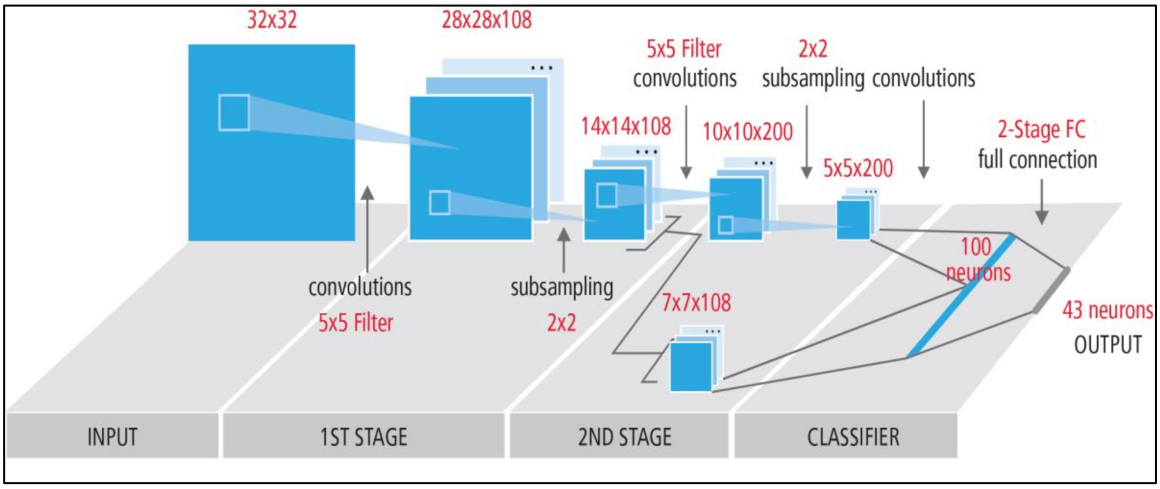

Figure 1. 11: Typical block diagram of a CNN.

1.6 Conclusion

To prove the efficiency of Artificial Intelligence , Edge AI and Computer Vision by using Machine learning and Deep learning, many application has been developed, our aim work in this dissertation is to develop an Edge computing mobile application for automatic recognition of scorpions, this project has been elaborated for many reasons, the main one preventing and saving people life.

Next chapter will talk about Scorpion in Algeria and describing its danger on people life.

CHAPTER 2

MOTIVATION OF THE STUDY

2.1 Introduction

Scorpion has been considered as danger on people life, one of this work motivation is to develop mobile application for alerting holders about this danger.

This chapter will highlight this work motivation, then provide a recap about Scorpion in Algeria and finally show some Danger Detection application using Artificial Intelligence.

2.2 Motivation

As we are living in south area, many risks are facing us from several dangers: scorpion sting, snakes sting, rats… with an issue of spacious area that is too difficult to be controlled by a human eye, we tend to motivate us for designing application that used Edge-AI to mitigate this mortal danger.

The sting is known as the bite or sting that is used in some types of insects which resulting very dangerous state because of its secretion of the toxin that penetrates into it due to its secretion; this secretion would be spreadable into arteries or blood veins that cause bed side effects that may be output life threatening.

2.3 State of the art of Scorpion in Algeria

2.3.1 Etiology of Scorpions

The causes of scorpion poisoning are mostly accidental. Scorpios are terrifying creatures that only sting when they feel threatened, in trouble, or worried (for example, when they are sitting or stepping on them). Curious people are at risk due to the increasing interaction with Scorpion.

The average lethal 50 doses (LD50) of various scorpions' venom in mg/k of subcutaneous injection to mice and according to David Cheng in his article Scorpion Envenomation: Background, Pathophysiology, Etiology, the territorial distribution is listed below (unfortunately, humans are more sensitive than mice):

- Leiurus quinquestriatus (Middle East) - 0.25 mg/kg

- Androctonus crassicauda (Saudi Arabia) - 0.08-0.5 mg/kg
- Centruroides noxius (Mexico) - 0.26 mg/kg
- Androctonus mauritanicus (North Africa) - 0.32 mg/kg
- Centruroides santa maria (Central America) - 0.39 mg/kg
- Tityus serrulatus (Brazil) - 0.43 mg/kg
- Buthus occitanus (North Africa) - 0.9 mg/kg
- Centruroides sculpturatus (Southwest United States) - 1.12 mg/kg
- Mesobuthus eupeus (Iran) - 1.45 mg/kg

Generally, most lethal scorpions have an LD50 below 1.5 mg/kg.

Under the electrical stimulation of certain species, the average production of each scorpion on the venom gland is as follows: [15]

- Tityus species - 0.39-0.62 mg
- L quinquestriatus - 0.62 mg
- Buthus species - 0.38-1.5 mg
- Milking the venom gland produces approximately a 4-fold increase in yield amount compared to electrical excitation.

2.3.2 Pathophysiology of Scorpions

In general, adult scorpions do not exceed 25cm, especially those from North Africa, ranging between 2 and 12cm. The body of a scorpion is divided into three parts: the prosoma or cephalothorax or head, the mesosoma or abdomen, the metasoma or post abdomen or tail (Figure 2. 1).

The first two parts form a group commonly referred to as the trunk [16].

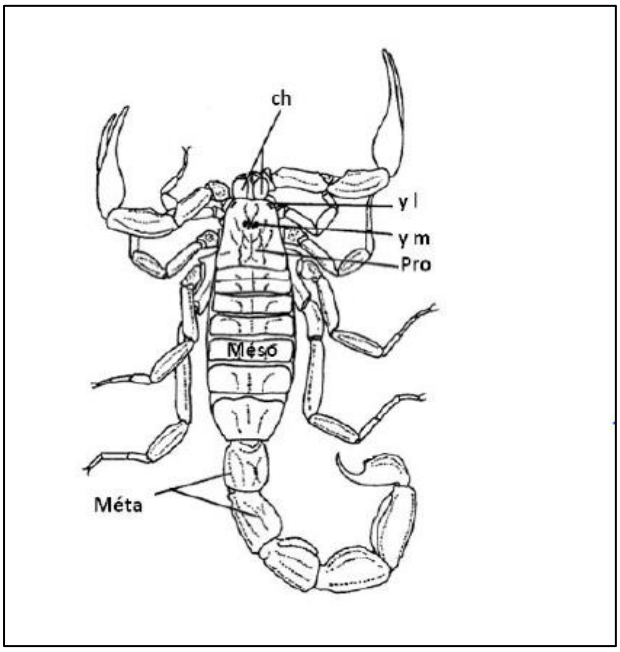

Figure 2. 1: Dorsal view of the scorpion.

ch: chelicerae.

Pro: prosoma or cephalothorax.

Meso: mesosoma or pre-abdomen.

Meta: metasoma or tail.

yl: side eyes.

ym: median eyes.

The venom gland is located on the tail near the tip of the stinger and consists of two types of tall cylindrical cells: one secretes toxins and the other secretes mucus. The intensity of the poison varies from species to species: some will cause mild flu, while others will kill within an hour. In addition, there are differences between species in the volume, flow rate, and duration of the poison. When the poison is deposited in the vein structure, it usually spreads rapidly in the tissue. Intravenous administration after injection can cause symptoms as few as 4 to 7 minutes, maximum tissue concentration after 30 minutes, and a total urine half-life of 4.2 to 13.4 hours. The higher the concentration of the poison in the blood, the faster the systemic symptoms appear [15].

2.3.3 Statistics of Scorpion Mortality cases in Algeria

Statistics show that Algeria has about 50,000 bites each year, resulting in 50 to 100 deaths. Dr. Farida Allian, a spokeswoman for the Pasteur Institute in Algeria, reported that a total of 20 willayas represent "the great danger of scorpion bites." In 2002, a poison extraction center was established in the following cities: one of Masila and the other is in El oued. As a result, other centers began to operate: in Gardaya, as each regionresponsible for their populations, the doctor Lakhdar Garyan of the Guardia Center is considered a member of the National Committee for Controlling Scorpion Poison. He said that the poisoning caused by the arachnid family "is not linked to size and color", despite, from an overall Of 40 scorpions living in Algeria, there are only 3 cases, which might cause mortal factor to human.

Given that the most affected category is that infants, who are the first victims, it is unreasonable to record only deaths caused by scorpions. We should treat this as an urgent investigation and carefully examine thoroughly the situation to attempt to reach efficient and permanent solutions for the proposed issue to help public health [17].

2.4 Danger Detection using Artificial Intelligence

For example, we have: [18]

1. Hermes — Wildfire Detection using NVIDIA Jetson and Ryze Tello.
2. Drowsines, blind spot, emotions and attention monitor for driving or handling heavy machinery. Also detects objects at the blind spot via Computer Vision powered by Pytorch and the Jetson Nano.
3. Smart (Ai) Pothole Detector (TensorRT/Jetson Nano).

2.5 Conclusion

A variety of applications has been developed in order to detect danger, this application has been deployed on phone to benefit from its portability, the use of embedded devices as well has impacted on their scalability and powerful.

Next chapter will illustrate Edge-AI Toolkit as using approach for this work.

CHAPTER 3

SUPPORTED TOOLS IN DEVELOPEMENT (NVIDIA® Jetson Nano)

3.1 Introduction

NVIDIA® Jetson Nano™ allows you to bring incredible new features to millions of small, energy-efficient AI systems. It opens up a new world of embedded IOT applications, including entry-level network video recorders (NVR), home robots, and smart gateways with complete analysis capabilities.

Jetson Nano is also the perfect tool to start learning artificial intelligence and robotics in a real-world environment. It provides projects ready to try and is supported by an active and enthusiastic developer community.

3.2 The Jetson Nano

The NVIDIA® Jetson Nano™ Developer Kit is a powerful small computer that allows you to run multiple neural networks in parallel for applications such as image classification, object detection, segmentation, and speech processing. All of this is in an easy-to-use platform that runs as low as 5 watts [19].

Figure 3. 1: Jetson Nano developer kit.

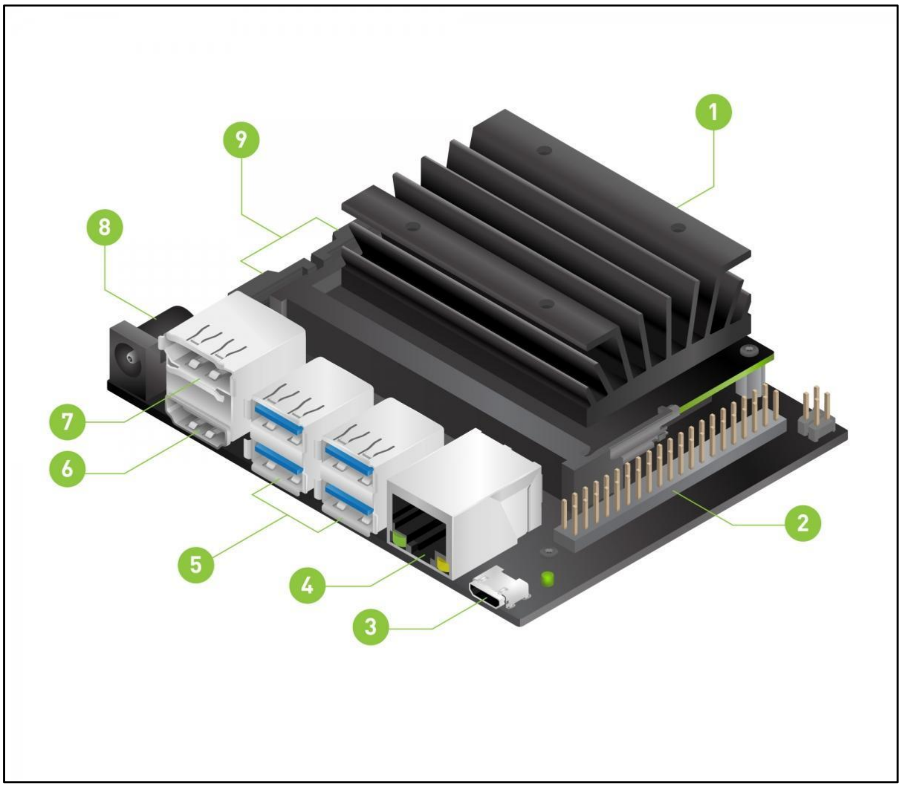

Figure 3. 2: Parts of Jetson Nano.

1. microSD card slot for main storage
2. 40-pin expansion header
3. Micro-USB port for 5V power input, or for Device Mode
4. Gigabit Ethernet port
5. USB 3.0 ports (x4)
6. HDMI output port
7. DisplayPort connector
8. DC Barrel jack for 5V power input
9. MIPI CSI-2 camera connectors

Figure 3. 3: Naming parts of Jetson Nano.

3.3 Setting Up and Running

3.3.1Required Components

- Jetson Nano Developer Kit
- Raspberry Pi Camera v2
- Micro-USB power supply
- Super Starter Kit
- MicroSD card (32GB UHS-1 minimum recommended)
- Keyboard, Mouse and Display Monitor with HDMI support

Initially, a computer with an Internet connection and the ability to flash a microSD card is also required.

3.3.2Items for Getting Started

MicroSD card

The Jetson Nano Developer Kit uses a microSD card as the boot device and main storage. For your project, it is important to have a card that is fast and large enough; the minimum recommended capacity is a 32 GB UHS-1 card.

Please refer to the instructions below to flash your microSD card using the operating system and software [19].

Micro-USB power supply

You need to use a high-quality power supply to power the developer kit, which can provide 5V⎓2A on the Micro-USB port of the developer kit. Not every power supply that promises "5V⎓2A" will actually do this.

As an example of a good power supply, NVIDIA has verified Adafruit's 5V 2.5A switching power supply and 20AWG Micro-USB cable (GEO151UB-6025). It is designed to overcome common problems with USB power; for details, please refer to the linked product page [19].

3.3.3Write Image to the microSD Card

To prepare a microSD card, you need a computer with an Internet connection and the ability to read and write SD cards through the built-in SD card slot or adapter.

1. Download the Jetson Nano Developer Kit SD Card Image, and note where it was saved on the computer.
2. Write the image to your microSD card according to your computer's operating system: Windows, macOS, or Linux.

After your microSD card is ready, proceed to set up your developer kit.

3.3.4Initial Setup with Display Attached

Figure 3. 4: Jetson Nano Setup with Display Attached.

3.3.4.1 Setup Steps

1. Unfold the paper stand and place inside the developer kit box.
2. Insert the microSD card (with system image already written to it) into the slot on the underside of the Jetson Nano module.

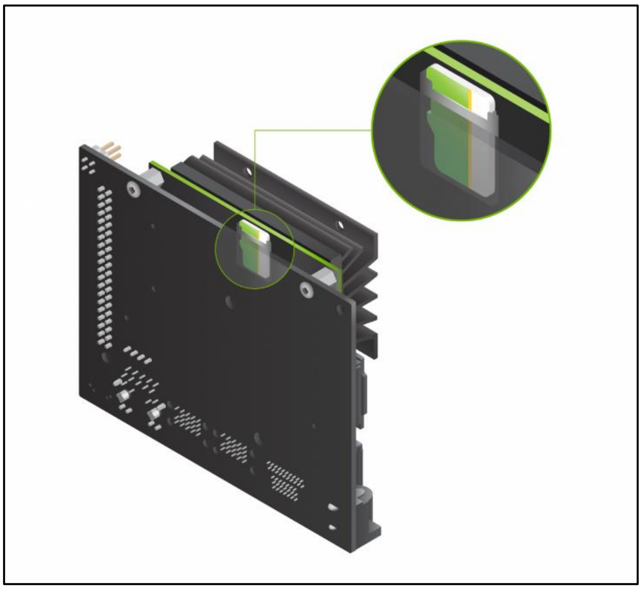

Figure 3. 5: Inserting MicroSD card into the slot.

3. Set the developer kit on top of the paper stand.

4. Power on your computer display and connect it.

5. Connect the USB keyboard and mouse.

6. Connect your Micro-USB power supply. The developer kit will power on and boot automatically.

3.3.4.2 First Boot

Once the developer kit is powered on, the green LED next to the Micro-USB connector will light up.

When you start for the first time, the developer kit will guide you through some initial settings, including:

- Review and accept NVIDIA Jetson software EULA
- Select system language, keyboard layout, and time zone
- Create username, password, and computer name
- Select APP partition size, it is recommended to use the max size suggested

After Logging In, you will see this screen.

Figure 3. 6: Jetson Nano First Boot Screen.

3.3.5Connecting the Camera to the Jetson Nano

1. Pull the CSI port and insert the camera ribbon cable in the port. Make sure to align the connection leads on the port with those on the ribbon. The connections on the ribbon should face the heat sink.
2. Power up the developer kit.
3. Verify that the camera is correctly configured and recognized.

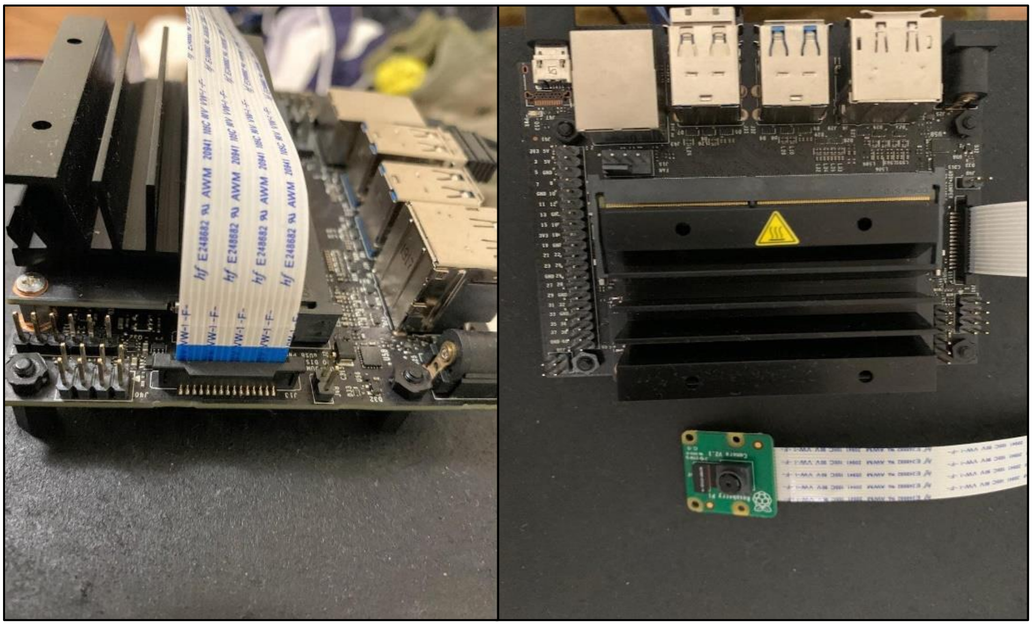

Figure 3. 7: Connecting the camera ribbon cable with the CSI port.

A simple command-line instruction to check for the camera connection is:

$ ls -l /dev/video0 Crw-rw----+ 1 root video 81, 0 Jan 2 14:30 /dev/video0

ls is a command to list the contents of a directory or location. The -l attribute displays the content in a long list format, which contains information about permissions, users, file owners, and so on. /dev/video0 is the location where all device files or special files in Linux are stored.

If the output of the command is blank, it means that there is no camera connected in the development kit. The NVidia Jetson series uses the GStreamer pipeline to process media applications. GStreamer is a multimedia framework for back-end processing tasks such as format modification, display-driven coordination, and data processing. It will also be used here with the Raspberry Pi camera [19].

3.4 Setting up Developing Environment

The Jetson Nano development kit requires some packages and tools to implement object detection and recognition tasks. All installations will be for Python3 [19].

- OpenCV- Open Source Computer Vision Library.
- Numpy - Scientific computing library supporting array objects.
- CMake - Meta-Build System for C++.

- Git - Version Control System.
- Pip3 - is the official package installer for Python 3.
- Keras - is an API designed for human beings.
- Prerequisites and Dependencies

Run these in the Jetson Nano terminal to install these packages:

```
$ sudo apt-get update
$ sudo apt-get install libhdf5-serial-dev hdf5-tools libhdf5-dev
zlib1g-dev zip libjpeg8-dev liblapack-dev libblas-dev gfortran
$ sudo apt-get install python3-pip3
$ sudo pip3 install -U pip testresources setuptools==49.6.0
$ sudo pip3 install -U numpy==1.19.4 future==0.18.2 mock==3.0.5 h5py==2.10.0
keras_preprocessing==1.1.1 keras_applications==1.0.8 gast==0.2.2 futures
protobuf pybind11
$sudo pip3 install adafruit-circuitpython-servokit
$sudo apt-get install git cmake libpython3-dev python3-numpy
$sudo apt-get install python3-opencv
```

3.5 Conclusion

Intelligent edge aims to integrate AI into the edge for many features dynamicity, adaptability on edge, maintenance and management. With the progress of diversity of communication technology and network access methods. At the same time, the infrastructure of edge computing plays role as an intermediate medium, establishing between ubiquitous end devices and the cloud more reliable and persistent.

As a distinctive and more extensively using new form of application, variety of deep learning basing on intelligent services [9], various deep learning-based intelligent services and applications have changed many aspects of people's lives due to the great advantages of Deep Learning (DL) in the fields of Computer Vision (CV) and Natural Language Processing (NLP).

Thus, the end devices, edge, and cloud are gradually merging into a community of shared resources. However, the maintenance and management of such a large and complex overall architecture (community) involving wireless communication, networking, computing, storage, etc., is a major challenge [33].

A variety of benefits containing Edge AI, this some citing with next points:

- Edge AI-monitored smart devices are supportive of many industries and activities because of its flexibility.
- Edge AI also offers a level of safety and security too elevated. Streaming data through devices connected to internet appears susceptible and vulnerable to breach also for other cybercrimes.

Many applications shows the impact of Edge-AI: Image analytics, Audio Analytics and Inertial Environmental Sensor Analytics {all sort of data} as analyzing, prediction and detecting.

CHAPTER 4

OBJECT DETECTION WITH JETSON NANO

4.1 Introduction

A diversity algorithms has been utilized for real-time image recognition, each algorithm has its advantages and disadvantages.

In this chapter a wide range of algorithms has been shown attempting to justify the use of the suggested Pipeline and architecture.

4.2 Object Detection

4.2.1 Definition of Object Detection

Object detection is a frequent computer vision problem that identifies and locates some categories of objects in an image. Object positioning can be interpreted in many ways, including creating a bounding box around the object [20].

- Input: An image with one or more objects, such as a photograph.
- Output: One or more bounding boxes (e.g. defined by a point, width, and height), and a class label for each bounding box.

Figure 4. 1: Object detection output.

4.2.2Data requirements

To train a custom model, you need to label the data. The label data in the object detection context is an image with corresponding bounding box coordinates and labels. That is the (x, y) coordinates of the lower-left corner and the upper right corner + class.

Example: The normalized bounding box coordinates for the dogs in the image are e.g. [0.1, 0.44, 0.34, 0.56] and [0.72, 0.57, 0.87, 0.77].

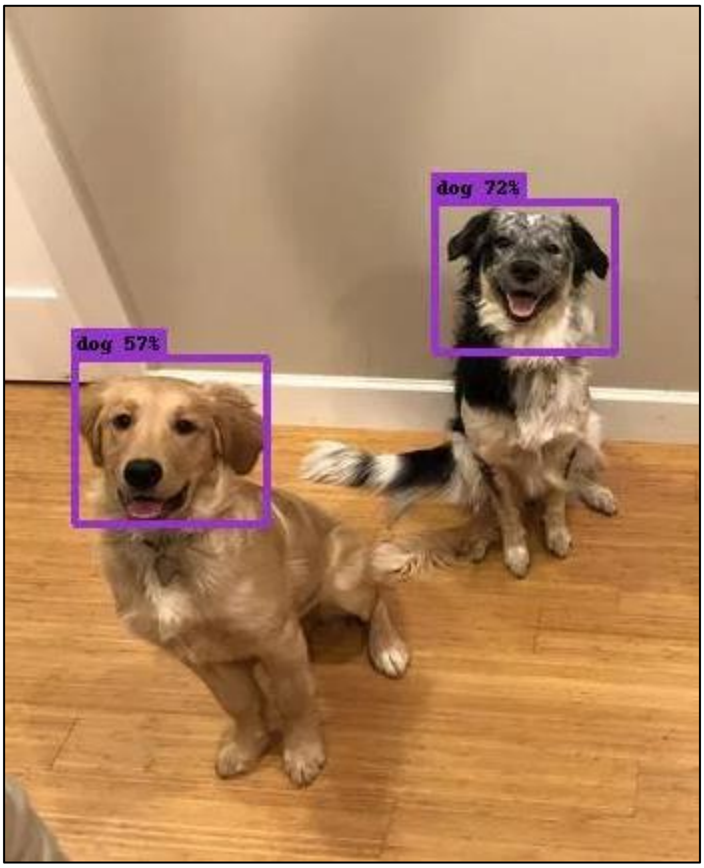

Figure 4. 2: The normalized bounding box coordinates example.

A frequently asked question is: How many pictures do I need to perform object detection on question X? On the contrary, it is more important to understand well which scenarios the model will deploy. Each class must have a large number (for example, >100 and possibly >1000) of representative images. In this case, representativeness means that they should correspond to the range of scenes where the model will use. If you are building a traffic sign detection model that will run in a car, you must use images taken under different weather, lighting, and camera conditions in the appropriate context. The object detection model is not magical. If the model does not have enough data to learn general patterns, it will not perform well in production [21].

4.2.3 General object detection framework

Generally, the object detection framework consists of three steps: [21]

1. First, use a model or algorithm to generate a region of interest or region proposal. These region proposals are a large set of bounding boxes (i.e. object positioning components) that span the entire image.
2. In the second step, extract visual features for each bounding box, evaluate them, and determine whether objects exist in the proposal based on the visual features (i.e. object classification components).
3. In the last step of post-processing, overlapping boxes will be integrated into a single bounding box (i.e. there is no maximum suppression).

4.3 Top-performing deep learning models for Object Detection

An overview about the models:

4.3.1 R-CNN Model Family

The R-CNN series of methods refer to R-CNN, which refers to "regions with CNN features" or "region-based convolutional neural networks" developed by Ross Girshick and al. [22]

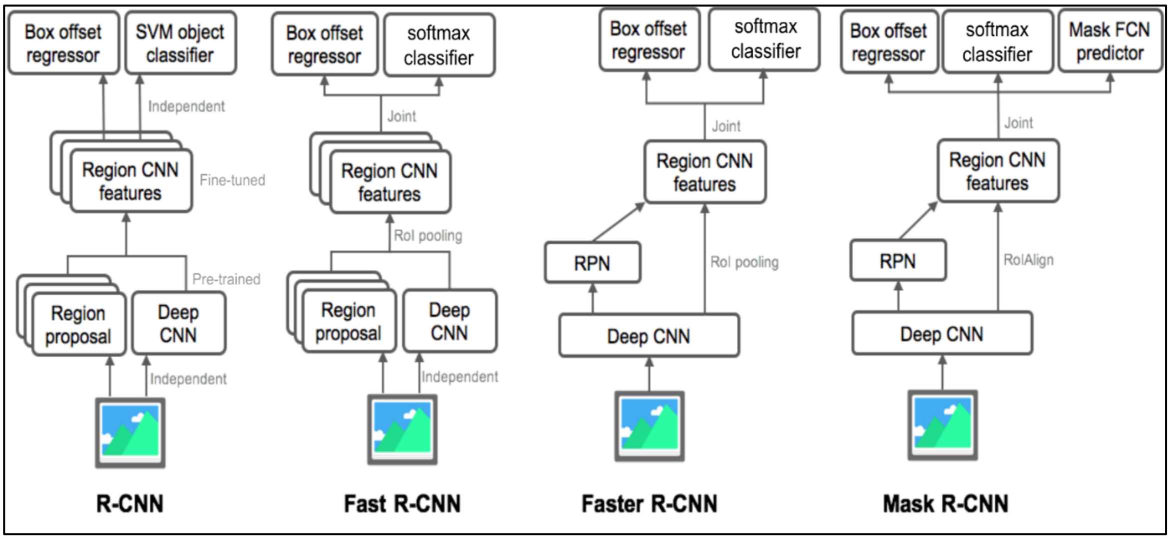

Figure 4. 3: R-CNN Family Summary.

Which includes these techniques: [22]

4.3.1.1 R-CNN

R-CNN is described in a paper by Ross Girshick and al, in 2014 from the University of California, Berkeley, titled "Rich Feature Hierarchy for Accurate Object Detection and Semantic Segmentation" [22].

It may be one of the first large-scale successful applications of convolutional neural networks in object localization, detection, and segmentation. The method has been proven on the benchmark data set, and the most advanced results at the time have been achieved on the VOC-2012 data set and the 200-category ILSVRC-2013 object detection data set[22].

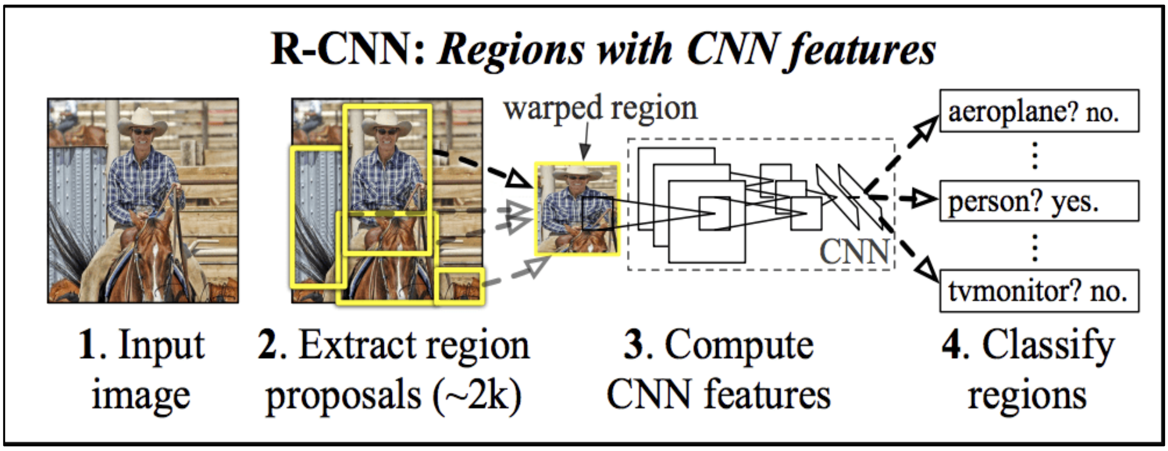

Figure 4. 4: Summary of the R-CNN Model Architecture.

4.3.1.2 Fast R-CNN

Given the great success of R-CNN, Ross Girshick, who was working at Microsoft Research at the time, proposed an extension to solve the speed problem of R-CNN in a 2015 paper entitled "Fast R-CNN".

Fast R-CNN presented as a single model instead of a pipeline to learn directly and output regions and classifications [22].

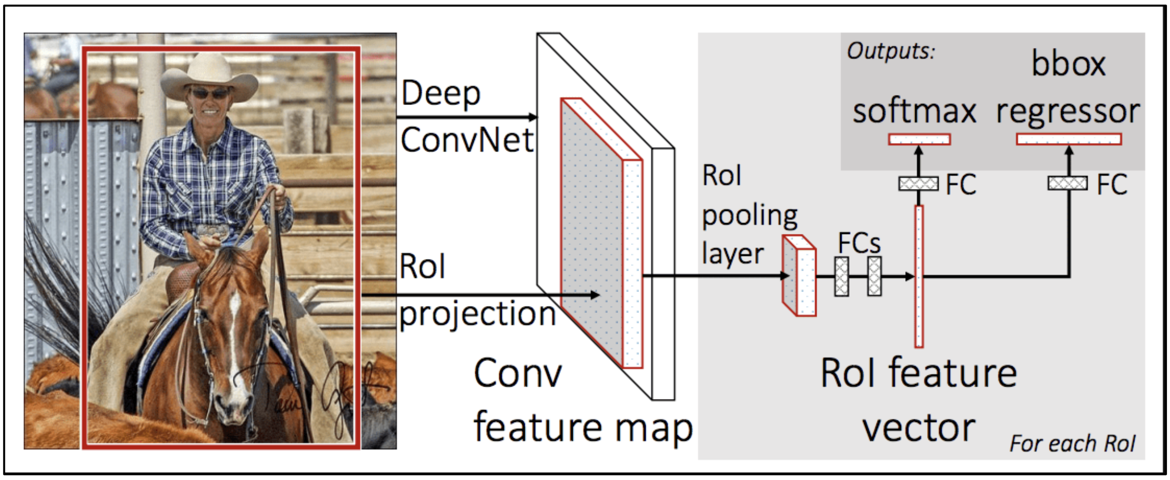

Figure 4. 5: Summary of the Fast R-CNN Model Architecture.

4.3.1.3 Faster R-CNN

The model architecture was developed by Shaoqing Ren and al, for the sake of improving speed training and recognition. Microsoft Research in 2016 article entitled "Faster R-CNN: Towards Real-Time Object Discovery with Regional Networks." The architecture was the basis for the first-place results achieved on both the ILSVRC-2015 and MS COCO-2015 object recognition and detection competition tasks [22].

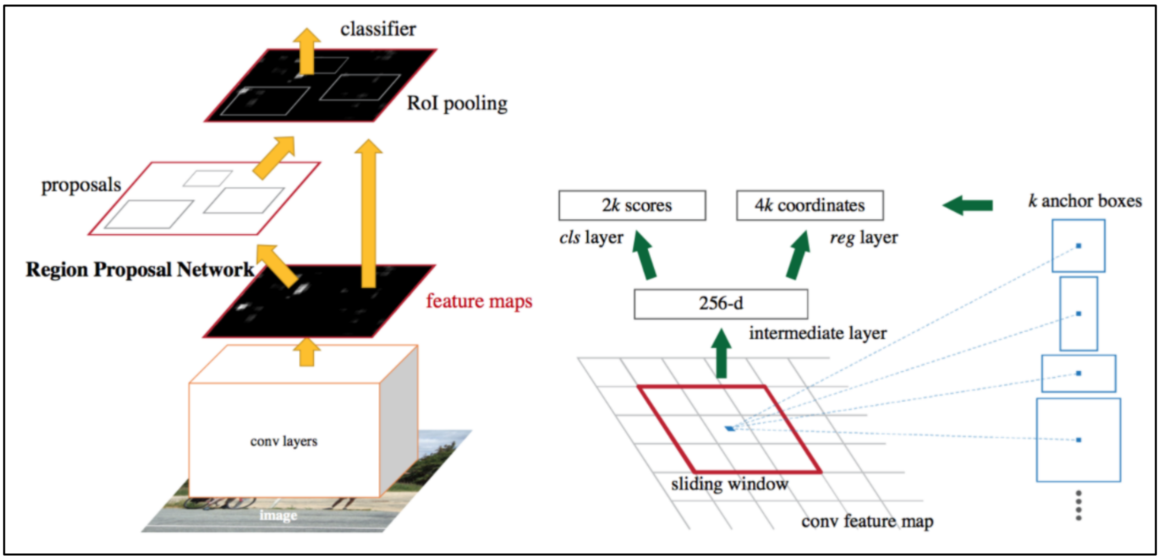

Figure 4. 6: Summary of the Faster R-CNN Model Architecture.

4.3.1.4 Mask R-CNN

Mask R-CNN (He et al., 2017) extends Faster R-CNN to pixel-level image segmentation. The key is to separate the classification and mask prediction tasks at the pixel level. Based on the Faster R-CNN structure. A third branch has been added to predict the object mask in parallel with the existing classification and positioning branches. The branch mask is a small, fully connected network that is applied to each RoI and predicts the segmentation mask pixel by pixel [22].

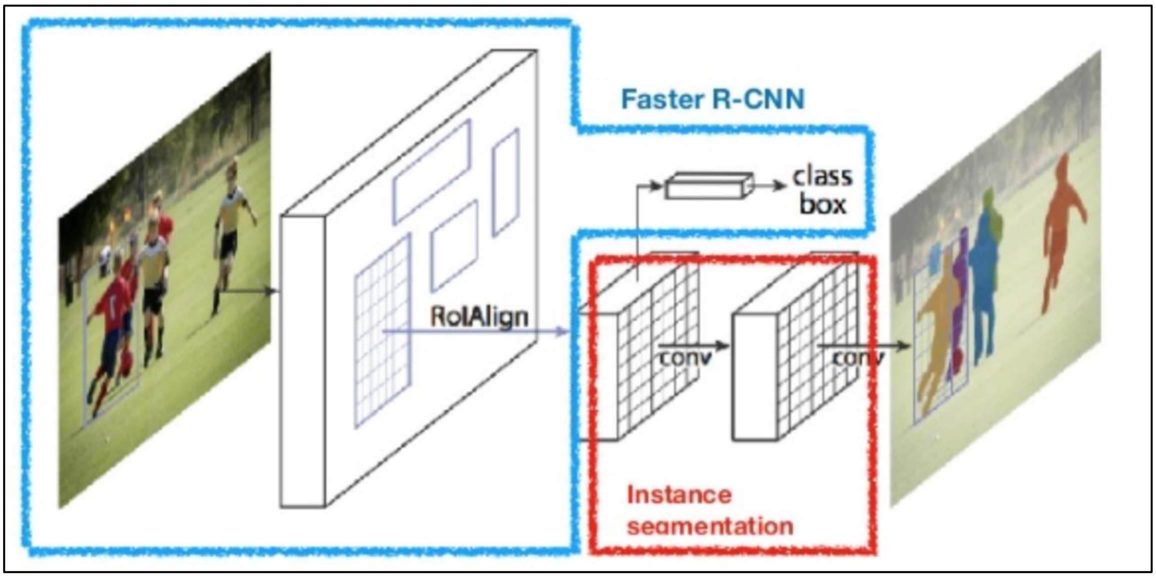

Figure 4. 7: Mask R-CNN is Faster R-CNN model with image segmentation.

4.3.2YOLO Model Family

Joseph Redmon and Al have developed another popular series of object recognition models, commonly known as YOLO or "You only see it once." The developed R-CNN model can be more accurate, but the YOLO series models are much faster than R-CNN and provide real-time object recognition.

Some of these methods include: [22]

4.3.2.1 YOLO

The YOLO model was created by Joseph Redmon and Al. An article entitled "Take a Look: Unified Real-Time Object Detection" in 2015 Note that the developer of R-CNN, Ross Girshik, was also the author and contributor of this work, and later worked at Facebook AI Research.

The method involves a single neural network trained end-to-end that takes photos as input and directly predicts the bounding box and class labels of each bounding box. For the speed-optimized version of the model, the second sum is up to 155 frames per second for a speed-optimized version of the model [22].

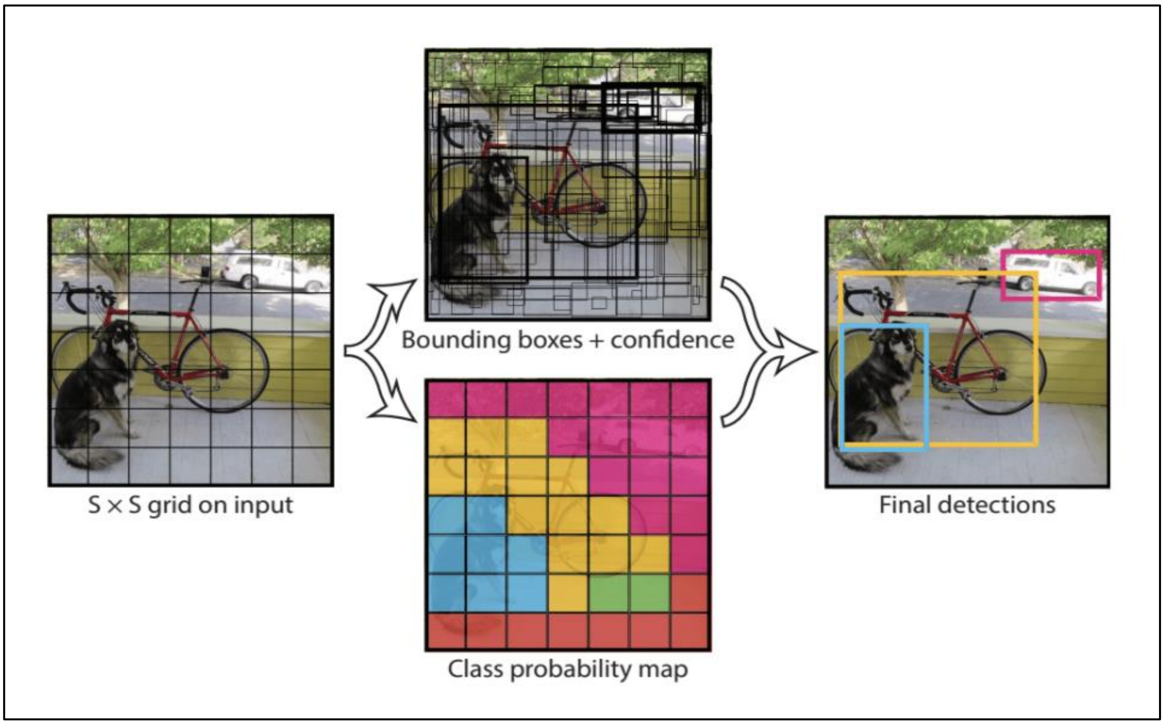

Figure 4. 8: Summary of Predictions made by YOLO Model.

4.3.2.2 YOLOv2 (YOLO9000) and YOLOv3

The model was updated by Joseph Redmon and Ali Farhadi in their 2016 article entitled "YOLO9000: Better, Faster, and Stronger".

Although this variation of the model is referred to as YOLO v2,it describes that it is trained on two sets of object recognition data in parallel, and can predict 9000 types of objects, hence given the name "YOLO9000".

Some training and architectural changes were made to the model, such as the use of batch normalization and high-resolution input images. Joseph Redmon and Ali Farhadi suggested further improvements to the model in their 2018 article "YOLOv3: Improvement over time". The improvements are small, including a deeper feature detection network and subtle rendering changes [22].

4.3.3MOBILENET-SSD

4.3.3.1 General Definition

The mobile-net-SSD model is a single-Shot multi-box Detection (SSD) network for object recognition.

The input to the model is a BLOB consisting of a single 1x3x300x300 image arranged in BGR order. Before uploading the image blob to the network, the average BGR value should be subtracted: [127.5, 127.5, 127.5]. In addition, these values must be divided by 0.007843.The result of the model is a typical vector containing the tracking object data, as described above [23].

4.3.3.2 Definition of MobileNet

MobileNet is an optimized architecture that uses deep analytical convolution to build deep and lightweight convolutional neural networks and provide efficient models for mobile and embedded image processing applications. The MobileNet framework is based on the depthwise separable filter, as shown in Figure (4. 9).

The depth wise convolution filter consists of a depth wise convolution filter and a point convolution filter. The deep convolution filter performs a single convolution on each input channel, and the output convolution filter linearly combines the deep convolution output with 1 * 1 convolution, as shown in Figure 4.10 [24].

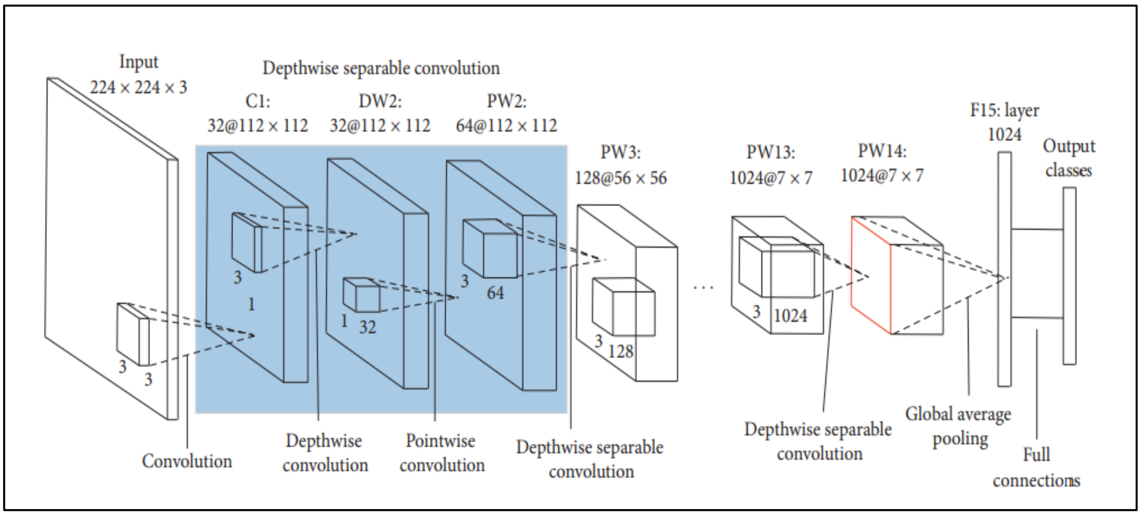

Figure 4. 9: Architecture of MobileNet.

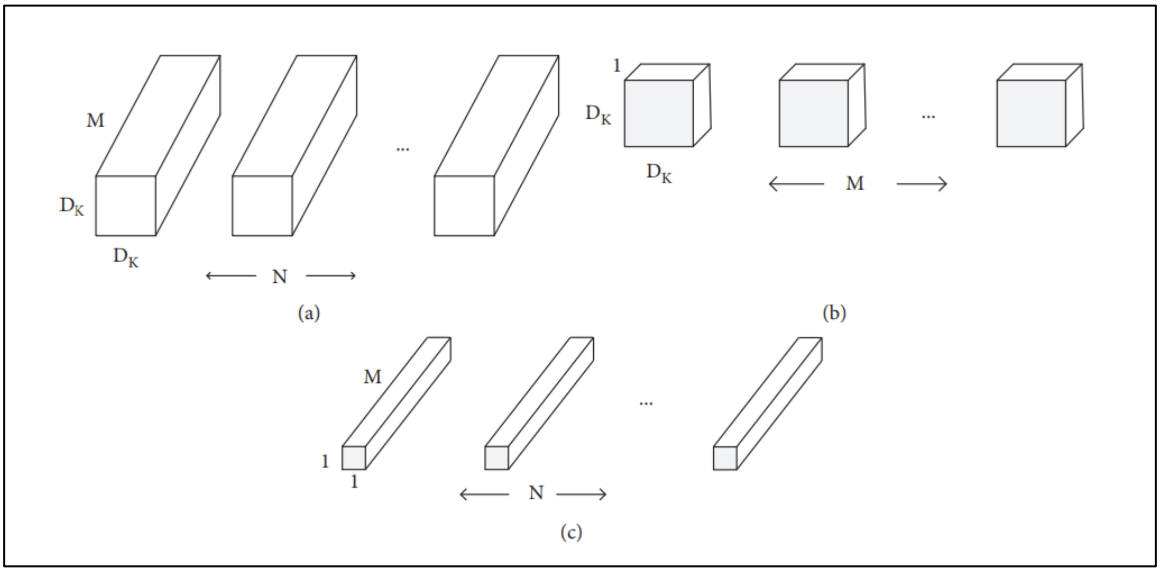

Figure 4. 10: Standard convolutional filters and depthwise separable filters. (a) Standard convolutional filters, (b) depthwise convolutional.

In many practical applications, such as self-driving cars, it is necessary to directly perform recognition tasks on limited computational devices. MobileNet was developed in 2017 to fulfill this requirement [25].

4.3.3.3 Definition of SSD

SSD: Single Shot Detector is a model developed by Google Research Group in 2016 to meet the needs of models that can trigger real-time on embedded devices without significantly reducing accuracy.

Object Recognition Take one single shot within the image SSD to recognize multiple objects in the picture. The SSD method is based on a convolutional feedback network, which creates a series of fixed-size bounding boxes and estimates the existence of feature class instances.

It consists of two parts:

- Extracting feature maps and
- Applying convolution filters for object recognition.

SSD is designed to be independent of the base network so, it can work with any base network such as VGG, YOLO, and MobileNet [25].

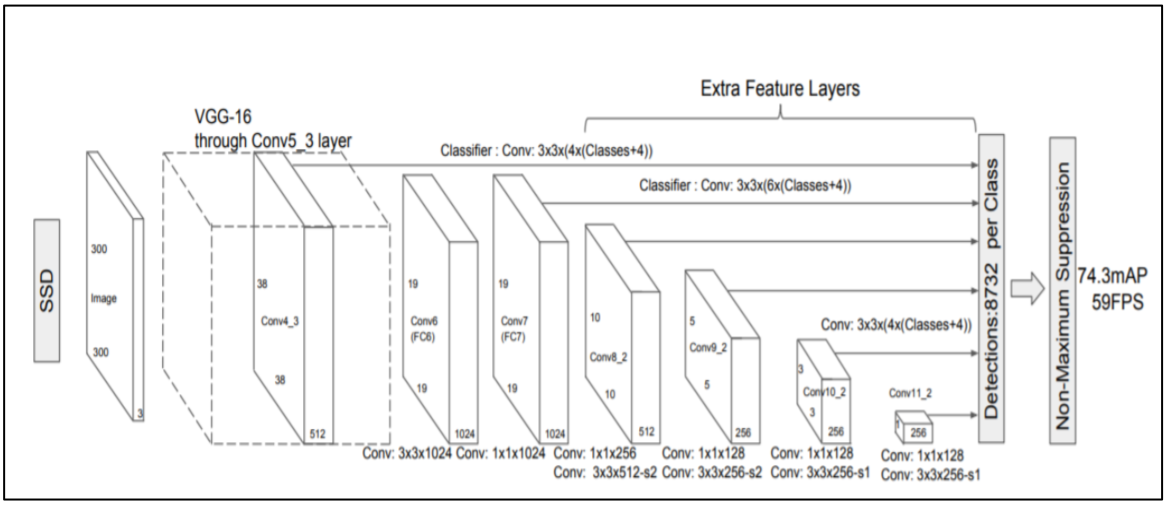

Figure 4. 11: SSD Architecture.

4.3.3.4 MobileNet * SSD

To further eliminate the practical limitations of resource- and energy-intensive neural networks running on low-level devices in real-time applications, MobileNet integrated into the SSD structure. When MobileNet will be used as the SSD backbone, it became MobileNet SSD [25].

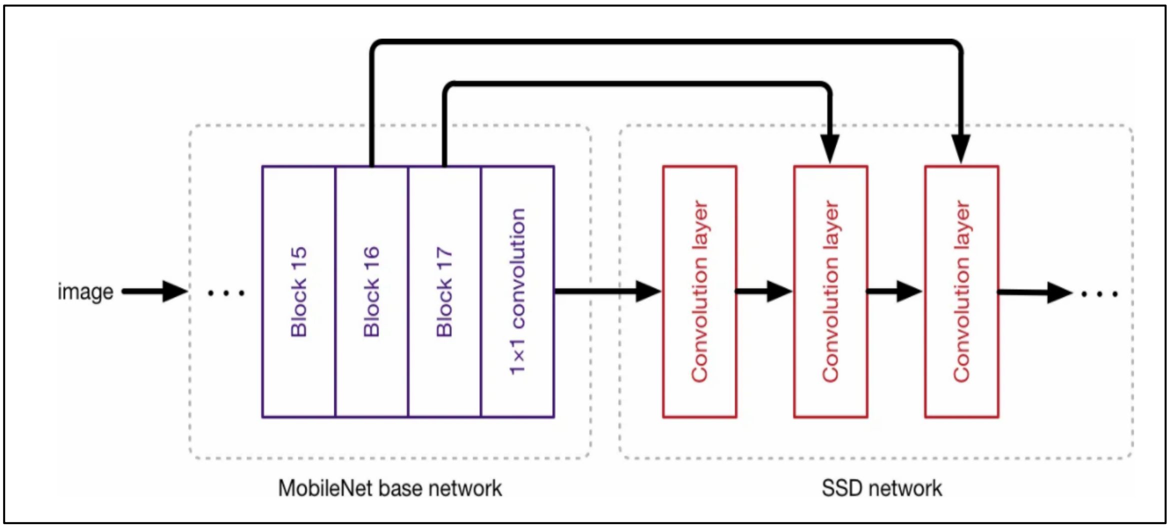

Figure 4. 12: MobileNet SSD overview.

Note: The MobileNet SSD method was trained firstly on the COCO data set and then adapted to PASCAL VOC, reaching 72.7% mAP (mean average precision).

4.4MOBILENET-SSD vs R-CNN vs YOLO in The Edge

A single shot detector (SSD) like YOLO takes only one picture to use multibox to detect multiple objects in an image. It is significantly faster and has a high-accuracy object recognition algorithm.

Here is below a quick comparison between the speed and accuracy of different object detection models onVOC2007: [26]

- SDD300 : 59 FPS with mAP 74.3%
- SSD500 : 22FPS with mAP 76.9%
- Faster R-CNN : 7 FPS with mAP 73.2%
- YOLO : 45 FPS with mAP 63.4%

When both the models (MobileNet-SSD and YOLO) trained on a COCO dataset (330K Images, 80+ object) the following results are obtained:

Type of model	Size on the disk	Detection speed
YOLO_v2	269.9 mb	2-3 fps
MobileNet	27.5 mb	6-8 fps

Table 4. 1: Comparison between MobileNet and YOLO_v2

Note: In our case, the project will develop in Edge AI. The best model for edge devices is MobileNet-SSD because his small space and compatibility with embedded devices. In addition, the results above showed that it has the best performance (detection speed…) and is the best in terms of capacity to work with similar devices.

4.5 Conclusion

This chapter summarizes different algorithms, also explained the used deep learning algorithms by validating the reason for using mobileNet.

Next chapter will give a detailed system by its realization and showing a real implementation.

CHAPTER 5

REALIZATION OF THE PROJECT

5.1 Introduction

To concretize the proposed pipeline and architecture, many steps should be explained in order to show step by step how this work has implemented this system.

The use of metrics that provide the evaluation and assessment of this system empirically and visually.

5.2 Setting Up Jetson Inference Library

In the repository there are quite a lot of other **git** submodules, the following instructions also utilize as a recursive keyword to clone them correctly:

```
$ git clone --recursive https://www.github.com/dusty-nv/jetson-inference.git
$ cd jetson-inference
$ mkdir build && cd build
$ cmake ../
```

Note: For installing some prerequisite software packages on Jetson, this command will run the CMakePreBuild.sh script, which requires **sudo** permission. This script also permit downloading the pre-trained network from the web service [27].

5.2.1 Installing PyTorch

Noting that JetPack 4.4 is used to launch the PyTorch. To retrain the network in migration learning later for this project, it should start by installing PyTorch on used Jetson. This step is facultative. For retraining the network in migration learning later in this project, it can select -install PyTorch –on Jetson. Once your project does not need to install PyTorch, this step can be skipped.

If necessary, select the PyTorch package version of Python 2.7 and/or Python 3.6 zhen it needs to install, and press Enter to continue. Otherwise, uncheck these options, it will skip the installation of PyTorch [27].

N.B */ The use of JetPack 4.2 or higher, you will now run another tool./*

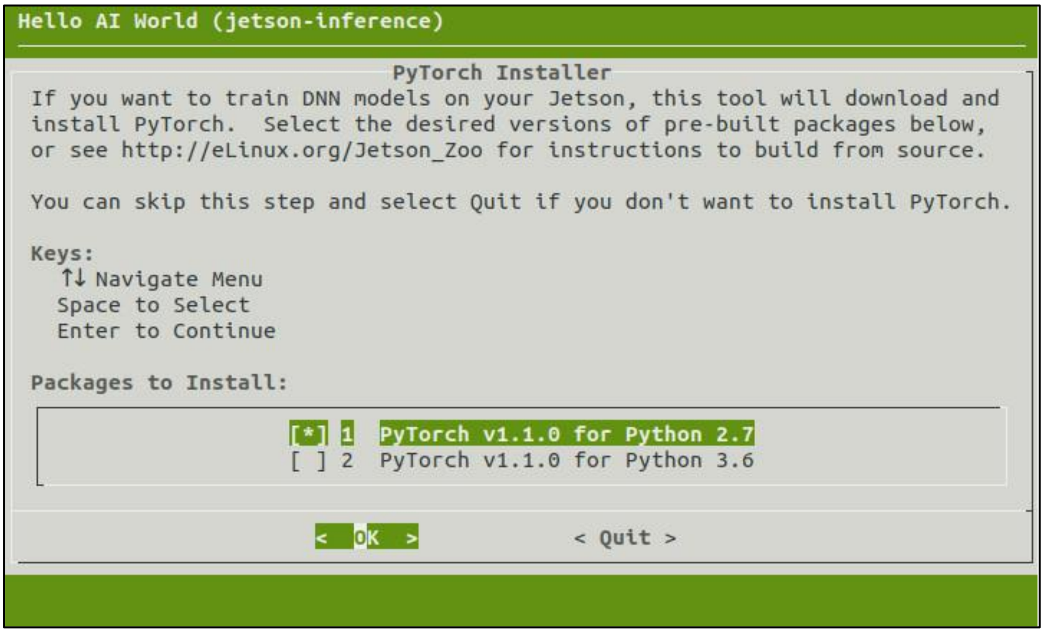

Figure 5. 1: Installing PyTorch window.

5.2.2 Compiling the Project

Make sure that still the Jetson-inference mode /build directory created above.

Then run make and sudo make install to build libraries, Python extension bindings and code examples: [27]

```
$ cd jetson-inference/build
# omit if working directory is already build/ from above
$ make
$ sudo make install
$ sudo ldconfig
```

The project will be built to jetson-inference/build/aarch64, with the following directory structure: [27]

```
|-build
  \aarch64
    \bin         where the sample binaries are built to
      \networks    where the network models are stored
      \images      where the test images are stored
    \include      where the headers reside
    \lib         where the libraries are build to
```

5.3 Collecting Detection Dataset

5.3.1 Creating the Classes File

Under jetson-inference/python/training/detection/ssd/data, create an empty directory to store the data set and a text file that defines class labels (usually called classes.txt). Each line of the label file contains a class label, for example:

```
scorpion
scorpionUv
ladybird
```

The use of container, then storing dataset in a Mounted Directory like above, so it's saved after the container shuts down [28].

5.3.2 Launching the Tool

The camera-capture tool accepts the same input URI's on the command line that has been found on the Camera Streaming and Multimedia page.

Below are some example commands for launching the tool: [28]

```
$ camera-capture csi://0       # using default MIPI CSI camera
$ camera-capture /dev/video0   # using V4L2 camera /dev/video0
```

5.3.3 Collecting Data

The following is the data capture control window, after the data set type drop-down list is set to detection mode (perform this operation first) [28].

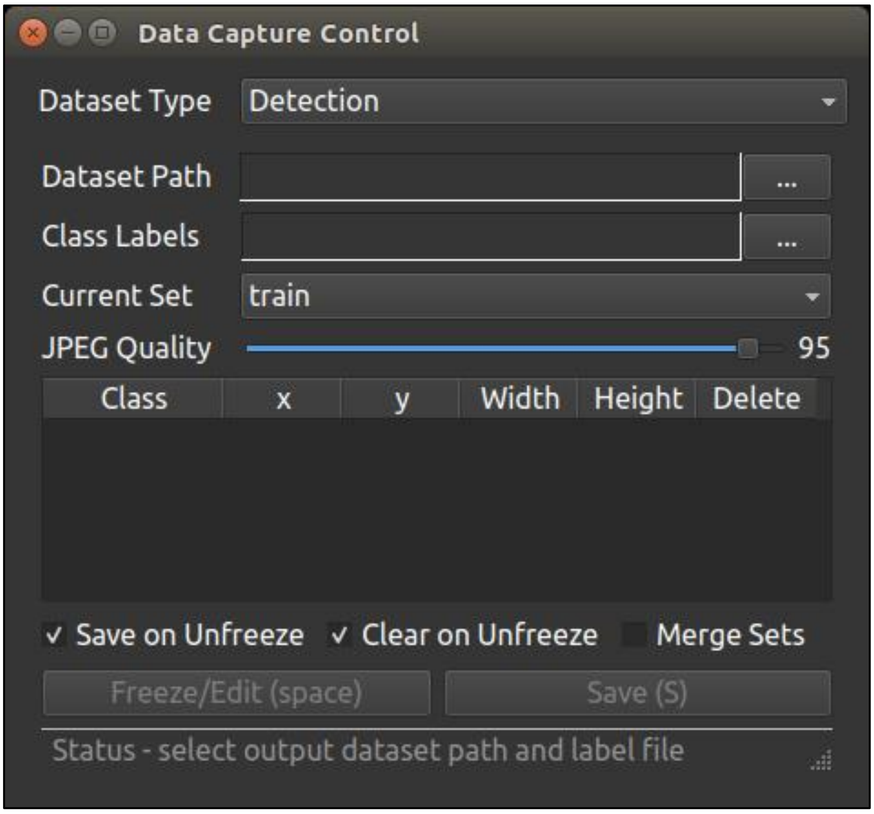

Figure 5. 2: Collecting Detection Dataset.

Then, open the dataset path and class label you created. The freeze/edit and save buttons will become active.

Place the camera on the object in the scene and click the "Freeze/Edit" button (or press the space bar). The live camera view will be "frozen" and you will be able to draw a bounding box on the object. After selecting the appropriate object class for each bounding box in the grid table in the control window. After finishing the image tagging, click the pressed "Freeze/Edit" button again to save the data and unfreeze the camera view of the next image [28].

Other widgets in the control window include:

- **Save on Unfreeze** - automatically save the data when **Freeze/Edit** is unfreezed
- **Clear on Unfreeze** - automatically remove the previous bounding boxes on unfreeze
- **Merge Sets** - save the same data across the train, val, and test sets
- **Current Set** - select from train/ val/ test sets
    - for object detection, you need at least train and test sets

  - although if you check Merge Sets, the data will be replicated as train, val, and test
- **JPEG Quality** - control the encoding quality and disk size of the saved images

It is important that your data is collected from different object orientations, camera viewpoints, lighting conditions, and ideally different backgrounds to create a model that is robust to noise and environmental changes. If you find that your model is not performing well, try adding more training data and adjusting the conditions [28].

The previously used camera-capture tool can also label object detection datasets from live video: [28]

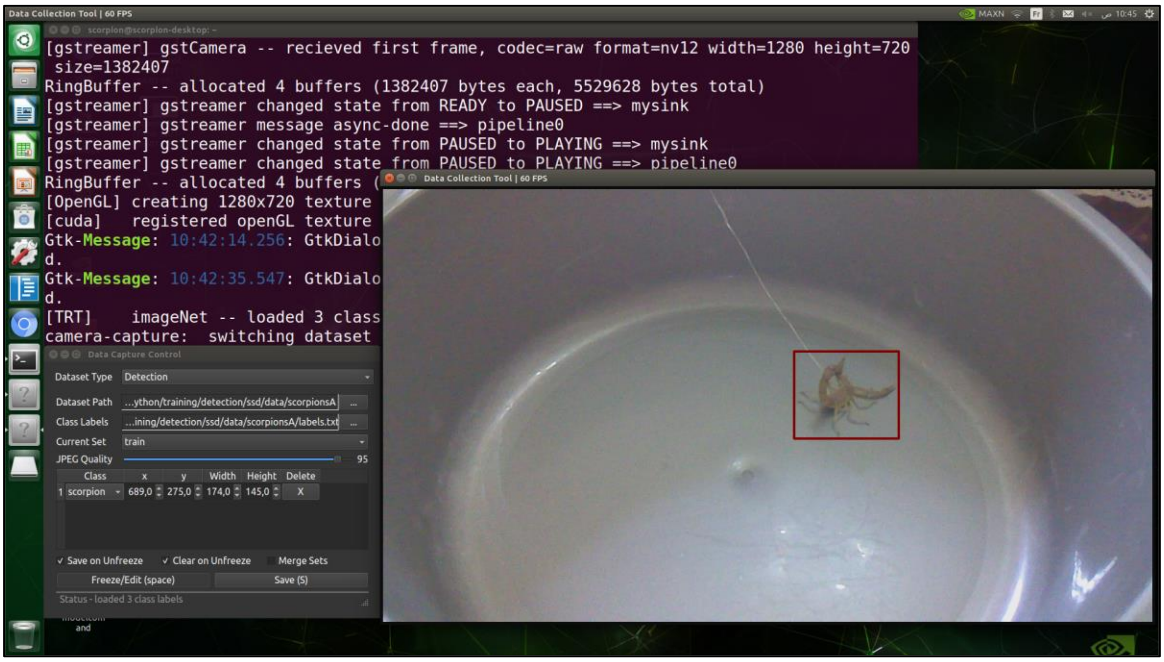

Figure 5. 3: Collecting Detection Dataset and libeling.

When the Dataset Type drop-down is in Detection mode, the tool creates datasets in **Pascal VOC** format (which is supported during training). [28]

Note: If you want to label a set of images that, you already have (as opposed to capturing them from camera), try using a tool like CVAT and export the dataset in Pascal VOC format. Then create a labels.txt in the dataset with the names of each of your object classes.

5.4 Training The Model

When you have collected a bunch of data, then you can try training a model on it using the same train_ssd.py script. The training process is the same as the previous example, with the exception that the --dataset-type=voc and --data=<PATH> arguments should be set:[28]

```
$ cd jetson-inference/python/training/detection/ssd
$ python3 train_ssd.py --dataset-type=voc --data=data/scorpion --model-dir=models/scorpion
```

Note: if you run out of memory or your process is "killed" during training, try Mounting SWAP and Disabling the Desktop GUI.

To save memory, you can also reduce the --batch-size (default 4) and --workers (default 2). Like before, after training you'll need to convert your PyTorch model to ONNX:

```
$ python3 onnx_export.py --model-dir=models/scorpion
```

The converted model will then be saved under <YOUR-MODEL>/ssd-mobilenet.onnx, which you can then load with the detectnet programs like we did in the previous examples:[28]

```
NET=models/scorpion
detectnet --model=$NET/ssd-mobilenet.onnx --labels=$NET/labels.txt \
--input-blob=input_0 --output-cvg=scores --                    output-bbox=boxes     \
csi://0
```

Note: it is important to run inference with the labels file that gets generated to your model directory, and not the one that you originally created for your dataset. This is because a BACKGROUND class gets added to the class labels by train_ssd.py and saved to the model directory (which the trained model expects to use).

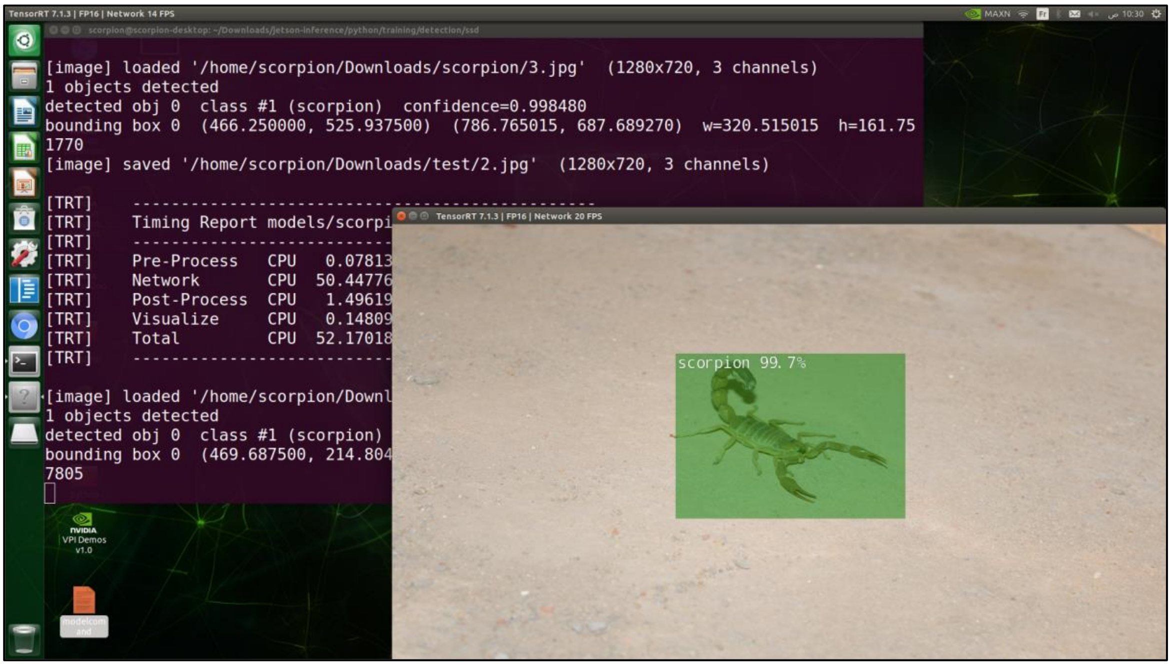

Figure 5. 4: Real time detection example for Scorpion.

5.5 Alerts and Notifications

In this section, we explain how we develop alerts (alarm, uvlight…) and scorpion detection app.

5.5.1 Alerts

5.5.1.1 Setting Up GPIO

To access NVIDIA Jetson Nano GPIO, please use Jetson's Jetson.GPIO library. This library is modified from the RPI GPIO (Raspberry Pi) library. You can use Jetson.GPIO to build Python applications to access NVIDIA Jetson Nano devices. You can find the Jetson.GPIO library in the following location https:// github.com/NVIDIA/jetson-gpio [29].

Now you can install Jetson.GPIO on the NVIDIA Jetson Nano using pip from Python.
Open Terminal and type this command:

$ sudo pip install Jetson.GPIO

After completing this installation, you can verify its success by opening Python shell. For instance, we use Python 3. Type these commands:

$ python3

>>> import Jetson.GPIO as GPIO

>>> GPIO.JETSON_INFO

>>> GPIO.VERSION

You should see the Jetson.GPIO library information and version. You can see my library information in Figure (5.5).

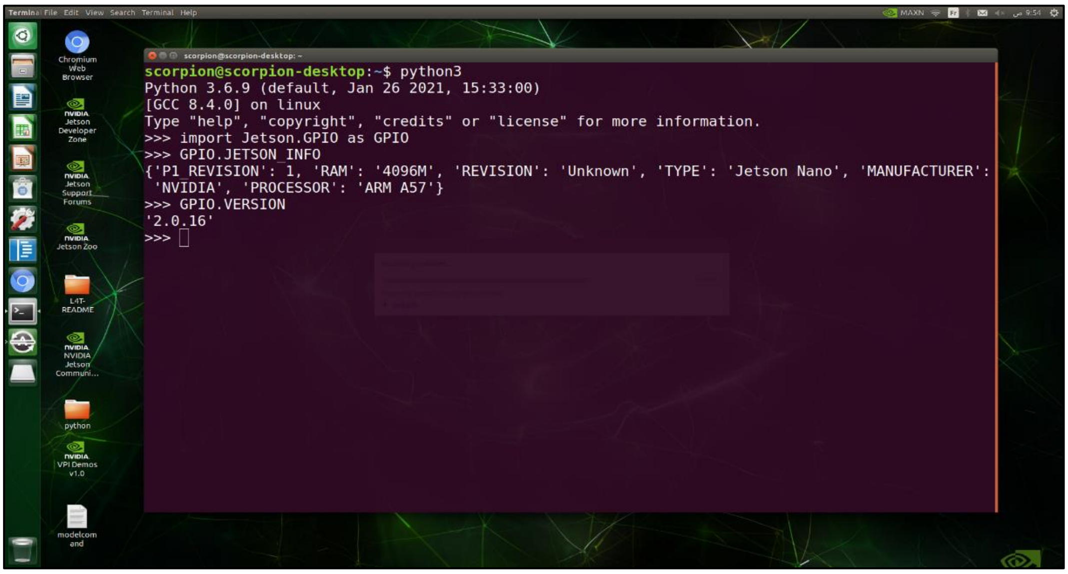

Figure 5. 5: Jetson.GPIO library information and version.

5.5.1.2 GPIO Programming

The NVIDIA Jetson Nano board has forty GPIO pins. You can use GPIO pins in input mode or output mode. You cannot use it in input and output modes at the same time.

For this demonstration, you can use the LED and make it blink. You need the following electronic components: [29]

- Breadboard
- LED Ultraviolet
- A resistor; 500 ohm
- Tow Transistor 2n2222
- button switch

- Buzzer
- motion sensor
- ten jumper cables
- implementation
- Connect the positive part of the button switch to pin 5v and the negative to pin GND and GPIO 23 pin on the NVIDIA Jetson Nano.
- Connect the positive part of the led Ultraviolet to GPIO 2 pin 5v and the negative to collector of transistor one 2n2222 on the Breadboard.
- Connect the base part of the transistor one to GPIO 23 pin and the emitter to GPIO 9 pin GND on the NVIDIA Jetson Nano.
- Connect the positive part of the Buzzer to GPIO 4 pin 5v and the negative to collector of transistor second 2n2222 on the Breadboard.
- Connect the base part of the transistor second to GPIO 26 pin and the emitter to GPIO 9 pin GND on the NVIDIA Jetson Nano.
- Connect the positive part of the motion sensor to GPIO 4 pin 5v and the negative to GPIO GND and vcc to GPIO 15 pin on the NVIDIA Jetson Nano.
- Connect a resistor between negative part of button switch and GPIO 9 pin GND.

5.5.2 Notifications

When the scorpion detected by the Jetson Nano, the program send a notification to the application to inform the user (example: homeowners…) that there is a scorpion around.

The Scorpion Detection Application

The application developed by **Flutter framework**, we choose this framework to develop our application because it allows the app to work with both operating systems **Android** and **iOS**.

This App enables the users to check their notifications and they can delete them after.

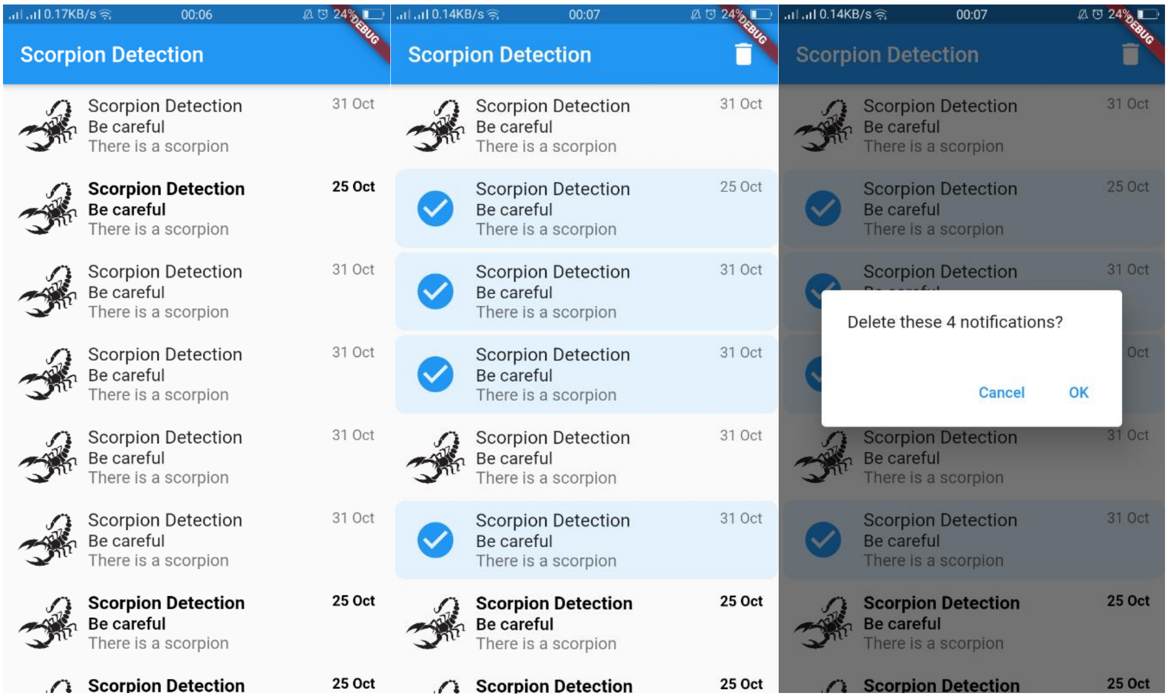

Figure 5. 6: Scorpion Detection Application User Interfaces.

The Transition Way of Notifications

In our project, we use the Firebase Api to create a real-time database Figure (5. 7). The moment the Jetson Nano detect a scorpion, it inserts an information to the database, any changes (insertion…) detected on the database the api will send a message to the Scorpion detection App, the App will get the new information from the database and notify the user.

Note: The user's phone must be connected the internet.

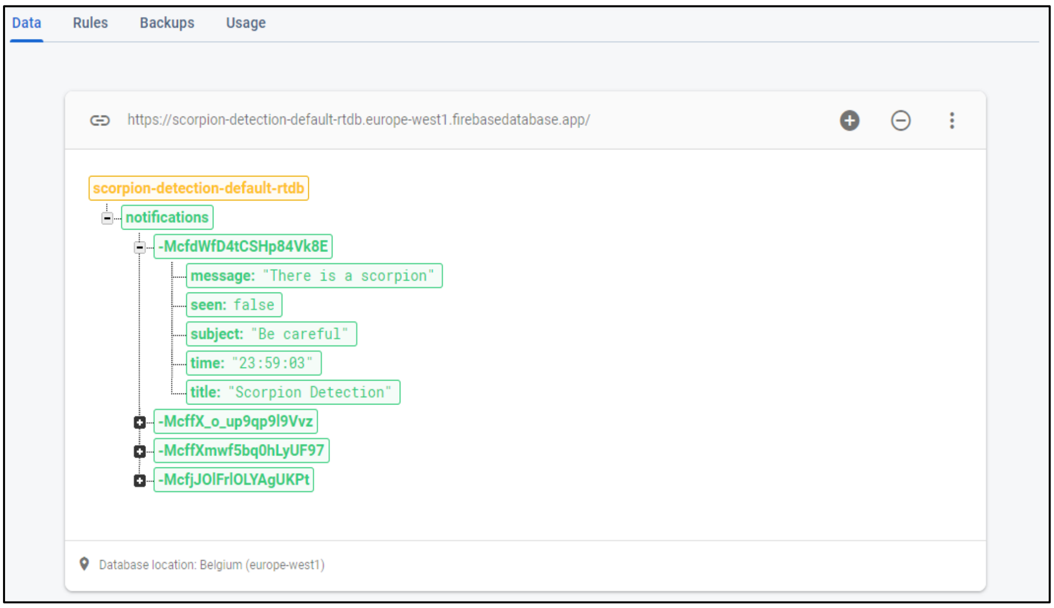

Figure 5. 7: Database Structure.

5.6 Results and Discussion

Some results of the scorpion detection:

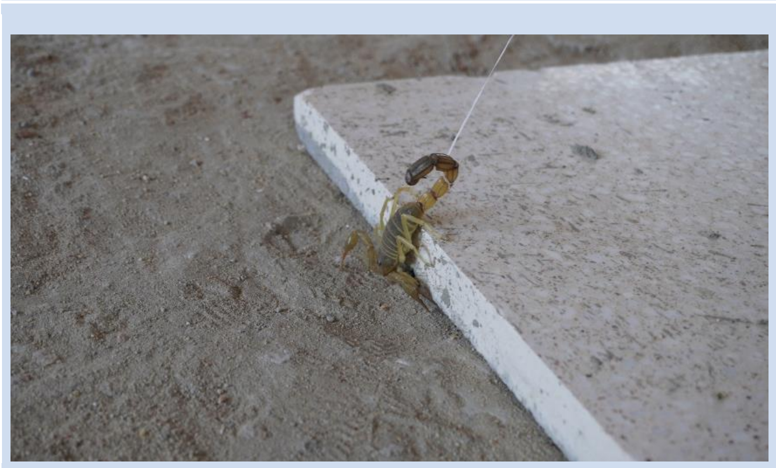

Figure 5. 8: Results of detection

Split the Original Set into Train, Validation and Test Data

First, we divided the data into two parts. The first part is to train the model and the second part is to test its accuracy. 700 images for training and 100 for testing. The original data contains 350 images of a scorpion in the daytime and 350 images of a scorpion at night under ultraviolet light. For the selection images, we took 40 images of a scorpion during the day, 40 images at night, and 20 images for various other insects.

	Positive	**Negative**
True	179	7
False	5	9

Table 5. 1: Confusion matrix.

Accuracy

Accuracy in classification problems is the number of correct predictions made by the model over all kinds predictions made. [30]

Accuracy = TP + TN / TP+TN+FP+FN

= 179 + 7 / 179 + 7 + 5 + 9

= 186 / 200

= 0.93 = 93 %

Precision

Precision is a measure based on CM that it tells us what percentage of the photos actually contain a scorpion. The predicted positives (image predicted contain scorpion are TP and FP) and the image actually contain a scorpion are TP [30].

Precision = TP / TP + FP

= 179 / 179 + 5

= 179 / 184

= 0.97 = 97 %

Recall

Recall is a measure that tells us the percentage of images that actually contain a scorpion and are classified by the algorithm as a scorpion. (Image predicted contain scorpion are TP and FN), and the image actually contain a scorpion are TP.

Note: FN is included because the image actually contain a scorpion even though the model predicted otherwise. [30]

Recall = TP / TP + FN

= 179 / 179 + 9

= 179 / 188 = 0.95 = 95 %

Specificity

Specificity is a measure that tells us the percentage of images without a scorpion, which the model predicted to be non-scorpion.

The actual negatives (image actually without a scorpion are FP and TN) and the photos that were categorized by the model as not containing a scorpion are TN.

Note: FP is included because the picture does not have a scorpion even though the model predicted specificity is the exact opposite of Recall. [30]

Specificity = TN / TN + FP

= 7 / 7 + 5

= 7 / 12

= 0.58 = 58 %

5.7 Conclusion

An Edge-AI approach is widely expanded these days according to its efficiency and usability, the aim of this work is to protecting people life by deploying a system has an inference to detect and classify a scorpion, also to notify and alert holder of its entrance.

This system provides a high performance in real-time of scorpion detecting using two modes: day and night.

GENERAL CONCLUSION

Nowadays, scorpion envenomation, often more deadly than snake bites, this remains very formidable in Algerian Sahara regions and even other regions in the globe where accumulate, as is often the case:

- Climatic conditions favorable to scorpions.
- Underdevelopment, which is accompanied by a deficit in prevention, infrastructure and means of therapeutic intervention.

In addition, the generalization of business or pleasure travel means that each of us can one day or another be confronted with this danger.

Insofar as the composition of the venoms is very varied, the symptomatology of envenomation ranges from simple pain at the point of sting, to the state of shock.

The efforts of advanced research laboratories have not yet achieved the efficient vaccines. Such vaccines would whether works adequately on some of species or not work on others, of course, to take a decisive step in the fight against scorpion. This last pass requires to explore a diversity type of scorpions, however, first of all a clear and insistent information of the local populations concerned and also of the travelers brought to frequent the same regions have multiplicity of reactions that conduct to a defensive reaction by scorpions.

Previously, one-way to prevent scorpion sting which is criteria of awareness that defined as a practical fight known and elaborated by the population:

- The epidemiology of the scorpion and the circumstances favorable to envenomation.
- The importance of hygiene in the habitat cleanliness in houses and around them.
- The advantage of wearing shoes, and the need not to lie on the floor anywhere, especially at night in summer.
- The role of chickens, ducks, geese... in the anti-scorpion fight.

- The interest of participating in the collection of scorpions.

This information is provided by health personnel at all levels, mobile teams, health committees and local officials. The information on prevention and first aid is one of the essential missions but not enough due to person face directly the danger, it takes on particular importance for the risk of scorpion stings, a problem that is often poorly understood, especially by those who are going to an infested region for the first time or even for locals who are known region.

The present work therefore proposes to lead to practical demarche, which can prevent persons from:

- The most dangerous scorpions from different risky regions of the globe.
- During perception of wanted object, the elementary rules of prevention is considered once the implemented model detect the scorpion.
- The event of Notification is considered as preventive gestures which make person life "protected".
- Alert with automatic buzzer in case that the mobile application holder did not see or notice the notification defined as a safer preventive person life to survive.

To accomplish this, the following phases are considered successively and more exhaustively by:

1. Acquisition step:
2. Object detection based deep learning algorithm;
3. Connections made between sensors, Jeston-nano and wireless network;
4. Notification submitted from edge AI device to mobile device;
5. Alert provided once the object detected.
6. Mobile interface interacting with user.

This project tracked the exploration of the prospect to run an Edge AI model for scorpion detection, trying to illustrate and reveal the ability that this technology can offer. The system we have implemented and tested has presented some functional insights about the advance possibility of collaborative neural computing. This model has tested on other classes different to scorpions.

The ”Edge Approach” is bit expensive according to students means, hence its advantages as bandwidth is resourceful and sends only concerned data to the cloud, or any mobile application desired. In addition, it boosts the responsiveness, latency and the scalability. Also, GPU offered for processing data demonstrated with performance metrics computed as added value to its size.

Docker offers all libraries and API (s) to allow the deployment and the execution of algorithms and applications in an isolated manner with a more density of algorithms and applications. To get real-time prediction and to afford analytics very advanced, this project has applied a cutting-edge machine learning and deep learning to IOT. [34]

The defined context, edge computing techniques can be employed to address the problems of latency, effectiveness, and to increase the load processing on the Jeston-Nano by bringing the processing into it to detect the wanted object from real streaming data in real-time. Moreover, combining IOT, AI at an edge level and mobile computing grant different applications such as images processing, images recognition, video processing, video recognition, object segmentation, object tracking and object detection etc.

The proposed architecture was analyzed using Edge Artificial Intelligence algorithms and models that has been trained on an NVidia Jeston-Nano, equipped with Tesla P100 card. Then, a Neuronal Stick is attached with one port of Jeston-Nano, also many cables are connected all devices :{ LED, Raspberry PI4}, a performance study was realized in terms of models evaluation (inference time) and accuracy through a test data set of 700 images.[35]

Experimental results illustrated that the Jetson-Nano has performed out 1 sec and accuracy reached approximately of 93%. An optimal result can be obtained that considerate into account the trade-off between reachable accuracy and resulting inference time.

Finally, the proposed model created a metric based on the accuracy of similarity score, an index used for assessing the quality of an image according to a similarity, to identify the performance of proposed system. In fact, training the neural network into images from training set then testing sample of images from testing set that allows to show the score of accuracy very high this can fairly provide good recognition of the wanted object and also offer a precise classification in the future work.

Limitation faced us in this project

As scorpion a dangerous insect, the hunted method for this animal to collect meaningful data is too hard more requires preventive means:

During this work, a considerable waiting-time to start implementation and programming this model because of these reasons:

- Lack of the exit of scorpions from their hibernation because of climatic conditions.
- Time Pressure for having more chances to find volunteer groups help in collection of scorpions.

These difficulties involve factors that provide delay or stopping the collection and analysis of data.

Perspective and Future work for this project

Many extensive works can been defined, as interesting perspective that faced the progress in this dissertation:

- Make a connection with biology research team working on scorpions and collaborate to enhance the dataset because of scorpion symptoms stings vary depending on the species and the amount of venom inoculated.
- Augment the dataset, with collecting many species, so build a new-labeled dataset through the defined species, with increasing the number of it by volunteer contributors.
- Utilize new models once the dataset is enhanced, as classification or segmentation to construct an authenticated model.
- Embedded this project in a robot that can automatically collect scorpions by itself and use it further war for vaccination serum invention.

BIBLOGRAPHY

[1] ‘Tout savoir sur l’intelligence artificielle’, *Microsoft experiences*, Feb. 09, 2018. https://experiences.microsoft.fr/articles/intelligence-artificielle/comprendre-utiliser-intelligence-artificielle/ (accessed May 04, 2021).

[2] ‘Les 5 chiffres à absolument connaître sur l’IA’, *Microsoft experiences*, Jul. 19, 2017. https://experiences.microsoft.fr/business/intelligence-artificielle-ia-business/ia-chiffres-cles/ (accessed May 04, 2021).

[3] ‘What is Edge AI, and Why Enterprises Should Care About It?’, *Matellio Inc*, Jun. 03, 2020. https://www.matellio.com/blog/what-is-edge-ai-and-why-enterprises-should-care-about-it/ (accessed May 07, 2021).

[4] M. K. | D. Scientist, ‘How does Edge AI work?’ https://www.advian.fi/en/blog/how-does-edge-ai-work (accessed May 10, 2021).

[5] ‘What is Edge AI? - Imagimob’. https://www.imagimob.com/blog/what-is-edge-ai (accessed May 07, 2021).

[6] ‘What is Computer Vision & How Does it Work? An Introduction | Adobe XD Ideas’, *Ideas*. https://xd.adobe.com/ideas/principles/emerging-technology/what-is-computer-vision-how-does-it-work/ (accessed May 07, 2021).

[7] C. Nwankpa, W. Ijomah, A. Gachagan, and S. Marshall, ‘Activation Functions: Comparison of trends in Practice and Research for Deep Learning’, *ArXiv181103378 Cs*, Nov. 2018, Accessed: Jun. 13, 2021. [Online]. Available: http://arxiv.org/abs/1811.03378

[8] ‘Machine Learning Overview: Everything You Need to Know’, *My TechDecisions*, Aug. 27, 2019. https://mytechdecisions.com/it-infrastructure/machine-learning/ (accessed Jun. 13, 2021).

[9] ‘Dey, Ayon. “Machine Learning Algorithms: A Review” 7 (2016): 6.’.

[10] ‘10 Machine Learning Methods that Every Data Scientist Should Know | by Jorge Castañón | Towards Data Science’. https://towardsdatascience.com/10-machine-learning-methods-that-every-data-scientist-should-know-3cc96e0eeee9 (accessed Jun. 13, 2021).

[11] J. Brownlee, ‘A Gentle Introduction to Transfer Learning for Deep Learning’, *Machine Learning Mastery*, Dec. 19, 2017. https://machinelearningmastery.com/transfer-learning-for-deep-learning/ (accessed Jun. 13, 2021).

[12] ‘Géron, Aurélien. “Hands-On Machine Learning with Scikit-Learn and TensorFlow,” n.d., 564.’.

[13] ‘Neural Networks and Introduction to Deep Learning , https://www.math.univtoulouse. fr/ besse/Wikistat/pdf/st-m-hdstat-rnn-deep-learning.pdf’.

[14] ‘Zakaria, Magdi, Mabrouka AL-Shebany, and Shahenda Sarhan. Artificial Neural Network: A Brief Overview, 4, no. 2 (2014): 6.’.

[15] David Cheng, MD; Chief Editor: Joe Alcock, MD, MS, ‘Scorpion Envenomation: Background, Pathophysiology, Etiology’, Apr. 2021, Accessed: Jun. 13, 2021. [Online]. Available: https://emedicine.medscape.com/article/168230-overview

[16] S. Sadine, ‘Contribution to the study of scorpion fauna of Northern Sahara, eastern Algeria (El Oued and Ouargla)’, 2012. doi: 10.13140/RG.2.2.14272.33284/1.

[17] Y. Laïd *et al.*, 'Incidence and severity of scorpion stings in Algeria', *J. Venom. Anim. Toxins Trop. Dis.*, vol. 18, pp. 399–410, 2012, doi: 10.1590/S1678-91992012000400008.

[18] 'Jetson Community Projects', *NVIDIA Developer*, Feb. 26, 2019. https://developer.nvidia.com/embedded/community/jetson-projects (accessed Jun. 13, 2021).

[19] 'Getting Started With Jetson Nano Developer Kit', *NVIDIA Developer*, Mar. 05, 2019. https://developer.nvidia.com/embedded/learn/get-started-jetson-nano-devkit (accessed Jun. 13, 2021).

[20] P. Ganesh, 'Object Detection : Simplified', *Medium*, Oct. 07, 2019. https://towardsdatascience.com/object-detection-simplified-e07aa3830954 (accessed Jun. 13, 2021).

[21] 'A Beginner's Guide to Object Detection', *DataCamp Community*, Apr. 19, 2018. https://www.datacamp.com/community/tutorials/object-detection-guide (accessed Jun. 13, 2021).

[22] J. Brownlee, 'A Gentle Introduction to Object Recognition With Deep Learning', *Machine Learning Mastery*, May 21, 2019. https://machinelearningmastery.com/object-recognition-with-deep-learning/ (accessed Jun. 13, 2021).

[23] 'mobilenet-ssd - OpenVINO™ Toolkit'. https://docs.openvinotoolkit.org/2021.2/omz_models_public_mobilenet_ssd_mobilenet_ssd.html (accessed Jun. 13, 2021).

[24] A. G. Howard *et al.*, 'MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications', *ArXiv170404861 Cs*, Apr. 2017, Accessed: Jun. 13, 2021. [Online]. Available: http://arxiv.org/abs/1704.04861

[25] S. Pokhrel, ‘Real-Time Vehicle Detection with MobileNet SSD and Xailient’, *Xailient*, Aug. 25, 2020. https://www.xailient.com/post/real-time-vehicle-detection-with-mobilenet-ssd-and-xailient (accessed Jun. 13, 2021).

[26] R. Khandelwal, ‘SSD : Single Shot Detector for object detection using MultiBox’, *Medium*, Nov. 30, 2019. https://towardsdatascience.com/ssd-single-shot-detector-for-object-detection-using-multibox-1818603644ca (accessed Jun. 13, 2021).

[27] D. Franklin, *dusty-nv/jetson-inference*. 2021. Accessed: Jun. 13, 2021. [Online]. Available: https://github.com/dusty-nv/jetson-inference/blob/19ed62150b3e9499bad2ed6be1960dd38002bb7d/docs/building-repo-2.md

[28] D. Franklin, *dusty-nv/jetson-inference*. 2021. Accessed: Jun. 13, 2021. [Online]. Available: https://github.com/dusty-nv/jetson-inference/blob/19ed62150b3e9499bad2ed6be1960dd38002bb7d/docs/pytorch-collect-detection.md

[29] A. Kurniawan, ‘NVIDIA Jetson Nano I/O Programming’, in *IoT Projects with NVIDIA Jetson Nano: AI-Enabled Internet of Things Projects for Beginners*, A. Kurniawan, Ed. Berkeley, CA: Apress, 2021, pp. 63–83. doi: 10.1007/978-1-4842-6452-2_5.

[30] ‘Performance Metrics for Classification problems in Machine Learning | by Mohammed Sunasra | Medium’. https://medium.com/@MohammedS/performance-metrics-for-classification-problems-in-machine-learning-part-i-b085d432082b (accessed Jun. 13, 2021).

[31] statistiques approved by the government; Institut Pasteur d'Algérie, Sidi Fredj, Staoueli - Alger; Standard: +213 (0) 23 39 37 42,Centre de Prélèvement, Tél : +213 (0) 23 39 37 48

[32] Coronavirus disease (COVID-19) (2020) World Health Organization. https://www.who.int/emergencies/diseases/novelcoronavirus-2019..Accessed November 2020

[33] Xiaoyue Jiang, Abdenour Hadid, Yanwei Pang, Eric Granger; book title:" Deep Learning in Object Detection and Recognition", ISBN-13978-9811506512,Publisher:Springer,28 th Nov 2020.

[34] Anirudh Koul, Siddha Ganju, Meher Kasam; Book title:"Practical Deep Learning for Cloud, Mobile, and Edge: Real-World AI &Computer-Vision Projects Using Python, Keras & TensorFlow", ISBN-13 978-1492034865, Publisher O'Reilly edia, November 12, 2019.

[35] Book title: "IoT Projects with NVIDIA Jetson Nano", ISBN 978-1-4842-6452-2, 1st ed., IX, 123 p. 79 illus., 77 illus. in color. IoT Projects with NVIDIA Jeston Nano AI-Enabled Internet of Things Projects for Beginners,
APS: Algerie Press Service.