TABLE OF CONTENTS :

Chapter 1: Background on Deep Learning for agricultural improvement

Introduction :

The introduction of deep learning in agriculture represents a major step forward in research and innovation. Deep Learning, a branch of artificial intelligence, offers promising possibilities for improving the productivity, sustainability and efficiency of agricultural practices.

Using deep neural networks and complex algorithms, Deep Learning enables the analysis of large amounts of agricultural data, such as satellite images, weather data, crop information and much more. Thanks to this deep analysis, farmers can make informed decisions about crop management, disease prevention, optimising yields and reducing losses.

For example, deep learning can be applied to detect plant diseases early from images, predict yields as a function of multiple variables or even optimise the use of resources by adapting farming practices to the specific conditions of each plot.

History :

The integration of deep learning in the field of agriculture is a relatively recent story, but one that has seen a meteoric rise in recent years. The beginnings of this revolution date back to the 2010s, when technological advances made it possible to process and analyse large, complex data sets, opening the way to new applications in various sectors, including agriculture.

The first applications: towards more precise agriculture

Among the first notable applications of deep learning in agriculture is the development of precision farming systems. These systems use sensors and data analysis to monitor crop and soil conditions in real time, enabling farmers to make more informed decisions about irrigation, fertilisation and pest control.

One of the pioneers in this field is the **"Field Phenomics"** project launched by the University of Pennsylvania in 2013. This project used cameras and sensors to collect detailed data on crop growth, enabling researchers to develop deep learning models capable of identifying diseased or stressed plants.

Towards more sustainable and productive agriculture

In addition to precision farming, deep learning is now being used in a wide range of applications to improve agricultural sustainability and productivity. For example, deep learning algorithms are being used to :

- **Analysing satellite and aerial images** to monitor the condition of crops, detect problem areas and optimise the use of resources.
- **Developing agricultural robots** capable of carrying out tasks such as weeding, harvesting and sorting fruit and vegetables.
- **Predict crop yields**, taking into account factors such as weather, disease and farming practices.
- **Improve water management** by optimising irrigation and reducing wastage.
- **Combating pests and diseases** by developing early detection systems and targeted treatment strategies.

A fast-growing sector with great potential

The use of deep learning in agriculture is still in its infancy, but the potential is immense. Future developments in this field could revolutionise the way we produce and consume food.

Issues :

"How can deep learning make a significant contribution to the optimisation of our business? What are the best practices for increasing yields, reducing the use of chemicals and promoting sustainable, environmentally-friendly agriculture?

"How do we ensure that farmers, particularly in rural areas, have access to a robust IT infrastructure and reliable internet connectivity to deploy deep learning solutions?"

"How can deep learning models, in particular deep neural networks, be made more explainable and interpretable for end users?"

"How to collect and prepare high-quality agricultural data in sufficient quantity to train deep learning models?"

"How can we reduce the cost and complexity of the IT infrastructure needed to deploy deep learning solutions in agriculture?"

These issues raise important questions about the impact and potential of deep learning in agriculture. It encourages reflection on the current challenges facing the agricultural sector and on the potential of technologies based on deep learning to provide effective and sustainable solutions to the urgent challenges of food security and sustainable development through deep learning.

Objective:

The main objective of deep learning in improving agriculture is to optimise agricultural processes through the analysis of massive data. This includes predicting crop yields, detecting plant diseases, managing water and nutrient resources, and automating farming tasks to increase efficiency and productivity.

Chapter 2: Materials and methods

Equipment :

To carry out our project, we need a few basic items of equipment, although there are also more sophisticated items. These materials are :

- *Powerful computer :* To be able to use the computer program without discomfort ;
- *Development environment (jupyter notebook, Google colab) :* To write an optimal programme ;
- *PH meter :* To measure the PH of the soil to be cultivated;
- *Luxmeter :* To define the amount of sunlight ;
- *Sedimentation test:* To determine the type of soil ;

Methods :

Once we've bought all the materials we need, we can get down to work:

1. Install Jupyter Notebook and all the packages on our computer;
2. Write an explanatory text or sketch to better understand the code
3. Writing the code itself;
4. Using the sedimentation test ;

5. Using the luxmeter ;
6. Using the PH meter ;
7. Enter the data in the program
8. Apply the crops represented.

How does the sedimentation test work?

The sedimentation test, also known as granulometric analysis by sedimentation, is a simple and inexpensive method for determining soil texture. It classifies soil according to the proportion of its particles: sand, silt and clay.

Materials required:

- A clear glass jar or bottle with a capacity of 500 ml to 1 litre.
- Distilled or demineralised water.
- A stopwatch or watch.
- A graduated ruler.
- Dispersant (optional).
- A thermometer (optional).

Test steps :

1. **Preparing the soil sample :**
   - Take a representative soil sample from the area you wish to analyse.
   - Dry the soil sample in the open air or in a low-temperature oven (around 105°C) until it is completely dry.
   - Crush the dry soil finely using a mortar and pestle or a sieve.
   - Pass the sieved soil through a 2 mm sieve to remove stones and other large debris.
2. **Preparing the suspension :**
   - Fill the glass jar or bottle halfway with distilled or demineralised water.
   - Add 50 grams of sieved soil to the water.
   - If you are using a dispersant, add a few drops to the suspension.

- Mix the suspension vigorously for 2 minutes using a shaker or by shaking the jar vigorously.

3. **Sedimentation of particles :**

   - Leave the suspension to stand undisturbed at room temperature (around 20°C).
   - Note the time at which you start the sedimentation.
   - After 2 minutes, measure and record the distance between the top of the suspension and the upper limit of the sand deposit (the coarsest layer).
   - After 1 hour, measure and record the distance between the top of the suspension and the upper limit of the silt deposit.
   - After 24 hours, measure and record the distance between the top of the suspension and the top of the clay deposit (the thinnest layer).

Calculating soil texture :

- **Percentage of sand:** Divide the distance of the sand deposit by the total height of the suspension and multiply by 100.
- **Percentage of silt:** Divide the distance between the sand deposit and the silt deposit by the total height of the suspension and multiply by 100.
- **Percentage of clay:** Divide the distance of the clay deposit by the total height of the suspension and multiply by 100.

Interpretation of results :

- **Sandy soil:** If the percentage of sand is greater than 70%, the soil is considered sandy.
- **Silty soil:** If the percentage of silt is between 30 and 70%, the soil is considered to be silty.
- **Clay soil:** If the percentage of clay is greater than 30%, the soil is considered to be clayey.

Chapter 3: Earnings

Presentation of results :

Problem situation :

Before getting started, we created a problem situation to make it easier to create the computer programme. Here is our problem situation:

We need to create a computer programme to display the best crops that can be grown on a given soil. The program will ask the user to enter : The type of soil (clay, sand, silt), the PH of the soil and the amount of sunshine (number of hours of sunshine per day on the soil). Once the user has entered the data, the programme returns the following response next:

- *If the user enters a soil type that is not listed, the programme displays: "TYPE NOT REGISTERED".*
- *If the user specifies a SAND type, PH<7, and less than 6 hours of sunshine, the programme displays: "The crops suitable for this soil are : Carrots, Radishes, Lettuce, Spinach, Bilberries".*
- *If the user specifies a SAND type, PH<7, and more than 6 hours of sunshine, the programme displays: "The crops suitable for this soil are: Blueberries, Tomatoes, Strawberries, Peppers, Bananas, Manioc, Potatoes".*
- *If the user gives a SAND type, PH>7, and less than 6 hours of sunshine, the programme displays: "The crops suitable for this soil are : Beans, Lettuce, Cucumbers, Melon, Strawberries, Raspberries".*
- *If the user specifies a SAND type, PH>7, and more than 6 hours of sunshine, the programme displays: "The crops suitable for this soil are: Lemon, Orange, Watermelon But, Bean".*
- *If the user specifies a type of soil with CLARITY, PH<7, and less than 6 hours of sunshine, the programme displays: "The crops suitable for this soil are : Raspberries, Spinach, Leeks, Potatoes".*
- *If the user specifies a type of soil with CLARITY, PH<7, and more than 6 hours of sunshine, the programme displays: "The crops suitable for this soil are : Blueberries, Strawberries, Tomatoes, Peppers, Green beans".*

- *If the user specifies a CLAY type, PH>7, and less than 6 hours of sunshine, the programme displays: "The crops suitable for this soil are : Carrots, Lettuce, Spinach, Cherry trees, Plum trees".*

- *If the user specifies a CLAY type, PH>7, and more than 6 hours of sunshine, the programme displays: "The crops suitable for this soil are : Tomatoes, Peppers, Beans, Melons, Plums, Sorghum, Durum wheat, Popcorn corn".*

- *If the user specifies a LIMONOUS type, PH<7, and less than 6 hours of sunshine, the programme displays: "The crops suitable for this soil are : Onions, Garlic, Beetroot, Blackcurrants, Leeks".*

- *If the user specifies a LIMONEOUS type, PH<7, and more than 6 hours of sunshine, the programme displays: "The crops suitable for this soil are: Blackcurrant, Apple, Tomato, Pepper, Melon, Blueberry, Banana, Cassava, Potato".*

- *If the user specifies a SLIM soil type, PH>7, and less than 6 hours of sunshine, the programme displays: "The crops suitable for this soil are: Lettuce, Broccoli, Spinach, Leek, Onion, Garlic, Carrot, Beetroot, Parsley, Mint, Basil": Lettuce, Broccoli, Spinach, Leek, Onion, Garlic, Carrot, Beetroot, Parsley, Mint, Basil".*

- *If the user specifies a LIMONEOUS type, PH>7, and more than 6 hours of sunshine, the programme displays: "The crops suitable for this soil are : Aubergines, Beans, Watermelons, Strawberries, Raspberries, Wheat, Maize, Rice".*

Presentation of the code and its algorithm :

Based on this situation, we created the code. Given that we are dealing with deep learning, we used the K-Nearest Neighbors (**KNN)** algorithm, which is a supervised learning algorithm used for classification and regression. It is a
machine learning algorithm that is relatively simple and intuitive to understand. It is used in the code to determine the most likely crop for a given soil.

This is our computer programme:

```
[1]: from collections import Counter
     import math

     def distance_euclidienne(point1, point2):
         return math.sqrt(sum((x - y) ** 2 for x, y in zip(point1, point2)))

     def afficher_meilleures_cultures(type_sol, ph, ensoleillement, k=3):
         """
         Affiche les meilleures cultures réalisables sur un sol donné.

         Args:
             type_sol: Le type de sol (argileux, sableux, limoneux).
             ph: Le pH du sol.
              ensoleillement: Le nombre d'heures de soleil par jour sur le sol.
             k: Le nombre de voisins à considérer pour KNN (par défaut 3).

         Returns:
             None.
```

```
data

    type_sol            data :

point_a_classifien      ph, eusoleillement

point_a_classifier      ph, eusoleillement

distances

    i      range(1in(data type_so1            ) ):
    pointdata       datatypesol             , data s t ype_so1                      i
    dist     di st anc e_euc 1id ienne ( point_a_c 1as sif ie n, point_data )
    di stance s .        ( (dist, data t ype_so1                  î ) )

distances .      ( keyx         : x       )

k p1u s pnoches_voi si ns      di stances : k

cooked some
     cooked      k_nearest_neighbours:
    crops.              (cults)

    culture, countCounter (cu1tu res ) . point          ():
    (                 cuit urecount              k

type_so1    input(
ph    float(input(                                 ))
sunshine1lement  +1oat(input(                                                    ))

display_best_cultures(soil_type, ph, sunshine)
```

Interpreting the result :

We'll start by explaining our code:

```python
from collections import Counter

import math
```

These lines import the Counter functions for counting occurrences in a list and math for mathematical operations such as the square root.

```python
def distance_euclidean(point1, point2):

    return math.sqrt(sum((x - y) ** 2 for x, y in zip(point1, point2)))
```

This function calculates the Euclidean distance between two points using the formula sqrt((x1 - x2)^2 + (y1 - y2)^2). It takes two points as input and returns the distance between them.

```python
def afficher_meilleures_cultures(type_sol, ph, ensoleillement, k=3):
```

This function takes as input the type of soil, the pH, the number of hours of sunshine and the parameter k (number of neighbours to be considered for KNN, default 3).

```python
if type_sol not in données:

    print("TYPE NON ENREGISTRE")

    return
```

This part checks whether the soil type entered by the user is present in the data dictionary. If it is not, a message is displayed and the function terminates.

```python
point_a_classifier = [ph, sunshine]
```

A point to be classified is created from the pH and sunlight values entered by the user.

```python
distances = []

for i in range(len(données[type_sol]["pH"])):

    point_data = [données[type_sol]["pH"][i], données[type_sol]["ensoleillement"][i]]

    dist = distance_euclidienne(point_a_classifier, point_data)

    distances.append((dist, data[soil_type]["crops"][i]))
```

This part calculates the Euclidean distance between the point to be classified and each point in the data for the specified soil type. The distances are stored with the associated crops.

distances.sort(key=lambda x: x[0])

The distances are sorted in ascending order to obtain the nearest neighbours first.

k_nearest_neighbours = distances[:k]

The k nearest neighbours are selected from the sorted distances.

cultures = []

for _, cults in k_nearest_neighbours:

cultures.extend(cults)

A majority vote is taken by combining the crops of the k nearest neighbours.

print("The appropriate crops for this soil are:") for

culture, count in Counter(cultures).items():

print(f"- {culture} ({count}/{k} votes)")

The most frequent crops among the k nearest neighbours are displayed, along with the number of votes for each crop.

Code testing :

The values for soil type, pH and sunshine are entered by the user, then the ***display_best_cultures*** is called with these values as input.

This code combines classification logic based on the k nearest neighbours (KNN) with Euclidean distance to recommend the best crops according to soil type, pH and sunshine.

After these explanations, we simulated with some data. Here are the results:

```
Entrez le type de sol (argileux, sableux, limoneux):  limoneux
Entrez le pH du sol:  9
Entrez le nombre d'heures de soleil par jour sur le sol:  7
Les cultures appropriées pour ce sol sont :
- Aubergines (1/3 votes)
- Haricots (1/3 votes)
- Pastèques (1/3 votes)
- Fraises (1/3 votes)
- Framboises (1/3 votes)
- Blé (1/3 votes)
- Maïs (1/3 votes)
- Riz (1/3 votes)
- Laitue (1/3 votes)
- Brocolis (1/3 votes)
- Epinard (1/3 votes)
- Poireau (1/3 votes)
- Oignon (1/3 votes)
- Ail (1/3 votes)
- Carotte (1/3 votes)
```

```
Entrez le type de sol (argileux, sableux, limoneux):  APILEUX
Entrez le pH du sol:  5
Entrez le nombre d'heures de soleil par jour sur le sol:  9
TYPE INEXISTANT
```

```
Entrez le type de sol (argileux, sableux, limoneux):  sableux
Entrez le pH du sol:  5
Entrez le nombre d'heures de soleil par jour sur le sol:  5
Les cultures appropriées pour ce sol sont :
- Carottes (1/3 votes)
- Radis (1/3 votes)
- Laitue (2/3 votes)
- Epinards (1/3 votes)
- Myrtilles (2/3 votes)
- Haricot (1/3 votes)
- Concombres (1/3 votes)
- Melon (1/3 votes)
- Fraises (2/3 votes)
- Framboises (1/3 votes)
- Tomates (1/3 votes)
- Piment (1/3 votes)
- Banane (1/3 votes)
- Manioc (1/3 votes)
- Patate (1/3 votes)
```

Chapter 4: Conclusion/ Suggestions

Conclusion:

Deep Learning has the potential to revolutionise agriculture by making it more accurate, sustainable and productive. Despite the challenges ahead, it is clear that Deep Learning is a powerful tool that can help address the major food security and sustainability challenges facing the world. By investing in research and development, and ensuring that this technology is accessible to all farmers, we can harness the power of Deep Learning to shape a more prosperous and sustainable agricultural future for all.

Proposed improvements :

Deep Learning has opened up exciting new prospects for transforming agriculture and meeting the challenges of food security and sustainability. However, for this revolutionary technology to reach its full potential, certain challenges need to be addressed and improvement strategies implemented. Here are some key suggestions for improving deep learning for agriculture:

1. Increase the quality and quantity of data :

- **Collect more comprehensive and diversified agricultural data:** This includes data on soils, crops, climate, farming practices, yields and environmental impacts.
- **Ensuring data quality and consistency:** Implementing rigorous data collection and processing protocols to guarantee the reliability and validity of the information.
- **Promote data sharing and collaboration:** Encourage collaboration between researchers, farmers, businesses and government agencies to create open and accessible agricultural datasets.

2. Strengthening expertise and training in deep learning :

- **Developing specialised training programmes:** Offer training and workshops for agronomists, computer scientists and farmers to provide them with the necessary skills in deep learning and artificial intelligence applied to agriculture.

- **Promoting continuous learning and knowledge exchange:** Encouraging participation in conferences, workshops and online communities to promote the updating of knowledge and the exchange of experience between practitioners and researchers.

- **Attracting and retaining talent:** Creating an environment conducive to innovation and deep learning research to attract and retain experts in the field.

3. Improving the adaptation and generalisation of models :

- **Developing more robust and adaptable deep learning models**
  Design models capable of adapting to the variability of agricultural conditions and the specific environments of each region.

- **Incorporate agronomic and local knowledge:** Incorporate agricultural knowledge and practices into the development of models to improve their relevance and effectiveness in realistic agricultural contexts.

- **Validate and rigorously test the models:** Evaluate the performance of the models under real conditions and on representative data sets before their large-scale deployment.

4. Promoting an inclusive and equitable approach:

- **Making deep learning accessible to all farmers:** Designing technological solutions that are affordable and easy to use, taking into account the different constraints and realities of farmers, particularly in developing countries.

- **Promoting digital inclusion and connectivity:** Guaranteeing access to the Internet and digital infrastructures in rural areas to enable farmers to benefit from deep learning technologies.

- **Building awareness and trust:** Educating farmers about the benefits and limitations of deep learning and encouraging open dialogue to address issues of ethics and social acceptability.

5. Supporting ongoing research and innovation :

- **Investing in fundamental and applied research in deep learning for agriculture:**
  Supporting research initiatives that explore new applications

and are developing more efficient algorithms tailored to the specific needs of agriculture.

- **Encouraging public-private partnerships:** Fostering collaboration between research institutions, private companies and farming organisations to accelerate the development and deployment of innovative deep learning solutions.
- **Exploring new applications and integrations:** Supporting research into the application of deep learning to new areas of agriculture, such as agricultural robotics, supply chain management and the fight against climate change.

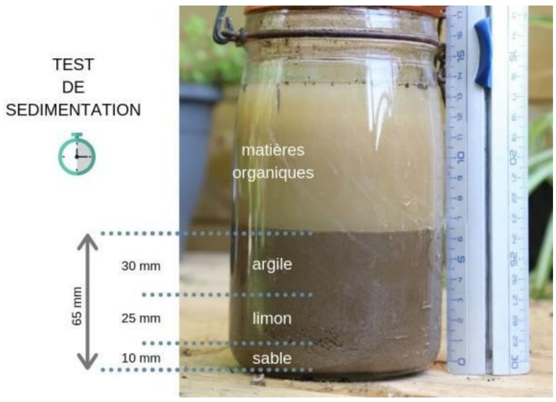
By addressing these challenges and implementing these suggestions, we can harness the full potential of deep learning to transform agriculture and shape a more sustainable, productive and equitable future for the global food system.

Some useful images:

Jupyter Notebook

Computer

A PH meter



A

luxmeterA Sedimentation Test

Bibliography :

"Deep learning and agriculture" by AgroTIC
: https://www.agrireseau.net/documents/Document_101642.pdf

"Improvement of crop monitoring systems using multi-source remote sensing and deep learning techniques" by CIRAD
: https://www.agrireseau.net/documents/Document_101642.pdf

"Deep learning and intelligent sensors in agriculture" by Agri-Réseau
: https://www.agrireseau.net/documents/Document_101642.pdf

"Deep learning for precision agriculture" by Innova IT
: https://www.sciencedirect.com/science/article/am/pii/S2667305322000400

"Artificial intelligence serving agriculture" by Actuagri
: https://www.actuagri.fr/

The 2020 report on AI for agriculture in Africa: https://aiforgood.itu.int/about-ai- for-good/un-ai-actions/ifad/

FAO e-Agriculture Programme: https://www.fao.org/home/en

GIZ: Building AI capacity for agriculture:
https://www.giz.de/en/workingwithgiz/93909.html

OECD : Artificial intelligence and agriculture : https://www.oecd.org/digital/artificial-intelligence/

The AI World Forum for agriculture: https://blogs.ifas.ufl.edu/news/2022/10/25/ai-center-coming-to-gulf-coast-research-and-education-center-will-provide-boost-for-agriculture/

AI ethics for agriculture: A guide: https://www.sciencedirect.com/science/article/pii/S1360138523004156

The 2021 report on the state of AI in agriculture: https://www.dtnpf.com/agriculture/web/ag/news/business-inputs/article/2023/11/14/artificial-intelligence-creates-ag

The Dataverse project for agriculture: https://learn.microsoft.com/en-us/power-apps/maker/data-platform/data-platform-intro

The 2020 report on AI for agriculture in Africa: https://aiforgood.itu.int/about-ai- for-good/un-ai-actions/ifad/

FAO e-Agriculture Programme: https://www.fao.org/home/en

The CGIAR LEAP programme: https://www.cgiar.org/initiative/sustainable-animal-productivity/