PEOPLE'S DEMOCRATIC REPUBLIC OF ALGERIA
Ministry of Higher Education and Scientific Research
Mohamed Kheider university of Biskra
Faculty of Exacts sciences and Sciences of nature and life
Computer science department

Master thesis

Presented to obtain the academic Master's in

Computer science

Speciality : Computer science applied to business management (*MIAGE*)[1]
AND Software engineering and distributed systems (*GLSD*)[2]

Detection and identification of original honey using AI

---

presented by
Lastab *Anfal*[1]
Elgarni *Fatiha*[2]
supervised by :
Dr. Abdelli Adel
Dr. Ammari Asma

Defended in front of the jury composed of:

Dr.	Saouli Rachida	Associate Professor	at UMKB	President
Dr.	ABDELLI Adel	Associate Professor	at UMKB	Supervisor
Dr.	AMMARI Asma	Associate Professor	at UMKB	Co-Supervisor
Dr.	Bettira Roufaida	Assistant Professor	at UMKB	Examiner

Academic year 2024/2025

Dedication

We dedicate this work to our dear fathers, who have always been our pillars of support through their constant care and boundless encouragement. Their presence by our side has given us the strength and resilience to overcome every challenge.

We also extend special thanks to our beloved mothers, whose unwavering love and tireless efforts created a nurturing environment, allowing us to focus and persevere. Their selfless support has left a profound mark on our lives.

To all our friends who have stood by us with encouragement—and to whom we wish continued success—we are deeply grateful.

Thank you!

Lastab Anfal and ELgarni Fatiha

Acknowledgment

With great pride and humility, we extend our deepest gratitude to Allah Almighty, whose grace and mercy have enabled this significant achievement. His unwavering support has been the light guiding our path to success.

We also express our sincere appreciation to our esteemed supervisor, Dr. Abdeli Adel, and our co-supervisor, Dr. Asma Ammari, for generously sharing their extensive knowledge and providing invaluable guidance throughout this research. Their insights and direction have profoundly impacted our academic growth and equipped us with essential mentorship skills.

We are deeply grateful to the esteemed committee members for evaluating our work and offering constructive feedback. Their time and expertise have motivated us to pursue our academic journey with renewed determination.

Finally, we would like to thank all those who contributed to this journey—whether through support, encouragement, or belief in our abilities. Their kindness has been the foundation of our accomplishment.

To each of you, we say from the depths of our hearts.

Thank you

Lastab Anfal and ELgarni Fatiha

Abstract

This project aims to develop a reliable and efficient system for authenticating honey and determining its botanical origin. The system identifies the floral sources from which the nectar was collected by analyzing the pollen content in the honey. To achieve this, advanced analytical methods are employed to examine the pollen, helping to detect possible adulteration and ensure that the honey is not mixed with artificial substances or lower-quality alternatives.

By identifying the botanical sources, the system also enhances product traceability, providing essential information about the honey's geographical and floral origin. This authentication process not only maintains the integrity of honey but also plays a crucial role in protecting consumers from fraudulent products. By offering verified information about the honey's origin and quality, the system boosts consumer confidence and enables laboratories to differentiate authentic honey from adulterated products.

Ultimately, this project contributes to strengthening trust among honey producers, regulatory authorities, and consumers by ensuring that honey available in the market meets established quality standards and remains free from contamination or misleading claims.

Our system assists laboratories in classifying pollen by automating this tedious and time-consuming task. It was built using convolutional neural networks (CNNs), a deep learning method known for its effectiveness in image classification. The system achieved an accuracy of 91% in classifying 23 different types of pollen. This AI model was integrated into a mobile application, which can later be connected to a microscope to assist laboratories in classifying honey pollen samples.

Keywords: Honey authentication, deep learning, pollen analysis, adulteration detection.

Résumé

Ce projet vise à développer un système fiable et efficace d'authentification du miel et de détermination de son origine botanique. Ce système identifie les sources florales d'où provient le nectar en analysant la teneur en pollen du miel. Pour ce faire, des méthodes d'analyse avancées sont utilisées pour examiner le pollen, permettant de détecter une éventuelle falsification et de garantir que le miel n'est pas mélangé à des substances artificielles ou à des alternatives de moindre qualité.

En identifiant les sources botaniques, le système améliore également la traçabilité du produit, fournissant des informations essentielles sur l'origine géographique et florale du miel. Ce processus d'authentification préserve non seulement l'intégrité du miel, mais joue également un rôle crucial dans la protection des consommateurs contre les produits frauduleux. En fournissant des informations vérifiées sur l'origine et la qualité du miel, le système renforce la confiance des consommateurs et permet aux laboratoires de distinguer le miel authentique des produits falsifiés.

En fin de compte, ce projet contribue à renforcer la confiance des producteurs de miel, des autorités réglementaires et des consommateurs en garantissant que le miel disponible sur le marché répond aux normes de qualité établies et est exempt de contamination ou d'allégations trompeuses.

Notre système aide les laboratoires à classer le pollen en automatisant cette tâche fastidieuse et chronophage. Il a été développé à l'aide de réseaux de neurones convolutifs (CNN), une méthode d'apprentissage profond reconnue pour son efficacité dans la classification d'images. Le système a atteint une précision de 91 % dans la classification de 23 types de pollen différents. Ce modèle d'IA a été intégré à une application mobile, qui peut ensuite être connectée à un microscope pour aider les laboratoires à classer les échantillons de pollen de miel.

Mots-clés : Authentification du miel, apprentissage profond, analyse du pollen, détection de falsification.

ملخص

يهدف هذا المشروع إلى تطوير نظام موثوق وفعال للتحقق من صحة العسل وتحديد أصله النباتي. يحدد النظام المصادر الزهرية التي جُمِع منها الرحيق من خلال تحليل محتوى حبوب اللقاح في العسل. ولتحقيق ذلك، تُستخدم أساليب تحليلية متطورة لفحص حبوب اللقاح، مما يساعد على كشف أي غش محتمل وضمان عدم خلط العسل بمواد صناعية أو بدائل رديئة الجودة.

من خلال تحديد المصادر النباتية، يُعزز النظام أيضًا إمكانية تتبع المنتج، موفرًا معلومات أساسية حول أصل العسل الجغرافي والزهري. لا تقتصر عملية التحقق هذه على الحفاظ على سلامة العسل فحسب، بل تلعب أيضًا دورًا حاسمًا في حماية المستهلكين من المنتجات المزيفة. من خلال توفير معلومات موثقة حول أصل العسل وجودته، يعزز النظام ثقة المستهلك ويُمكّن المختبرات من التمييز بين العسل الأصلي والمنتجات المغشوشة.

في نهاية المطاف، يُسهم هذا المشروع في تعزيز الثقة بين منتجي العسل والهيئات التنظيمية والمستهلكين من خلال ضمان أن العسل المتوفر في السوق يفي بمعايير الجودة المعمول بها ويبقى خاليًا من التلوث أو الادعاءات المضللة.

يساعد نظامنا المختبرات على تصنيف حبوب اللقاح من خلال أتمتة هذه المهمة الشاقة والمستهلكة للوقت. بُني النظام باستخدام الشبكات العصبية التلافيفية (CNNs)، وهي طريقة تعلّم عميق معروفة بفعاليتها في تصنيف الصور. حقق النظام دقةً بلغت 91% في تصنيف 23 نوعًا مختلفًا من حبوب اللقاح. دُمج نموذج الذكاء الاصطناعي هذا في تطبيق جوال، يمكن توصيله لاحقًا بمجهر لمساعدة المختبرات على تصنيف عينات حبوب لقاح العسل.

الكلمات المفتاحية: مصادقة العسل، التعلم العميق، تحليل حبوب اللقاح، كشف الغش

Contents

General Introduction 23

Chapter 1: Background **26**

1.1 Definition of Honey 27

1.2 Importance of Honey 27

1.2.1 Nutritional Value 27

1.2.2 Medicinal and Therapeutic Benefits 28

1.2.3 Economic and Environmental Importance 28

1.2.4 Cultural and Religious Significance 28

1.3 The issue of Honey Adulteration 29

1.3.1 Definition of Honey Adulteration 29

1.3.2 Methods of Honey Adulteration 29

1.3.3 Impact of Honey Adulteration 30

1.4 Challenges in Detecting Natural vs. Adulterated Honey 30

1.4.1 Variability in Natural Honey Composition 30

1.4.2 Advanced Techniques Used in Honey Modification 31

1.4.3 Limitations of Traditional Detection Methods 31

1.4.4 Inconsistent Honey Regulations and Standards 31

1.4.5 High Costs and Accessibility of Advanced Testing 32

1.4.6 Consumer Awareness and Market Challenges 32

1.5 Limitations of Traditional Honey Authentication 32

1.5.1 Physical Tests Are Inaccurate and Unreliable 32

1.5.2 Limitations of Chemical Analysis Methods 33

1.5.3 Microscopic and Pollen Analysis Challenges 35

1.5.4 Regulatory and Standardization Issues 36

1.5.5 Economic and Accessibility Challenges 37

1.6 Palynology 37

1.6.1 Pollen 37

1.6.2 Composition of a pollen grain 38

1.6.3 Structure of a Pollen Grain 39

1.6.4 Main types of pollen 39

1.6.5 Pollen Types and Corresponding Plants 41

1.6.6 Pollen Analysis in Honey Authentication 41

1.7 Key Features for Honey Classification 42

1.7.1 Microscopy-Based Classification 42

1.7.2 Spectral Analysis 43

1.8 The role of artificial intelligence in food authentication 45

1.8.1 Definition of Food Authentication 45

1.8.2 The role of artificial intelligence 45

1.9 Methods Needed for an AI-Based Approach 46

1.9.1 Importance of Using AI for Detecting Honey (Natural and Adulterated) 46

1.9.2 Machine Learning for Honey Authentication 46

1.9.3 Deep Learning for Honey Authentication 46

1.9.4 Advantages of AI Over Traditional Methods 46

1.10 Types of Honey Analyzed Using Machine Learning 47

1.10.1 Machine Learning for Natural Honey Classification 47

1.10.2 Machine Learning for Adulterated Honey Classification 47

1.10.3 Machine Learning for Processed Honey Classification 48

1.11 Data sources 48

1.11.1 Microscopy-Based Analysis 48

1.11.2 Spectral Data Acquisition 48

1.11.3 Chemical Composition Analysis 49

1.11.4 Machine Learning and AI-Driven Data Utilization 49

1.12 AI techniques used 49

1.12.1 Image-Based Structural Analysis 49

1.12.2 Mathematical Discrimination Models 50

1.12.3 Layered Decision Frameworks 50

1.12.4 Comprehensive Data Fusion 50

1.13 Conclusion 50

Chapter 2: Artificial Intelligence and Machine Learning **51**

2.1 Introduction 52

2.2 Artificial Intelligence 52

2.3 Machine Learning 52
2.3.1 Machine Learning Techniques 52
2.3.1.1 Supervised Learning 53
2.3.1.2 Unsupervised Learning 54
2.3.1.3 Reinforcement Learning 55
2.3.1.4 Semi-Supervised Learning 55
2.3.2 Regression 56
2.3.3 Classification 56
2.4 Deep learning 57
2.4.1 Definition 57
2.4.2 Difference Between Machine Learning and Deep Learning 57
2.4.3 Application areas of deep learning 58
2.4.4 Deep learning architecture 58
2.5 Artificial neural networks 59
2.5.1 Biological neuron 59
2.5.2 Artificial neuron 60
2.5.3 Artificial neural networks 60

2.5.4 Activation function 61

2.6 Convolutional neural networks 64

2.6.1 Convolutional layer: 64

2.6.2 Pooling layer: 66

2.6.3 Fully connected layer (dense layer): 67

2.7 Popular CNN Architectures 68

2.7.1 VGGNet 68

2.7.2 EfficientNet 69

2.7.3 LeNet 69

2.7.4 MobileNet 70

2.7.5 ResNet 70

2.8 Conclusion 71

Chapter 3: System Design, Implementation and Results **72**

3.1 Introduction 73

3.2 Foctionalty architecture 73

3.3 Material and methods 74

3.3.1 Detailed CNN architecture 74
3.3.2 Data description 75
3.3.3 Data preprocessing and augmentation 76
3.3.4 Test 79
3.3.5 Transfer learning 79
3.4 Model training and results 80
3.4.1 Results 82
3.5 Implementation 83
3.6 Mobile Application Development 84
3.6.1 Definition Of Mobile Applications : 84
3.6.2 Why Mobile Applications Choice ? 85
3.6.3 The Pattern Design Used 85
3.6.4 Implementation tools and languages 87
3.6.5 Conception Of The Application 95
3.7 Application graphical interfaces 98
3.7.1 Splash Screen : 99
3.7.2 Home Page : 99

3.7.3 Sing up : 100

3.7.4 Login Page : 104

3.7.5 CameraPage 107

3.7.6 Analyzing Result Page 108

3.7.7 History Page : 109

3.7.8 Chat Page : 109

3.7.9 Profile Page 110

3.7.10 Tokens Page : 110

3.7.11 Admin : 112

3.8 Conclusion: 117

General Conclusion **119**

List of Tables

1.1 Types of Pollen and Corresponding Plants 41

1.2 The mineral constituents of honey (mg/kg) 44

1.3 Physicochemical Analysis Results of some Honey Samples 44

3.1 Hyperparameters used for model training 80

3.2 Classification Report 82

List of Figures

1.1 Honey Composition 27

1.2 Benefit of pure honey 28

1.3 Honey Adulteration 29

1.4 Home Honey Test Methods 33

1.5 Hydroxymethylfurfural (HMF) Honey Test 35

1.6 Ultrafiltration 36

1.7 Some pollen grains from the flora of Africa 38

1.8 Composition of a pollen grain 39

1.9 Structure of a Pollen Grain 39

1.10 Somme types of pollen 40

1.11 Pollen types from floral visiting bees' bodies of Solanum 43

1.12 The Kaleidoscopic Colors of Honey Under a Microscope 43

2.1 Different techniques from AI. 53

2.2 Supervised learning workflow [67]. 54

2.3 Unsupervised learning workflow [67]. 55

2.4 Renforcement learning workflow [67]. 56

2.5 The sub-branches of artificial intelligence 57

2.6 Feature extraction area in machine deep learning [8]. 58

2.7 Multi-layer perceptron [73]. 59

2.8 Biological neuron. 60

2.9 Artificial neuron. 60

2.10 An artificial neural network. 61

2.11 ReLU Function. 62

2.12 Sigmoid function. 63

2.13 Tanh Function. 63

2.14 CNN architecture [22] 64

2.15 How machines understand colored images. 65

2.16 Convolution operation [88]. 66

2.17 Types of polling operation [94]. 67

2.18 Fully connected layer [1]. 68

2.19 Basic architecture of VGGNet 69

2.20 EfficientnetB0 architecture 69

2.21 Basic architecture LeNet-5 70

2.22 MobileNet architecture 70

2.23 ResNet architecture 71

3.1 Global architecture. 74

3.2 Global CNN architecture. 75

3.3 Pollen classes. 76

3.4 Deployment Model use fastAPI and Render 83

3.5 app mobile for pollen detection. 85

3.6 Model-View-Controller pattern. 86

3.7 Architecture of Flutter-Firebase platform 87

3.8 python 88

3.9 OpenCV [81] 88

3.10 Matplotlib [71] 89

3.11 NumPy [79] 89

3.12 Scikit-learn [84] 90

3.13 TensorFlow [17] 90

3.14 Keras [24] 90

3.15 Pandas [72] 91

3.16 Android Studio [44] 91

3.17 Flutter [47] 91

3.18 Dart [45] 92

3.19 Firebase 92

3.20 FastAPI [91] 92

3.21 render [12] 93

3.22 DrawSQl [29] 93

3.23 Google Colab [48] 93

3.24 OverLeaf [82] 94

3.25 Github [42] 94

3.26 Github [37] 95

3.27 Usecases of the system 95

3.28 Sequence diagram of login. 97

3.29 Database CDM. 98

3.30 splash screen 99

3.31 home page 99

3.32 Sign-up page 100

3.33 already used email 100

3.34 Bad formatted email 101

3.35 empty fields 101

3.36 some empty fields 102

3.37 password not much 102

3.38 weak password 103

3.39 Account created successfully interface 103

3.40 User guide interface 104

3.41 login page 104

3.42 incorrect email 105

3.43 Forgotten password interface 105

3.44 Creating a new password interface 106

3.45 Successfully password changed interface 106

3.46 account not approved 107

3.47 cropping image 108

3.48 Analyzing pollens interface 108

3.49 pollen analyzing result 109

3.50 user history 109

3.51 user chatting with admin 110

3.52 user profile 110

3.53 purchase tokens page 111

3.54 incomplete cart 111

3.55 confirm purchase 112

3.56 confirm purchase 112

3.57 admin login 113

3.58 admin page 114

3.59 search bar 114

3.60 User details interface 115

3.61 User details interface 115

3.62 validated users list by admin 116

3.63 profile admin page 116

3.64 Profits monitoring 117

3.65 chat page 117

General Introduction

Honey is a highly valuable natural product due to its nutritional, medicinal, and economic benefits. It has been used for centuries as a food source and a natural remedy because of its antioxidant and antimicrobial properties. In addition to its health benefits, honey production plays a significant role in the economy, supporting beekeepers and agricultural industries worldwide.

However, in recent years, concerns have increased regarding the purity and authenticity of honey, mainly due to widespread adulteration. Large quantities of counterfeit or diluted honey have flooded the markets, mixed with artificial sweeteners such as sugar syrup, glucose, and synthetic additives, misleading consumers and undermining the reputation of genuine honey producers. This issue not only leads to a loss of consumer trust but also causes economic losses for farmers and beekeepers.

To ensure honey quality and purity, several traditional verification methods have been developed, including chemical analysis, spectroscopic examinations, and microscopic pollen analysis. Although these methods are effective, they often require specialized equipment, scientific expertise, and significant time, making large-scale verification difficult. Furthermore, the increasing sophistication of honey adulteration techniques has made it even harder to detect counterfeit honey using conventional methods.

With advancements in artificial intelligence (AI) and machine learning (ML), new opportunities have emerged for automating and improving honey authentication. Deep learning models, such as Convolutional Neural Networks (CNNs), can analyze microscopic images of honey, while machine learning algorithms can detect patterns in chemical and spectral data. Additionally, hyperspectral imaging, image recognition, and advanced classification techniques can enhance the accuracy, efficiency, and scalability of the verification process.

Moreover, can also determine the floral origin of honey by analyzing pollen composition. Identifying the specific flowers that bees have pollinated provides valuable information about honey authenticity, geographical origin, and potential medicinal properties. This feature adds an extra layer of verification, ensuring both purity and traceability in honey production. This research seeks to answer the following key questions :

- **How can machine learning be utilized to develop an model for honey authentication?**
- **How can AI technologies help determine the botanical origin of honey ?**

To achieve these objectives, the study will focus on :

- **Building a dataset containing microscopic images of honey samples, including both pure and adulterated varieties, along with pollen data from different floral sources.**
- **Training AI models to classify honey based on its authenticity and botanical origin.**

Through this research, we aim to develop a reliable and user-friendly tool to assist producers, consumers, and regulatory authorities in verifying the quality of honey and ensuring its authenticity. Additionally, it provides accurate information about the botanical source of honey, enhancing the reliability of honey products in global markets.

The project will be structured into three main chapters :

- **Chapter 1: Background of the Work**
  This chapter provides an overview of honey composition, its economic and nutritional importance, and common adulteration techniques. Additionally, it explores traditional methods used for honey authentication, including chemical analysis, spectroscopic techniques, and microscopic pollen identification.
- **Chapter 2: Deep Learning and Convolutional Neural Networks .**
  This section introduces key AI and ML concepts relevant to the project, including deep learning techniques such as Convolutional Neural Networks (CNNs) for image analysis. It also reviews existing research on AI-driven honey authentication and discusses the advantages of using machine learning over conventional verification methods.
- **Chapter 3: System design, implementation and results**
  This chapter focuses on the design and implementation of the proposed system. It details the dataset preparation, model selection, training process, and evaluation metrics. Additionally, it explains how the system classifies honey samples based on authenticity and floral origin, ensuring a scalable and efficient verification process. Also, in this chapter we explained how we built the application and illustrated the tools and the technologies used in the development process.

Chapter 1: Background

1.1 Definition of Honey

Honey is a naturally occurring sweet substance created by honeybees (Apis mellifera) through the collection and transformation of nectar from flowers or plant secretions. Once gathered, the bees process the nectar with enzymes, store it within the honeycomb, and allow it to undergo evaporation. This natural process results in the formation of a thick, golden liquid, commonly known as honey.

According to the Codex Alimentarius, an international food standard jointly established by the Food and Agriculture Organization (FAO) and the World Health Organization (WHO), honey is defined as:

The natural sweet substance produced by honey bees from the nectar of plants or from secretions of living parts of plants, which the bees collect, transform, and combine with specific substances of their own, deposit, dehydrate, store, and leave in the honeycomb to ripen and mature[25].

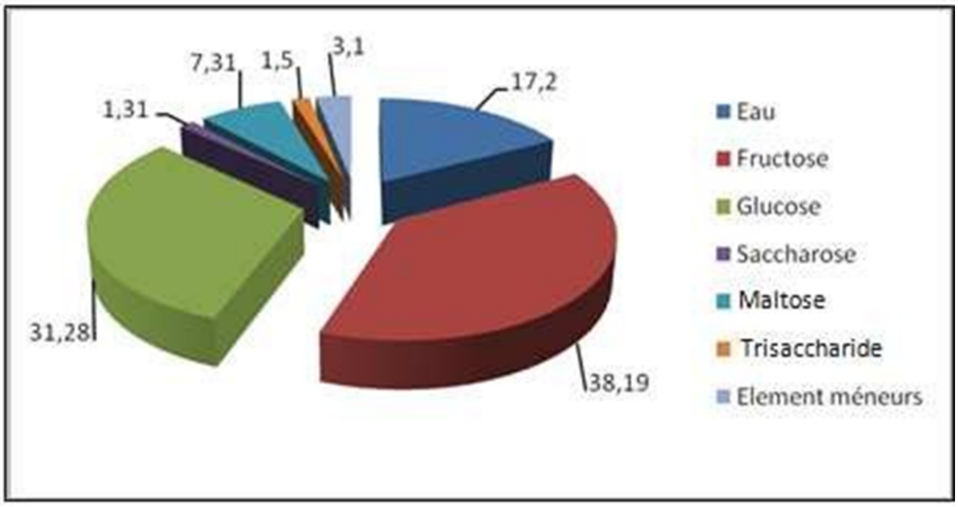


Figure 1.1: Honey Composition
[50]

1.2 Importance of Honey

1.2.1 Nutritional Value

- Honey is a rich source of natural carbohydrates, primarily fructose (38
- It contains essential vitamins, including B-complex vitamins (B1, B2, B6), and key minerals like calcium, magnesium, potassium, and iron, supporting overall well-being.
- The presence of antioxidants, such as flavonoids and phenolic compounds, helps protect cells from oxidative stress and supports immune health[76].

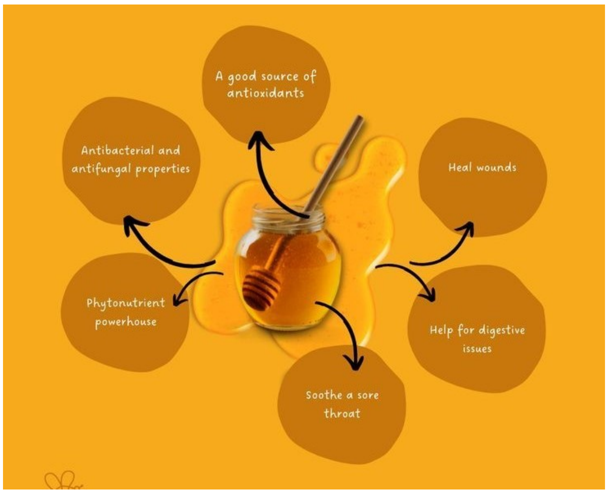


Figure 1.2: Benefit of pure honey

1.2.2 Medicinal and Therapeutic Benefits

- **Antibacterial & Antifungal Properties:** Honey naturally inhibits bacterial and fungal growth due to its low moisture content and hydrogen peroxide production, making it effective for wound care and infection prevention.
- **Cough and Sore Throat Relief:** Honey is a well-known natural remedy for cough suppression and soothing throat irritation.
- **Digestive Health:** Honey supports a healthy gut microbiome, alleviates acid reflux, and acts as a prebiotic, promoting beneficial bacteria.
- **Wound Healing:** Certain varieties, such as Manuka honey, are widely used in medical applications for treating burns, ulcers, and skin injuries. [57]

1.2.3 Economic and Environmental Importance

- The honey industry supports millions of beekeepers worldwide, contributing significantly to local and global economies.
- Bees play a crucial role in pollination, helping sustain agricultural productivity and promoting biodiversity.
- Beyond honey, beekeeping produces valuable byproducts like beeswax, royal jelly, and propolis, which are widely used in cosmetics, pharmaceuticals, and food processing. [36]

1.2.4 Cultural and Religious Significance

- Honey is historically associated with health, purity, and longevity, holding deep significance in various cultures.

- It is referenced in ancient texts, including the Quran, the Bible, and Ayurveda, where it is described as a remedy for ailments and a symbol of nourishment and healing.

1.3 The issue of Honey Adulteration

1.3.1 Definition of Honey Adulteration

Honey adulteration involves the introduction of foreign substances into honey, which diminishes its purity, nutritional content, and health benefits. This fraudulent practice is often carried out to increase production volume and maximize profits.

Common substances used to dilute or modify honey include:

- Industrial sugar syrups, such as high-fructose corn syrup (HFCS), cane sugar syrup, and rice syrup
- Molasses and glucose-based solutions that imitate natural honey's sweetness.
- Artificial sweeteners and synthetic chemicals, can alter taste and texture.
- Excessive water content, which increases weight but reduces quality. [33]

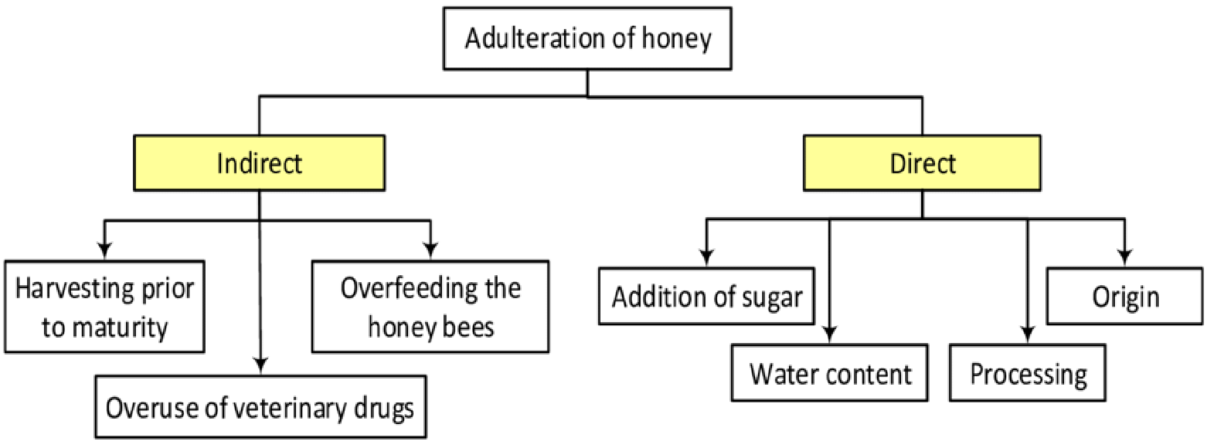


Figure 1.3: Honey Adulteration

1.3.2 Methods of Honey Adulteration

- **Blending with Industrial Syrups:** Large-scale manufacturers often mix honey with artificial sugar solutions that mimic its appearance but lack essential enzymes and bioactive compounds.

- **Feeding Bees Sugar Syrup Instead of Nectar:** Instead of allowing bees to collect nectar from flowers, some producers feed them sugar solutions, leading to a weaker final product with reduced medicinal properties.
- **Ultra-filtration Techniques:** This method removes natural pollen traces, making it impossible to verify the honey's geographical origin and authenticity. [77]

1.3.3 Impact of Honey Adulteration

- Health Risks for Consumers
  - Fake honey lacks crucial antioxidants, natural enzymes, and antimicrobial properties, significantly diminishing its health benefits.
  - High-fructose corn syrup (HFCS) and artificial sweeteners in fake honey may contribute to obesity, diabetes, and metabolic disorders.
  - Some counterfeit honey may contain harmful contaminants such as lead, antibiotics, or heavy metals, which could pose long-term health hazards.
- Economic and Market Impact
  - **Unfair Competition:** The availability of low-cost, adulterated honey undercuts the market for genuine beekeepers and ethical honey producers.
  - **Loss of Consumer Trust:** Reports of widespread honey fraud erode public confidence in honey products, making people skeptical about authenticity.
  - **Challenges for Small-Scale Beekeepers:** Independent honey producers often struggle to compete with large manufacturers selling diluted or synthetic honey at lower prices. [108]
- Environmental and Ecological Impact
  - **Declining Pollination Activity:**Reduced demand for authentic honey discourages sustainable beekeeping, negatively affecting pollination and biodiversity.
  - **Industrial Pollution:**The mass production of artificial sugar syrups used in honey adulteration contributes to higher waste output and environmental pollution.

1.4 Challenges in Detecting Natural vs. Adulterated Honey

1.4.1 Variability in Natural Honey Composition

- **Diverse Botanical Origins:** Since honey originates from a wide variety of floral sources, it exhibits differences in taste, appearance, viscosity, and sugar ratios. This variation complicates the establishment of a universal benchmark for authentic honey.

- **Environmental and Seasonal Influence:** The geographical location, climate conditions, and seasonal factors impact the composition of honey, altering moisture levels, pollen characteristics, and biochemical markers.
- **Complex Chemical Profile:** Natural honey contains a mix of enzymes, organic acids, amino acids, and trace elements, which fluctuate naturally. This complexity makes it challenging to differentiate between natural variations and intentional adulteration.

1.4.2 Advanced Techniques Used in Honey Modification

- **Blending with Sugar-Based Substitutes:** Industrially modified honey often incorporates high-fructose corn syrup (HFCS), glucose solutions, or artificially inverted sugar syrups. These substances closely resemble the natural sugar profile of honey, making detection difficult through conventional tests.
- **Pollen Removal via Filtration:** Some manufacturers employ ultrafiltration to eliminate pollen grains and other natural indicators, preventing verification of honey's botanical and geographic origin.
- **Chemically Synthesized Honey:** Certain manufacturers produce entirely synthetic honey by replicating chemical compositions artificially, making traditional laboratory identification methods less effective.

1.4.3 Limitations of Traditional Detection Methods

- **Basic Physical Tests Lack Reliability:** Simple at-home methods such as the water drop test, thumb test, and flame test do not provide scientifically valid results in detecting alterations.
- **Chemical Analysis Challenges:**
  - Tests like HMF (Hydroxymethylfurfural) analysis may indicate heat exposure or the presence of additives, but some counterfeit honey products are specifically manufactured to pass this test.
  - Chromatography and spectroscopic methods (such as NMR, FTIR, or LC-MS) are highly precise, but they demand advanced technology and expert knowledge.
- **Microscopic Analysis Limitations:**
  - Some modified honey products still contain residual pollen, misleading analysts into incorrectly classifying them as natural.
  - Excessive filtration can remove pollen from genuine honey, leading to false positives for adulteration.

1.4.4 Inconsistent Honey Regulations and Standards

- **Varied Regulatory Policies:** The criteria for honey authenticity differ between countries. Some regions allow the inclusion of additives and heavy processing, while others enforce stricter purity standards.

- **Lack of Harmonized Testing Protocols:** Laboratories around the world use different analytical methods, leading to inconsistent authentication results for the same honey sample.

1.4.5 High Costs and Accessibility of Advanced Testing

- **Expensive Scientific Equipment:** Methods such as Nuclear Magnetic Resonance (NMR) and Isotope Ratio Mass Spectrometry (IRMS) provide high accuracy but require costly devices and trained specialists.
- **Accessibility Barriers for Small-Scale Producers:** Many independent beekeepers lack financial resources and access to advanced facilities, making it easier for low-quality honey to enter markets undetected.

1.4.6 Consumer Awareness and Market Challenges

- **Difficulty in Recognizing Altered Honey:** Most buyers struggle to distinguish pure honey from modified versions based on flavor, texture, or home-based tests.
- **Competitive Pricing and Fraudulent Practices:** Since adulterated honey is cheaper to manufacture, unethical sellers can offer it at lower prices, negatively affecting genuine beekeepers and quality producers[35].

1.5 Limitations of Traditional Honey Authentication

1.5.1 Physical Tests Are Inaccurate and Unreliable

Many consumers rely on home-based physical tests, but these methods lack scientific accuracy:

- **Water Dissolution Test:**
  - One of the most common methods that helps determine the purity. Of honey is the water dissolution test. The theory behind this test is that pure honey has a high viscosity and density, which prevents it from dissolving easily. as soon as it is immersed in cold water. Instead, it sinks to the bottom of the glass and needs. Stirring to combine. In contrast, adulterated honey is highly soluble mixing easily. and rapidly with very little stirring into water once it makes contact. [26]
  - **Limitation:** Some pure honey varieties dissolve faster depending on their moisture content and viscosity.
- **Thumb or Stickiness Test:**

  - Suggests that natural honey is thicker and less sticky than adulterated honey.
  - **Limitation:** The texture of honey varies based on floral source, temperature, and processing methods. [99]

- **Flame Test:**
  - Believes that pure honey burns when exposed to fire, while adulterated honey does not.
  - **Limitation:** Some pure honey varieties contain natural moisture that can prevent burning, and some adulterants do not affect flammability.[99]

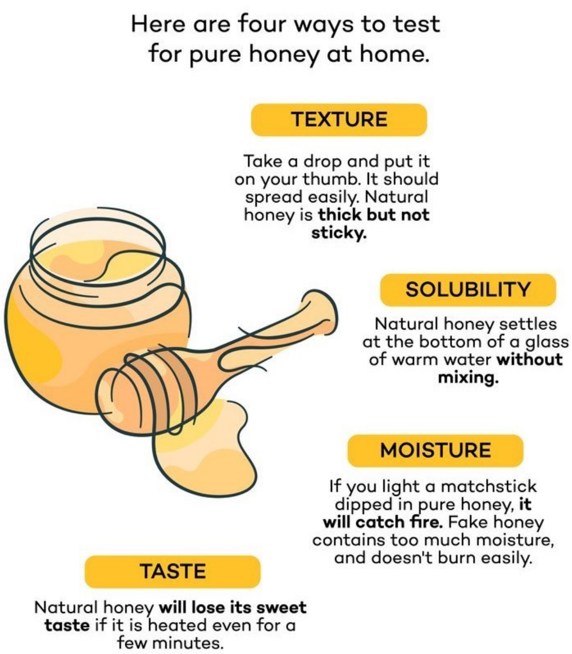


Figure 1.4: Home Honey Test Methods

1.5.2 Limitations of Chemical Analysis Methods

- High-Performance Liquid Chromatography (HPLC) and Gas Chromatography (GC)

  According to the cases were two sophisticated detection/analytical methods for detection and analysis, High-Performance Liquid Chromatography (HPLC) and Gas Chromatography (GC), glucose and organic Compounds Present in Honey

  - Then High-Performance Liquid Chromatography (HPLC):

    * Because it separates and identifies every sugar found in honey this method is widely used for detecting sugars from, e.g. glucose, fructose, and others. Also, it can find all the compounds present in honey. So you have to change all your ingredients before eating food when buying that thing again.

* A liquid sample is pushed by high pressure through a column full of a stationary phase. The compounds in the honey have a different pattern of contact with the stationary phase, hence they separate based on their chemical characteristics.

* The identification of sugars using HPLC is beneficial since it affords an exact determination of the sugar composition in honey. This can be useful for determining an adulteration with fruit syrups such as high fructose corn syrup (HFCS).

– Gas Chromatography (GC):

* GC is another crucial technique for honey volatile compound analysis. Whereas HPLC is for liquid sample detection, GC does the job on gases (vapors that like) and volatile compounds. It's therefore ideal for picking out substances such as hydrocarbons, alcohols, and sugar that are present in honeyscream.II.
*In GC, the sample is vaporized and then carried by an inert gas (such as helium or nitrogen). Through a column made of the stationary phase. Different compounds will move tinas the honey at different rates so that they can be separated and each is recognized by its retention time. [20]

– **Limitation:**
  * Some adulterants closely mimic natural sugars in honey and may not be detected.
  * Requires expensive equipment and trained personnel.

• Hydroxymethylfurfural (HMF) Test

* Hydroxymethylfurfural (HMF) is an organic compound derived from the dehydration of certain sugars. HMF is generally absent in fresh honey but is formed during heating, conditioning, and storage. Fera can measure HMF levels in a range of food products, including honey.

* HMF can be used as an indicator for excess heat treatment in many food products, such as honey, fruit juice, UHT milk, jams, alcoholic products, and biscuits. Fera conducts analysis of samples with high-performance liquid chromatography with UV detection (HPLC-UV), with detection down to 1 mg/kg.

* In honey HMF is a quality marker, Annex II of Council Directive 2001/110/EC lays down composition criteria for honey, including HMF content (determined after processing and blending). HMF is limited to 40 mg/kg in general (except baker's honey, which is exempt), and 80 mg/kg in honey declared to be from a tropical region.

* Fresh honey generally contains less than 15mg/kg HMF, but over 40 mg/kg is used to guarantee the honey has not undergone excessive heating during processing. Fera's HPLC-UV method can accurately quantify levels of HMF to prove your products satisfy these criteria. Fera's testing for HMF can be used accurately as a quality guide, aiding consumer trust in your product by verifying quality claims. Testing can also aid wholesalers and retailers, by ensuring the authenticity of the products

at any point in the supply chain and providing evidence products offered for sale are true-to-label. [32]

- **Limitation:**
  * HMF levels naturally vary based on floral source and storage conditions.
  * Some fake honey manufacturers add controlled HMF levels to pass this test.

Figure 1.5: Hydroxymethylfurfural (HMF) Honey Test

- Spectroscopic Techniques (UV-Vis, FTIR, NMR)
  - Development of non-destructive methods, which are based on the analysis of the sample without any alterations of product attributes, presents a problem of great practical significance. Spectroscopic methods occupy an important place among the comprehensive tests [Posudin, 2005]. These methods include the measurement of the difference between input and output light signals during the interaction of light with the sample (absorption, transmission, reflection, scattering, re-emission) and analysis of the dependence of this difference on the wavelength. Besides, spectroscopic methods are rather fast and precise.
  - Effects of honey type, age, temperature, water content, and degree of sugar adulteration on the spectral properties of honey can also be studied.
  - Certain spectroscopic methods that have been applied to honey control:
    * spectrophotometry
    * nuclear magnetic resonance
    * atomic spectroscopy [56]
  - **Limitation:**
    * Spectral profiles can overlap, making it difficult to differentiate between genuine and adulterated honey.
    * Requires specialized equipment and expertise, making it inaccessible for small beekeepers.

1.5.3 Microscopic and Pollen Analysis Challenges

- **Pollen Grains Verification:**

- Adulterated honey may still contain some pollen grains, misleading the analysis: Microscopic analysis is used to detect honey by identifying pollen grains, but the presence of pollen does not always indicate that the honey is pure. Adulterated honey may also contain traces of pollen or have pollen artificially added to it, or it may be mixed with sugar syrup and original honey.[85]
  - **Limitation:** Some fraudsters use pollen additives to mimic natural honey.

- **Microscope-Based Purity Tests:**
  - Cannot detect chemical adulteration such as sugar syrups or artificial honey compounds: Microscopic purity tests detect physical contaminants such as pollen or particles, but they are ineffective in identifying chemical adulterants like sugar syrups or artificial compounds in honey. These tests analyze the physical characteristics of honey but cannot differentiate between pure honey and adulterated honey which includes additives like high-fructose corn syrup or glucose syrup.[75]

- **Ultrafiltration Issues:**
  - Ultrafiltration is a treatment used to transform raw honey into a material suitable for food processing [55]. Some adulterated honey is ultrafiltered to remove natural pollen markers, making detection impossible through pollen analysis. While this process can enhance the clarity and shelf-life of honey, it has also been exploited as a method for honey adulteration. Specifically, some producers ultrafilter adulterated honey to deliberately remove natural pollen markers [19]

[49]

Figure 1.6: Ultrafiltration

1.5.4 Regulatory and Standardization Issues

- Different countries have varying honey quality standards, making it difficult to create a universal authentication method.[34]

- Some regulatory tests fail to detect sophisticated honey fraud due to evolving adulteration techniques.[34]
- Limited Access to Advanced Technology: Developing countries may lack access to sophisticated testing methods like NMR or mass spectrometry, making the detection of advanced adulteration harder.[101]
- Misleading Labeling: Loopholes in labeling laws can allow misrepresentation of origin, floral source, or purity.[101]
- International organizations and honey safety ensuring the safety of honey is a critical consideration given several factors. Adulteration jeopardizes the integrity of honey, potentially introducing foreign substances, sweeteners, or harmful chemicals. Agricultural pesticide residues may contaminate honey, posing health risks, while inadequate handling and processing can lead to microbial contamination and subsequent foodborne illnesses. [74]

1.5.5 Economic and Accessibility Challenges

- Advanced techniques like NMR and Mass Spectrometry are costly, limiting their use in small-scale production and testing centers.[54]
- Traditional laboratory-based methods are time-consuming, making real-time honey verification difficult for large-scale markets. [54]
- Factors influencing honey prices in 2025 Several factors will impact honey prices in 2025. Changes in climate can affect honey production levels drastically. Weather patterns influence the availability of flowering plants essential for bees. Economic challenges faced by beekeepers can lead to either lower or higher prices depending on the circumstances. Government policies surrounding agriculture often shape the economic landscape. These policies can affect subsidies and support for sustainable practices.[18]
- Supply chain issues have created additional economic challenges. Delays in shipping can lead to wasted products and lost sales. Many beekeepers rely on specific suppliers for equipment and services. When these supplies are hard to obtain, businesses suffer. Prolonged supply issues can shake the very foundation of honey-producing operations.[18]

1.6 Palynology

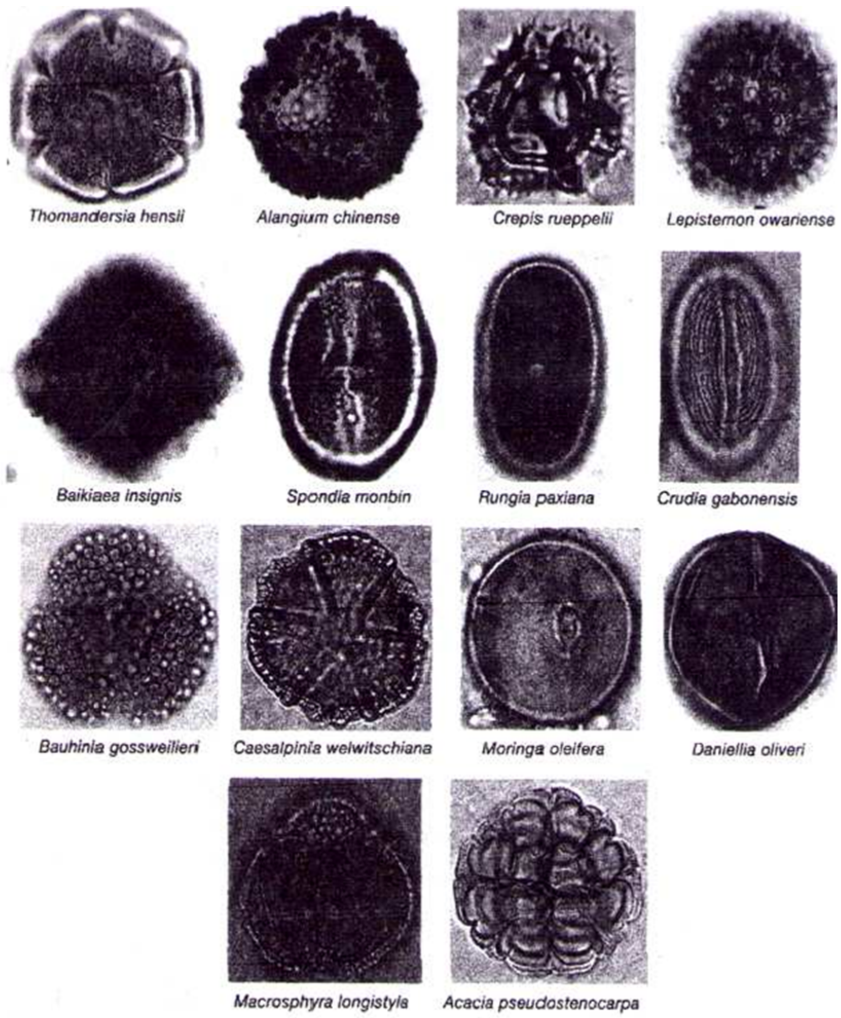
1.6.1 Pollen

Pollen consists of small spherical or oval-shaped particles ranging in size from 20 to 40 microns, found in the pollen sacs of the flower's anthers. It is used to fertilize the female part of the flower and represents the male gametes in the plant kingdom.

It is a collection of male reproductive cells that serve as the raw material bees use to produce honey.

Pollen is essential for bee nutrition, and its high nutritional value makes it one of the most complete foods in nature. It contains all the necessary nutrients, including protein and all the essential amino acids required for human nutrition.

Honey naturally contains pollen, which is an essential component in determining its authenticity. The presence and analysis of pollen in honey allow for the identification of the floral sources visited by bees, providing valuable information about the origin and botanical composition of the honey. Additionally, the ratio of fructose to glucose in honey influences its crystallization process.

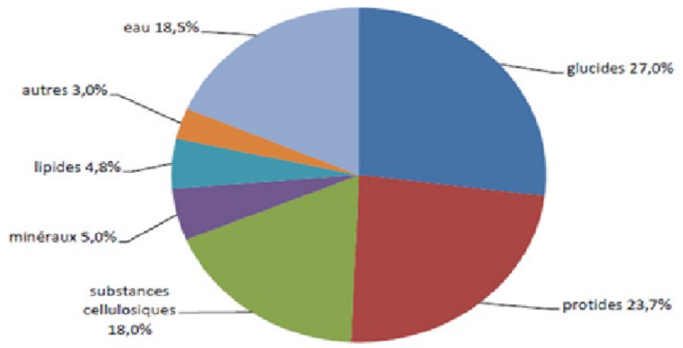


[49]

Figure 1.7: Some pollen grains from the flora of Africa

1.6.2 Composition of a pollen grain

The composition of a pollen grain is shown in this Figure.

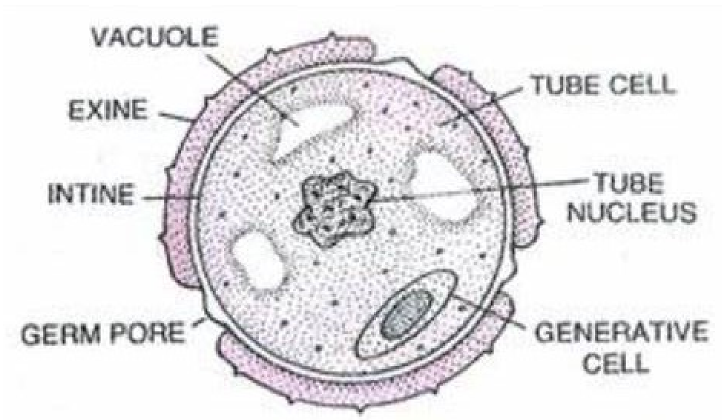


D'après Bruneau 2009 [50]

Figure 1.8: Composition of a pollen grain

1.6.3 Structure of a Pollen Grain

A pollen grain consists of two non-partitioned cells and therefore contains two haploid nuclei. One is large and vegetative, while the other is smaller and generative or reproductive. The entire structure is covered by a protective layer in the form of a double envelope.



[50]

Figure 1.9: Structure of a Pollen Grain

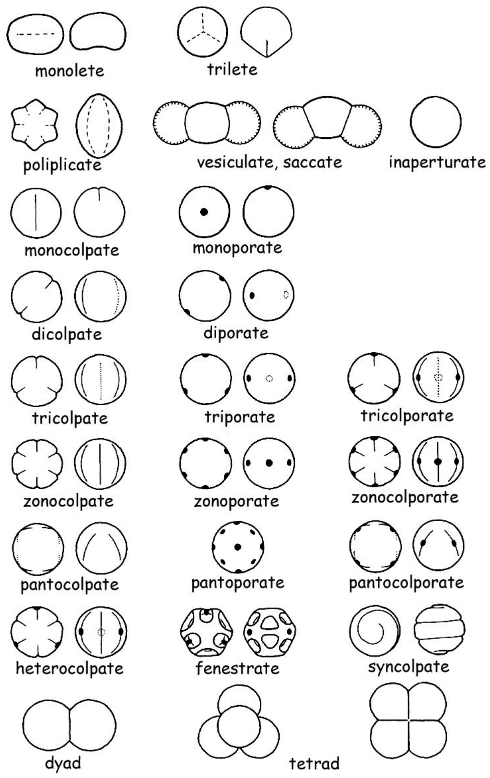
1.6.4 Main types of pollen

Les pollens sont caractérisés par les scientifiques selon divers critères :

- **Taille :**
  La taille du grain peut varier avec l'âge et les conditions de végétation de la plante. Cependant les rapports P/E restent globalement constants pour une même espèce et s'expriment en microns ou micromètres. La taille est comprise entre 5 et 250 microns. Le myosotis est parmi les plus petits grains de pollens et les cucurbitacées parmi les plus gros (200 et 250 microns), certains conifères à deux ballonnets (Renault-Miskovsky, 1990).

Bien que toutes les tailles intermédiaires existent entre ces extrêmes, un très grand nombre de grain sont une taille de l'ordre de 20 à 50 micron (Belaid , 1999).[21]

- **Les apertures :**
  Pore ou sillon ou association des deux ou encore absence d'apertures comme le mélèze par exemple. Le nombre d'apertures varie selon les espèces. Les pollens inaperturés sont des pollens ne présentant ni pore, ni sillon. Les pollens porés comportent de petites ouvertures circulaires (pores). Les pollens colpés comportent seulement une ouverture très allongée (sillon). Les pollens colporés comportent à la fois des pores et des sillons. (Homrani, 2020). (Figure 10) [21]



[50]

Figure 1.10: Somme types of pollen

1.6.5 Pollen Types and Corresponding Plants

Pollen Type	Examples
Inaperturate	*Juniperus sp*
Monocolpate	*Lilium sp*
Monocolporate	*Poaceae*
Dicolpate	*Hypecoum sp*
Tricolpate	*Quercus sp*
Stéphanolcolpate	*Rosmarinus sp*
Péricolpate	*Polygonum amphibium*
Diporate	*Broussonetia*
Triporate	*Corylus, Betula*
Stéphanoporate	*Ulmus, Alnus*
Périporate	*Saponaria*
Tricolporate	*Fagus*
Stéphanolcolporate	*Anchusa*
Péricolporate	*Rumex*
Hétérocolporate	*Lythrum salicaria*
Syncolpate	*Myrtus*
Fenestrate	*Taraxacum*

[50]

Table 1.1: Types of Pollen and Corresponding Plants

1.6.6 Pollen Analysis in Honey Authentication

- **Quality and Authenticity Assurance:**
  - **Ensures that honey is genuine and not adulterated by verifying its floral composition :**

    ***Fraud Detection:** Pollen analysis allows identification of the botanical source of honey. If pollen grains are absent or foreign types appear, this indicates fraud in classification.
    ***Purity Verification:** Comparing the pollen content with the declared source confirms whether the honey comes from specific flowers or has been mixed.
    ***Standards Compliance:** Regulatory standards (such as those of the European Union or Codex) require verification of floral origin as part of quality certification.

- **Identification of Flower Sources:**
  - **By examining pollen grains in honey, the plant species visited by bees can be identified :**

    ***Botanical Composition:** Different flowers produce pollen with distinct shapes and surface structures. Microscopic analysis helps identify the types of flowers visited by bees.
    ***Monofloral or Polyfloral Honey Classification:** If one type of pollen dominates (usually more than 45 for 100), the honey can be classified as "monofloral"

(like lavender honey). If it contains a mixture of different types, it is classified as "polyfloral."

*Nectar Source Tracing:** Knowing the plants visited by bees helps link the therapeutic properties of honey to its botanical source.

- **Verification of Honey Origin:**
  - **Pollen analysis helps determine the geographical and floral source of honey :**

    ***Identifying Geographical Origin:** The variation in plants across regions allows the determination of honey's origin (such as the Mediterranean, tropical areas, or mountainous regions) through pollen spectrum analysis.

    ***Combating Labeling Fraud:** Honey may be exported under a specific regional name while its actual source is different. Pollen analysis reveals such mislabeling.

1.7 Key Features for Honey Classification

Honey authentication is a crucial process to differentiate between natural and adulterated honey. Several advanced classification methods, including **microscopy and spectral analysis**, help in accurately identifying the purity of honey

1.7.1 Microscopy-Based Classification

Microscopy is a widely used technique to analyze the physical properties of honey. It enables classification based on various visual and structural characteristics, including:

- **Classification Based on Shape:**
  - Microscopy allows the identification of pollen grains present in honey, which can indicate its floral origin.
  - The shape of pollen and other particles can help determine whether the honey is natural or mixed with additives.

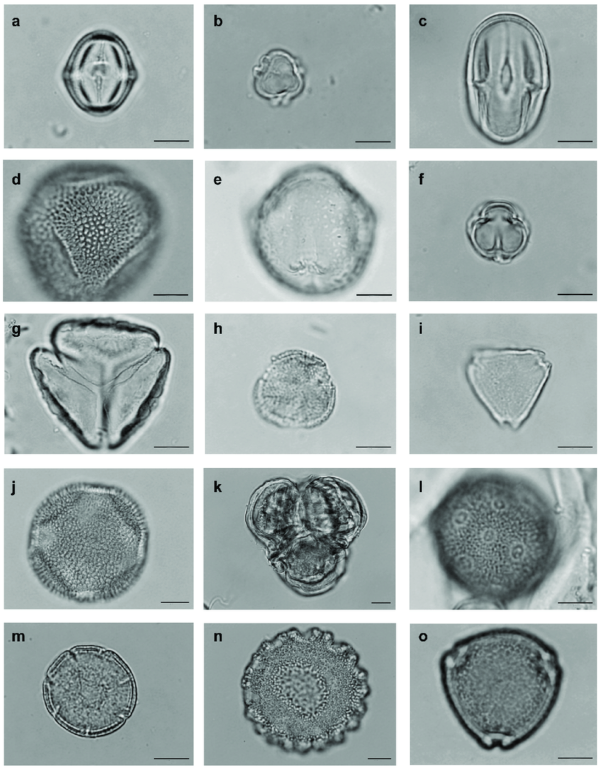


Figure 1.11: Pollen types from floral visiting bees' bodies of Solanum

- **Classification Based on Color:**
  - Pure honey exhibits a range of natural colors from light golden to dark amber, depending on its floral source.
  - Under a microscope, impurities, and artificial colorants can be detected, which helps in determining adulteration.

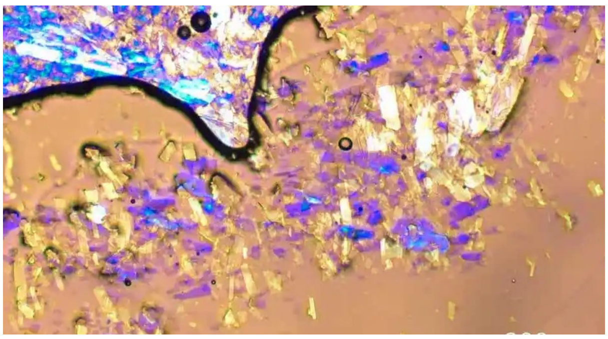


Figure 1.12: The Kaleidoscopic Colors of Honey Under a Microscope

1.7.2 Spectral Analysis

Spectral analysis is a non-destructive technique that provides insights into honey composition by analyzing how it interacts. The classification features in spectral analysis include:

- **Spectral Values:**
  - Spectral data is often recorded in CSV format for analysis.
  - These values represent the unique spectral fingerprint of honey, allowing comparison with standard pure honey profiles.
  - Different wavelengths of light are absorbed and reflected differently by natural and adulterated honey.
  - These spectral values help classify honey based on its molecular composition.

Elements	Light Honey			Dark Honey		
	Averages	Minimum	Maximum	Averages	Minimum	Maximum
K	205	100	588	1676	115	4733
Cl	52	23	75	113	48	201
S	58	36	108	100	56	126
Ca	49	23	68	51	5	266
Na	18	6	35	76	9	400
P	35	23	50	47	27	58
Mg	19	11	56	35	7	126
SiO_2	22	14	36	36	13	72
Si	8.9	7.2	11.7	14	5.4	28.3
Fe	2.4	1.2	4.8	9.4	0.7	35.5
Mn	0.3	0.17	0.44	4.09	0.52	9.53
Cu	0.29	0.14	0.70	0.56	0.35	1.04

Table 1.2: The mineral constituents of honey (mg/kg)

- **Chemical Analysis:**
  - Sugar Composition: Identifies glucose, fructose, and sucrose levels.
  - Acidity and pH Levels: Differentiates between natural and processed honey.
  - Presence of Additives: Detects artificial sweeteners, preservatives, and other adulterants.

N (s)	Floral Origin	CE $\times 10^4$ (S/cm)	pH	Color (Cm Pfund)	Color Type	Water Content (%)	MS (%)	HMF (mg/kg)	AL (meq/kg)	TMM (%)
1	Multi-flower	1.12	4.24	1.1	Light	14.8	83.46	2.84	11.0	0.08
2	Wild Carrot	6.41	4.50	9.9	Dark	18.8	85.40	5.68	11.5	0.48
3	Galactite	2.34	3.50	Off scale	Very Dark	14.03	85.44	117.7	25.0	0.17
4	Euphorbia	1.48	3.70	14	Dark	15.8	85.26	90.7	47.0	0.11
5	Multi-flower	1.46	4.16	6.2	Light	14.4	83.60	9.28	37.0	0.11
6	Multi-flower	2.61	4.25	6.2	Light	15.03	83.49	9.92	15.0	0.19
7	Multi-flower	0.83	4.48	1.1	Light	16.2	82.25	3.83	18.0	0.06
8	Multi-flower	5.06	4.42	8.3	Dark	17.4	81.10	12.87	20.0	0.38
9	Multi-flower	1.6	4.50	6.2	Light	14.6	84.60	4.19	16.5	0.12
10	Multi-flower	1.38	4.14	7.1	Dark	16.2	82.00	7.48	11.0	0.10
11	Multi-flower	3.49	4.38	8.3	Dark	15.1	85.10	21.25	26.0	0.26
12	Multi-flower	1.64	4.53	9.2	Dark	18.8	81.16	74.55	17.0	0.12
13	Multi-flower	5.86	4.49	Off scale	Very Dark	18.06	81.40	92.44	14.0	0.44
14	Jujube	4.04	4.04	9.2	Dark	14.2	83.70	2.89	2.89	0.30
15	Date	8.9	5.2	14	Dark	21.5	76.90	97.7	42.0	0.66
16	Multi-flower	2.0	3.80	9	Dark	18.03	79.70	81.1	29.0	0.15

Table 1.3: Physicochemical Analysis Results of some Honey Samples

1.8 The role of artificial intelligence in food authentication

1.8.1 Definition of Food Authentication

Ensuring food products are genuine, free from adulteration, and accurately labeled is essential for maintaining food safety and quality. The rising cases of food fraud, mislabeling, and contamination have heightened concerns about consumer health and food security. Traditional authentication methods, such as chromatography, chemical analysis, and spectroscopy, are reliable but often involve complex procedures, high costs, and significant time investment.

Artificial Intelligence (AI) is revolutionizing food authentication by providing automated, accurate, and efficient solutions for detecting food fraud. AI-powered technologies—including machine learning (ML), deep learning (DL), and computer vision—enhance the precision of food analysis and improve traceability throughout the supply chain.

1.8.2 The role of artificial intelligence

- Detection of Food Fraud
  - AI-based machine learning models analyze large datasets to identify adulteration, mislabeling, and counterfeit food products.
  - Advanced spectroscopic techniques, such as near-infrared (NIR) imaging, Raman spectroscopy, and hyperspectral analysis, integrated with AI, enhance the ability to verify food authenticity and composition.
- Quality Control & Safety Assurance
  - AI algorithms process image and chemical data to assess food quality and ensure compliance with safety standards.
  - Deep learning techniques detect irregularities in food, such as contamination, spoilage, or inconsistencies in texture and color.
- Traceability & Supply Chain Monitoring
  - AI integrated with blockchain technology provides complete transparency in the food supply chain, ensuring product authenticity from origin to consumer.
  - Smart sensors and IoT devices, powered by AI, continuously monitor environmental conditions like temperature and humidity, preventing food tampering.
- DNA-Based Authentication
  - AI aids in DNA sequencing analysis, helping detect ingredient substitutions and verifying the authenticity of meat, seafood, and plant-based products.
- Consumer Trust & Transparency
  - AI-driven mobile applications allow consumers to scan product barcodes and receive instant verification of food authenticity.
  - Chatbots and AI-powered recommendation systems provide consumers with verified information, helping them make informed purchasing decisions. [53]

1.9 Methods Needed for an AI-Based Approach

1.9.1 Importance of Using AI for Detecting Honey (Natural and Adulterated)

- Traditional honey authentication methods, such as chemical analysis and spectroscopy, are often time-consuming and require specialized equipment.
- AI-based methods offer a faster, more accurate, and cost-effective solution for detecting honey adulteration.

1.9.2 Machine Learning for Honey Authentication

- **Pattern Recognition:** Machine learning models can analyze spectral or chemical data to distinguish between pure and adulterated honey.
- **Classification Algorithms:** Supervised learning techniques (e.g., SVM, Random Forest) can classify honey samples based on their authenticity.
- **Feature Extraction:** AI can automatically extract meaningful features from honey composition data, improving detection accuracy.

1.9.3 Deep Learning for Honey Authentication

- **Image-Based Analysis:** Convolutional Neural Networks (CNNs) can detect impurities in honey by analyzing microscopic images.
- **Spectral Data Processing:** Deep learning models process spectroscopy data to identify adulterants.
- **Neural Networks for Predictive Analysis:** AI models can predict honey purity based on data and trends.

1.9.4 Advantages of AI Over Traditional Methods

- **Speed and Efficiency:** AI-driven authentication can analyze samples in real time.
- **Higher Accuracy:** Machine learning models improve detection rates compared to manual testing.
- **Automation and Scalability:** AI can process large datasets without human intervention, making large-scale testing feasible.

1.10 Types of Honey Analyzed Using Machine Learning

The rapid development of machine learning has opened new possibilities for the classification and analysis of honey types. By utilizing machine learning algorithms and image recognition techniques, it is now possible to classify honey based on various types, including natural, adulterated, and processed honey.

1.10.1 Machine Learning for Natural Honey Classification

Natural honey, often regarded as the purest form of honey, is free from additives and adulterants. Machine learning algorithms can analyze pictures of natural honey to identify unique characteristics that distinguish it from other types. The classification process involves training models using a large dataset of images of pure honey.

Key Features for Classification:

- **Color:** Natural honey varies in color depending on its floral source, ranging from light golden to dark amber.
- **Viscosity:** The flow behavior of natural honey, often thicker and slower than processed honey, can be captured in images.
- **Presence of Pollen:** Microscopic images can reveal the presence of pollen grains, which are characteristic of natural honey.

Machine learning models can be trained using these features to accurately classify honey as natural based on image data.

1.10.2 Machine Learning for Adulterated Honey Classification

Adulterated honey is often diluted or mixed with other substances like sugar, corn syrup, or artificial sweeteners. These adulterants alter the honey's appearance, texture, and overall composition, making it possible to distinguish adulterated honey from pure honey through machine learning techniques.

Key Features for Classification:

- **Color Deviations:** The presence of artificial colorants or additives can affect the color of honey, making it stand out from natural varieties.
- **Viscosity Changes:** Adulterated honey may be thinner and more liquid than natural honey, which can be observed in images.

- **Lack of Pollen Grains:** The absence of pollen grains or other natural particles can indicate adulteration.

Machine learning models can be trained to identify these changes in images and classify honey as adulterated based on these distinguishing features.

1.10.3 Machine Learning for Processed Honey Classification

Processed honey refers to honey that has been subjected to heat, filtration, or other industrial processes. These processes alter the physical and chemical properties of honey, which can be detected through machine learning-based image analysis.

Key Features for Classification:

- **Clarification and Filtration:** Processed honey often appears clearer and less cloudy than natural honey, as impurities are removed during filtration.
- **Color Alteration:** The heat used in processing can darken or change the color of honey.
- **Loss of Natural Elements:** Processed honey may have a lack of visible pollen or other natural elements.

Machine learning models trained on images of processed honey can recognize these features and classify honey accordingly.

1.11 Data sources

1.11.1 Microscopy-Based Analysis

- **Pollen Identification**: Observing honey under a *microscope* allows for the *detection of pollen grains*, which help in determining *botanical and geographic origin.*
- **Crystalline Structure Examination**: The *size, shape, and distribution of honey crystals* under microscopic imaging provide valuable insights into *natural versus modified compositions.*
- **Detection of Foreign Particles**: Advanced imaging can reveal *residues from adulterants*, such as *sugar granules, starch, or synthetic additives.*

1.11.2 Spectral Data Acquisition

- **Fourier Transform Infrared Spectroscopy (FTIR)**: This method captures the *unique infrared absorption patterns of honey*, making it possible to *identify chemical* *modifications.*

- **Nuclear Magnetic Resonance (NMR) Spectroscopy**: By analyzing *molecular structures*, NMR helps differentiate *pure honey from sugar-based substitutes.*
- **Hyperspectral and Near-Infrared (NIR) Imaging**: These techniques scan honey samples at *different wavelengths*, highlighting differences in *composition and purity.*

1.11.3 Chemical Composition Analysis

- **Hydroxymethylfurfural (HMF) Content Testing**: HMF is a *heat-induced compound* that indicates whether honey has been *overheated or adulterated with inverted sugars.*
- **Chromatographic Separation (GC-MS, LC-MS)**: Gas chromatography-mass spectrometry (GC-MS) and liquid chromatography-mass spectrometry (LC-MS) help detect *non-natural sugars, preservatives, and chemical contaminants.*
- **Isotope Ratio Mass Spectrometry (IRMS)**: This technique identifies the *carbon isotope ratios* in honey, distinguishing *natural nectar-based sugars from artificial corn syrup additives.*

1.11.4 Machine Learning and AI-Driven Data Utilization

- **Pattern Recognition Algorithms**: AI models process *microscopic images, spectral data, and chemical signatures*, learning to differentiate *authentic honey from altered versions.*
- **Deep Learning-Based Spectral Classification**: Neural networks trained on *large datasets of spectral readings* can classify *honey purity with high accuracy.*

1.12 AI techniques used

1.12.1 Image-Based Structural Analysis

Systems capable of interpreting visual data are employed to examine detailed patterns in provided samples.
Example:
- **Convolutional Neural Networks (CNNs):** These are deep learning models specifically designed for image-based tasks. CNNs apply convolutional layers to analyze visual patterns, such as edges, textures, and shapes, which can be used for classification tasks like identifying objects in images or detecting tumors in medical imaging.

1.12.2 Mathematical Discrimination Models

Methods grounded in statistical computation assist in distinguishing various dataset categories based on key differentiating characteristics.
Example:
- **Support Vector Machines (SVMs):** A supervised machine learning algorithm that finds the hyperplane that best separates different classes in the dataset. SVM is widely used for binary classification tasks like spam email detection or handwriting recognition.

1.12.3 Layered Decision Frameworks

Techniques leveraging interconnected pathways enhance precision in classification.
Example:
- **Random Forests:** An ensemble learning method that constructs multiple decision trees and combines their outputs. Each tree in the forest is trained on a random subset of the data, and the final classification is determined by the majority vote or average of all the trees.

1.12.4 Comprehensive Data Fusion

A combination of multiple observational and analytical sources ensures improved consistency and reliability in assessment outcomes.
Example:
- **Multimodal Learning:** A method that combines information from different modalities such as text, image, and audio to improve performance in tasks like sentiment analysis or multimodal translation. For instance, in medical diagnostics, multimodal fusion might combine imaging data (X-rays) with patient health records to enhance diagnosis accuracy.

1.13 Conclusion

Honey is a vital product, but its authenticity is increasingly compromised by adulteration, posing risks to consumers and markets. Traditional methods of detection are limited, making it difficult to ensure purity. The integration of advanced computational techniques, such as machine learning, offers a more accurate and efficient solution. Analyzing diverse data types can significantly improve the identification of pure and adulterated honey, paving the way for more reliable quality control and enhanced consumer protection in the industry.

Chapter 2 :

Artificial Intelligence and Machine Learning

2.1 Introduction

Computer science has witnessed significant development in all aspects, from software engineering to the hardware industry, leading to major improvements and new programming approaches that solve real-world problems.

One of the most important branches of computer science is Artificial Intelligence (AI), which aims to create intelligent machines capable of solving problems that traditional programming approaches cannot address.

2.2 Artificial Intelligence

Artificial intelligence (AI) is the science that empowers machines to think, using technology and algorithms to help computers solve problems. Most AI products today, from chatbots to self-driving cars and manufacturing robots, rely heavily on deep learning (neural networks) and machine learning. These techniques train computers to perform specific tasks by processing large amounts of data and identifying patterns within it [104].

2.3 Machine Learning

Machine learning is a branch of computer science concerned with building algorithms that rely on collections of examples to be useful. These examples may come from nature, be handcrafted by humans, or be generated by other algorithms. Machine learning can also be defined as the process of solving real-world problems by collecting a dataset and algorithmically building statistical models based on that dataset. These statistical models are then used to solve real problems.

2.3.1 Machine Learning Techniques

Machine learning techniques are fundamentally algorithms that work on data to retrieve insights. These insights may include discovering, predicting, or forecasting patterns and trends. The goal is to develop a model utilizing a combination of data and algorithms that can be applied to new, unseen data to derive actionable insights.

Every technique relies on the type of data it works on and the specific problem it aims to solve. It is important to note that there is no universal machine-learning algorithm that can solve all the problems.

The main inputs for a machine learning algorithm are features extracted from data through a process known as feature extraction, often accompanied by feature engineering, which involves developing new features from existing ones. A feature represents an aspect of the dataset, such as location, age, number of shared posts, etc., when dealing with social media user data [87].

Machine learning techniques can be categorized into four main types:

- Supervised learning
- Unsupervised learning
- Reinforcement learning
- Semi-supervised learning

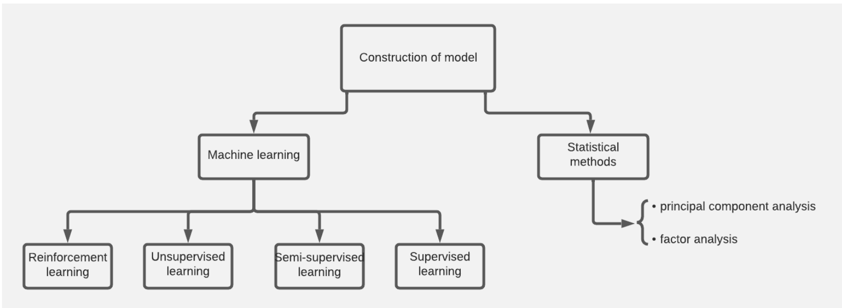


Figure 2.1: Different techniques from AI.

2.3.1.1 Supervised Learning

A type of machine learning is considered supervised when the dataset used in the process is already classified or contains examples with relevant correct classes. The algorithm uses this data to learn and classify future outcomes. For instance, in classifying handwritten digits, a supervised learning algorithm requires numerous images with their corresponding labels. The algorithm learns patterns between images and their labels to predict labels for entirely new, unseen images [59].

To demonstrate how supervised learning works, consider forecasting an individual's annual salary based on the number of accomplished educational years. Formally, we aim to create a model that approximates the relationship between the accomplished educational years X and the corresponding annual salary Y :

$$Y = f(X) + \epsilon \tag{2.1}$$

where:

- X (input) = accomplished educational years
- Y (output) = annual salary
- f = function describing the relationship between X and Y
- ϵ = random error term (positive or negative) with mean zero

In supervised learning, the machine attempts to comprehend the relationship between salary and education by processing classified training data through a learning algorithm.

The resulting model can then predict salaries for individuals whose salary Y is unknown, based on their educational years X [70]. The workflow of supervised machine learning algorithms is illustrated in Figure 2.2.

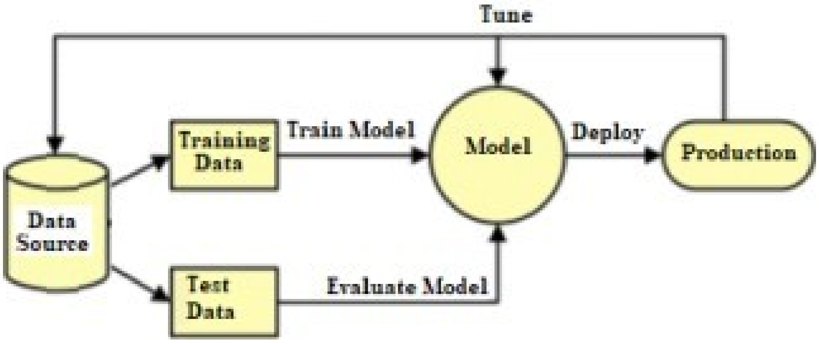


Figure 2.2: Supervised learning workflow [67].

Here are some algorithms that can be used in supervised learning:

- Decision Tree
- Linear Regression
- Logistic Regression
- K-NN (K-Nearest Neighbors)
- SVM (Support Vector Machine)

2.3.1.2 Unsupervised Learning

Unsupervised learning is more complex because the system must identify similarities in the input data and organize them accordingly. This method has significant advantages, particularly in the classification phase [59].

When using unsupervised learning, we don't care about the target output because the goal of the algorithm is to find relationships in the data and group data points based only on the input data. Supervised learning involves labeling data for prediction purposes, but unsupervised learning does not.

Here are some of the most important unsupervised learning algorithms (Clustering):

- K-Means
- Hierarchical Cluster Analysis (HCA)
- Visualization and Dimensionality Reduction
- Principal Component Analysis (PCA)
- T-Distributed Stochastic Neighbor Embedding (t-SNE)
- Association Rule Learning

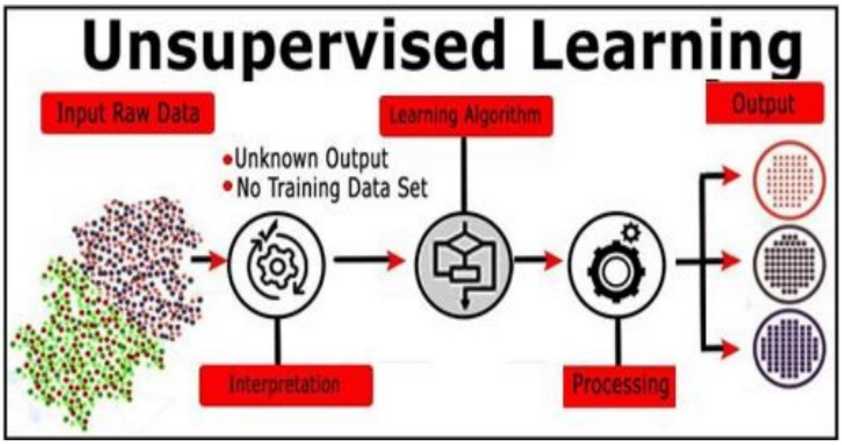


Figure 2.3: Unsupervised learning workflow [67].

2.3.1.3 Reinforcement Learning

Reinforcement learning is a class of computational algorithms that specifies how an artificial agent (e.g., a real or simulated robot) can learn to select actions to maximize the total expected reward. This computed difference, termed reward-prediction error, has been shown to correlate very well with the phasic activity of dopamine-releasing neurons projecting from the substantia nigra in non-human primates.

Reinforcement learning is a special case of supervised learning, where the exact desired output is unknown. The teacher supplies only feedback about the success or failure of an answer. This is cognitively more plausible than supervised learning since a fully specified correct answer might not always be available to the learner or even the teacher. It is based only on the information as to whether or not the actual output is close to the estimate. Reinforcement learning is a learning procedure that rewards the neural network for its good output result and punishes it for the bad output result [30].

2.3.1.4 Semi-Supervised Learning

There are other types of classification based on other types of learning methods such as “semi-supervised learning.” In effect, semi-supervised learning is a good compromise between the two types of “supervised” and “unsupervised” learning because it makes it possible to process a large amount of data without needing to label them all, and it takes advantage of the benefits of both types mentioned.

On the other hand, a priori labeling of all the data requires the intervention of a human expert. This is a difficult or even tedious operation when the number of data points is large. In concrete applications, it is often impossible for the expert to assign all the training data to the classes present [96].

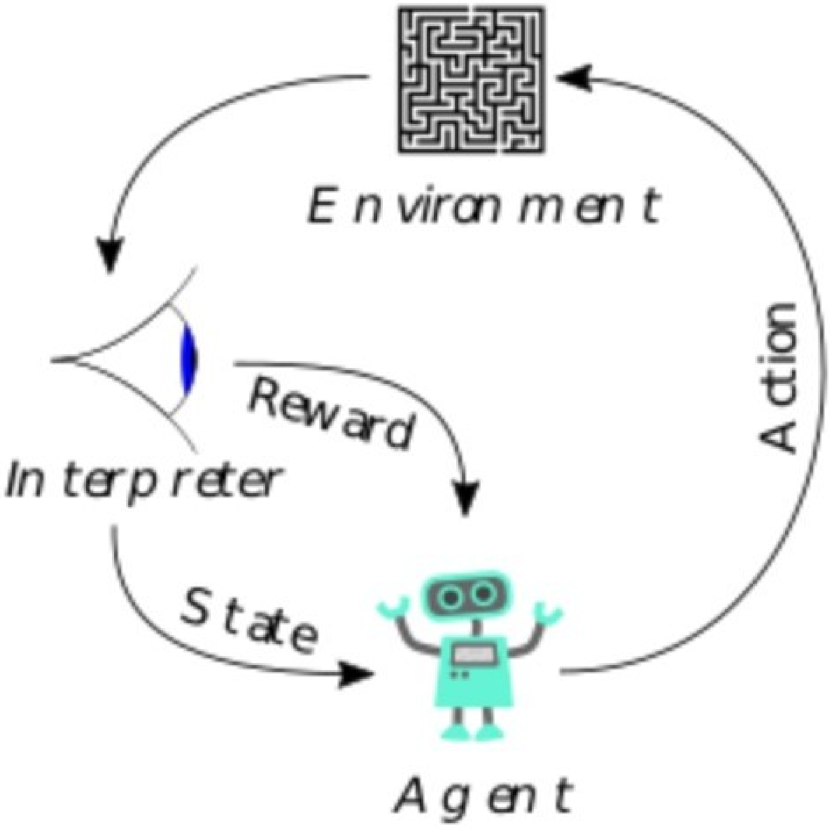


Figure 2.4: Renforcement learning workflow [67].

2.3.2 Regression

Regression is used to estimate the value of a label from a collection of related features. Labels can be any ground-truth value, rather than from a limited set of values as in classification tasks. Regression algorithms relate the dependencies of a label to its associated features to indicate how the label changes when the feature value changes. A regression algorithm takes a set of samples with known label values and provides a model that predicts the label values for a new given set of features [90]. Here are some examples where regression can be used:

- Estimate home prices based on a range of characteristics such as number of rooms, community composition, home size, and usable area.
- Predict Bitcoin price based on historical data and current market trends.
- Forecast product sales based on advertising budget [90].

2.3.3 Classification

For classification tasks, the algorithm takes a set of labeled samples and outputs a classifier that we can use to predict the class of new unlabeled instances, where the number of classes is limited. Classification can be used for:

- Classify face expressions as 'happy', 'sad', or 'angry'.
- Classify Bitcoin as "Save" or "Danger" to buy using current market trends.

2.4 Deep learning

2.4.1 Definition

Deep learning is a branch of machine learning that deals with algorithms inspired by the networks and operations of the brain, named artificial neural networks. By way of explanation, it mirrors how our brains work. Deep learning algorithms are closer to the structure of a nervous system, where each neuron is connected and transmits information.

Deep learning models work in layers, and a standard model has at least three layers. Every layer takes information from the previous layer and passes it to the next layer [60].

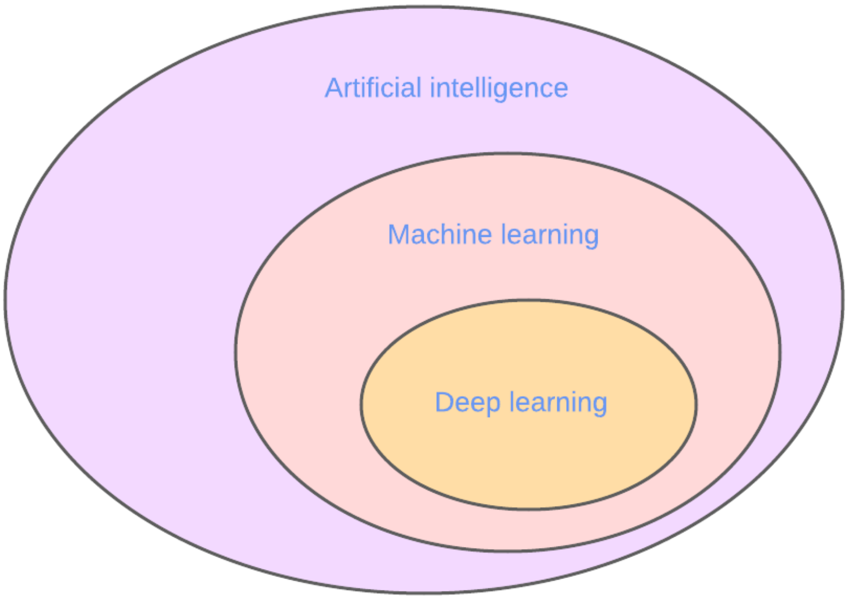


Figure 2.5: The sub-branches of artificial intelligence .

2.4.2 Difference Between Machine Learning and Deep Learning

Deep learning models usually function well on large datasets, while older machine learning models stop enhancing after reaching a saturation point.

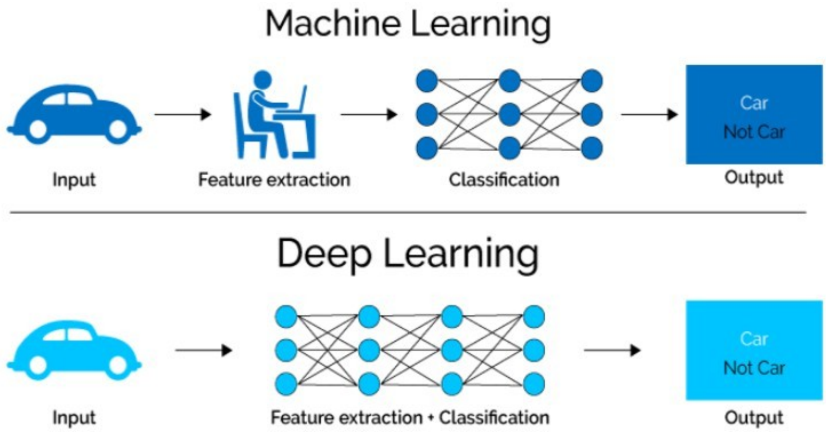


Figure 2.6: Feature extraction area in machine deep learning [8].

One of the differences between machine learning and deep learning models is in the field of feature extraction. Feature extraction is done by humans in machine learning, while deep learning models compute it themselves.

2.4.3 Application areas of deep learning

These techniques are developed in the field of computer science and applied to NTIC (visual recognition - for example a road sign by a robot or an autonomous car and voice recognition), robotics, bioinformatics, pattern recognition or comparison of forms, security, health, etc, computer-assisted education, and more generally artificial intelligence. Deep learning can for example allow a computer to better recognize highly deformable objects and/or analyze for example the emotions revealed by a photographed or filmed face, or analyze the movements and position of the fingers of a hand, which can be useful to translate sign language, to improve the automatic positioning of a camera, etc. They are used for some forms of medical diagnostic assistance (e.g., automatic recognition of cancer in medical imaging), or prediction (e.g.: prediction of the properties of a soil filmed by a robot) [62]

2.4.4 Deep learning architecture

Deep learning is a neural network with a large number of parameters and layers, the basic example is the multilayer perceptron.

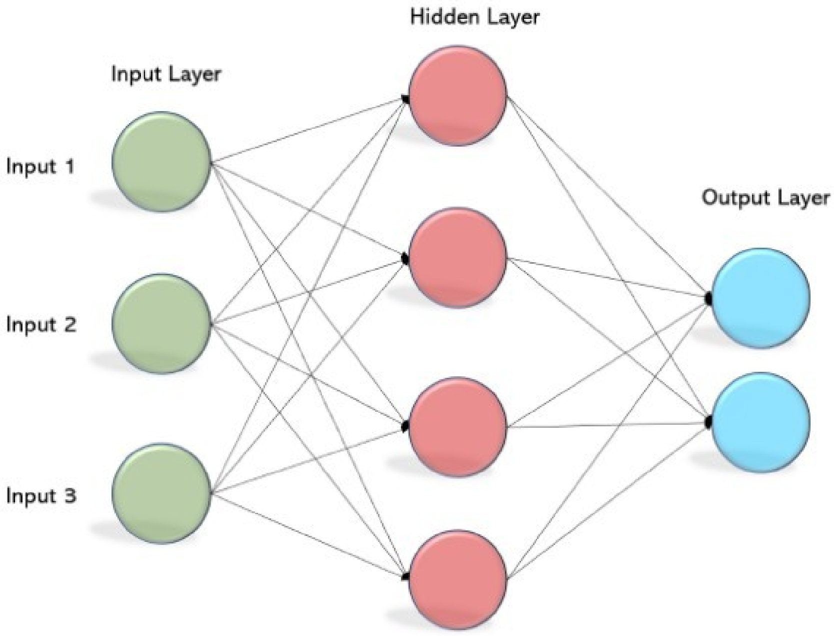


Figure 2.7: Multi-layer perceptron [73].

There are a large number of deep architecture variables. Most of them are derived from some original architectures. We will start with the convolutional neural networks (CNNs).

2.5 Artificial neural networks

To understand artificial neural networks first, we need to know some basic notions of biological neural networks and the adaptations made in the artificial neural network model.

2.5.1 Biological neuron

A biological neuron (see figure 2.8) is a cell composed of a nucleus. The cell body is divided to form the dendrites, it is by the dendrites that the information is transported from the outside to the soma (Cell body), the processed information continues its way through an axon to be transmitted to the other neurons. The transmission between two neurons is not direct, but rather there is a small space 109 between the axon of the afferent neuron and the dendrites of the other neuron. This space, defined as intercellular, is called a synapse [4] [5].

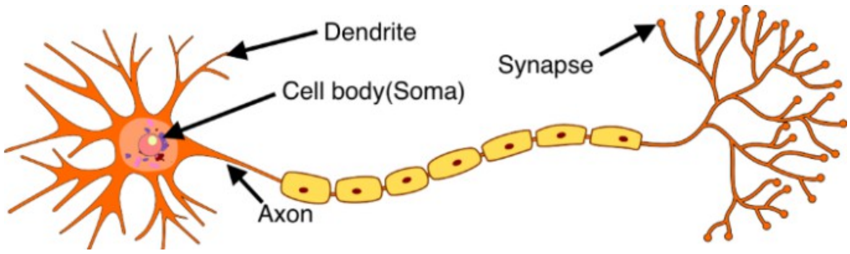


Figure 2.8: Biological neuron.

2.5.2 Artificial neuron

An artificial neuron (see 2.9) is a simplification of the biological neuron, each artificial neuron is an elementary processor. It takes as input several variables X = x1,x2,x3,...xn called the input layer, each input of an artificial neuron is associated with a weight w representing the value of the connection. An activation function f transforms the weighted sum of the input variables and their weights Pn i=1 (w i × xi) to a value which will then be transmitted to the output layer to be compared with a threshold value, and then provide an output response [105] [5].

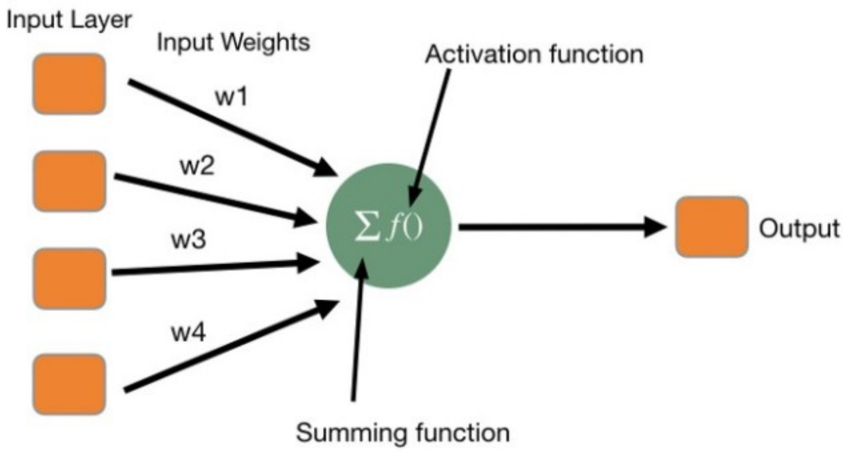


Figure 2.9: Artificial neuron.

2.5.3 Artificial neural networks

The artificial neural network (multi-layer) consists of several layers of neurons: an input layer, a hidden layer, and an output layer. This neural network is an improvement of the previous one to address more complex problems.

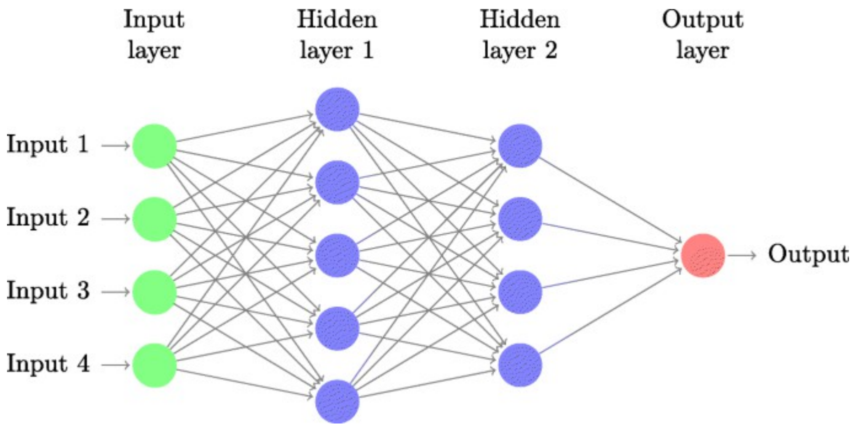


Figure 2.10: An artificial neural network.

The artificial neural network has 3 types of layers

* Types of layers

Input layer: The first layer in the neural network, is the layer that receives data, composed of input neurons.

Hidden layer: This layer or these layers are located between the input and the output layers. In these layers, the problems will be solved. The choice of hidden layers depends on the complexity of the problem.

Output layer: This is the last layer of an artificial neural network. This last layer produces the output of the program. If we have a classification problem, then the output layer neuron size will be equal to the number of classes.

2.5.4 Activation function

We mentioned the activation function in the last subsections. In this subsection, we will detail more about the activation function.

An artificial neuron calculates the sum of the inputs and their weights as follows:

$$\sum_{i=1}^{n}(w_i \times x_i) + b_i, \tag{2.2}$$

where this sum can take any value between $-\infty$ and ∞. To determine the threshold value for activation, an activation function is used.

For the bias, we add a bias value to the weighted sum to obtain the final value for prediction by our neuron [5].

The activation function is used to introduce non-linearity in the functioning of an artificial neuron, unlike biological neurons, which have a binary activation.

The activation function of an artificial neuron has continuous values, allowing an infinite number of possible values within an interval of $[-1, 1]$ or $[0, 1]$.

There are several forms of activation functions, each used in a specific context. We will mention the most commonly used ones.

- ReLU Function

  Defined by the following equation:

$$A(x) = \max(0, x) \tag{2.3}$$

  The activation function ReLU (Rectified Linear Unit) is a linear function that is considered the most popular and most used activation function. Its operation is based on the fact that it replaces any negative input value with 0, and without modifying positive value [2].

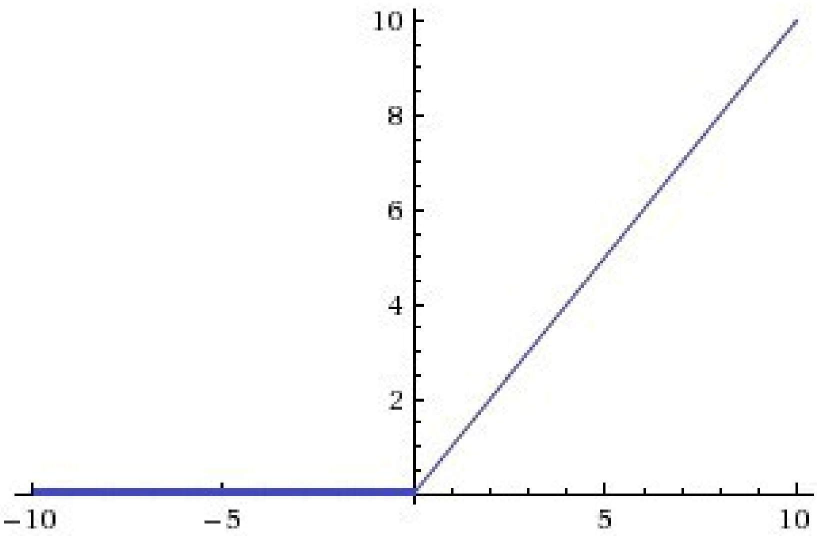


Figure 2.11: ReLU Function.

- Sigmoid function

  Defined by the following equation:

$$f(x) = \frac{1}{1 + e^{-x}} \tag{2.4}$$

  Two values are possible for this activation function: 0 or 1. Its curve takes the form of an S-shape. This function is used in the output layer when we have a binary classification problem [2].

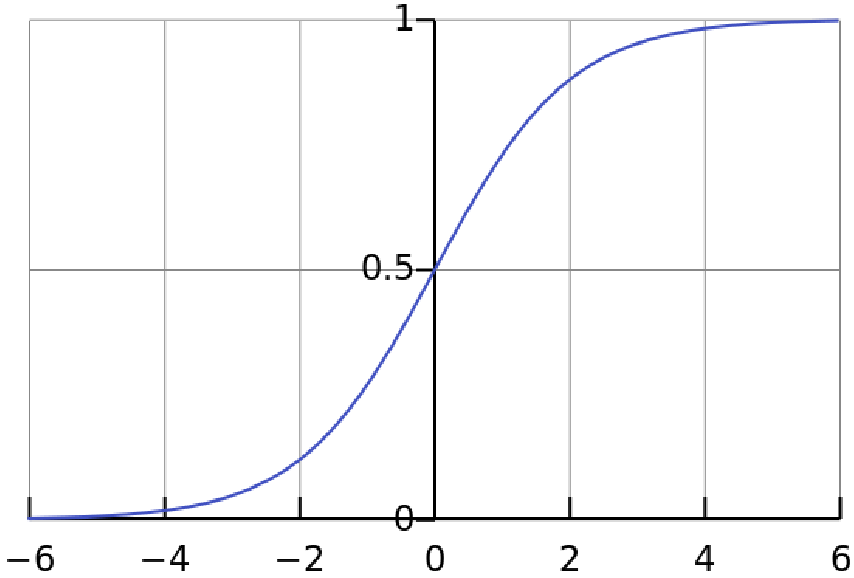


Figure 2.12: Sigmoid function.

- Tanh function Defined by the following equation:

$$f(x) = \frac{2}{1 + e^{-2x}} - 1 \tag{2.5}$$

This function transforms any real input into a value between $[-1, 1]$ [2].

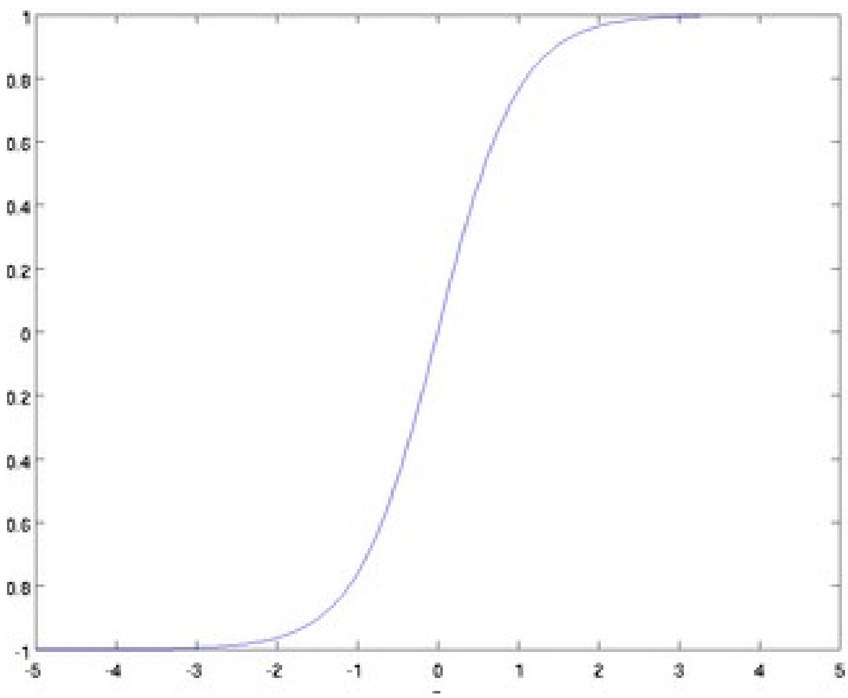


Figure 2.13: Tanh Function.

Tanh is a variant of the sigmoid function. The relation between the Tanh function and the sigmoid function is given by:

$$\tanh(x) = 2 \cdot \text{sigmoid}(2x) - 1 \tag{2.6}$$

- Softmax Function

  A normalized exponential function, the softmax function is used to represent a categorical probability distribution over a vector $\mathbf{z} = (z_1, z_2, \ldots, z_K)$ of K real numbers by transforming them into a vector $\sigma(\mathbf{z})$ with K probabilities, where the sum of these probabilities equals 1.

  The softmax function is defined as follows:

$$\text{softmax}(z)_i = \sigma(z)_i = \frac{e^{z_i}}{\sum_{k=1}^{K} e^{z_k}}, \quad \text{where } \mathbf{z} = (z_1, z_2, \ldots, z_K) \in \mathbb{R}^K. \tag{2.7}$$

  This function is used in the output layer for the case of K-class classification with $K \geq 2$ in order to compute the probability of an input z_i belonging to a particular class (the class with the highest probability) [6].

2.6 Convolutional neural networks

The convolution neural network was introduced by Yann Lecun in November 1998. CNNs are a specific type of deep neural network model that is mainly used to recognize images because they are good at extracting features from Images that are invariant to rotation and change in scale (to an extent only through, you still require a very variant and large dataset to tackle translational variance), As CNNs are the heart of all the other architectures we are going to explain it deeply [65].

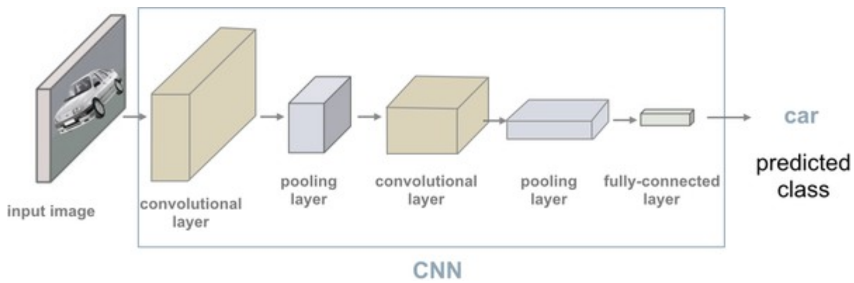


Figure 2.14: CNN architecture [22] .

2.6.1 Convolutional layer:

From its name, we deduce the first-word" convolution", which means that the network uses a mathematical operation called convolution. This is a special type of linear operation. The main goal of convolution is to extract features such as edges, colors, and vertices from the input. As we dig deeper into the network, the network also begins to recognize more complex features, such as shapes, numbers, and parts of faces. But how does this layer extract features such as edges, colors, etc? To understand how convolutional layers extract these features, let's first understand how our machine sees images.

Every image or picture in our machines is represented by pixels and every pixel has a value of the color scale between 0 and 255, 0 means this pixel is black and 255 means this pixel is white, and so on.

In grayscale images, each pixel represents the intensity of only one shade how bright or dark the pixel is. In other words, it has only one channel. On the other hand, for colored images, we have three color channels: R, G, and B (red, green, and blue). Standard digital cameras have three (RGB) channels.

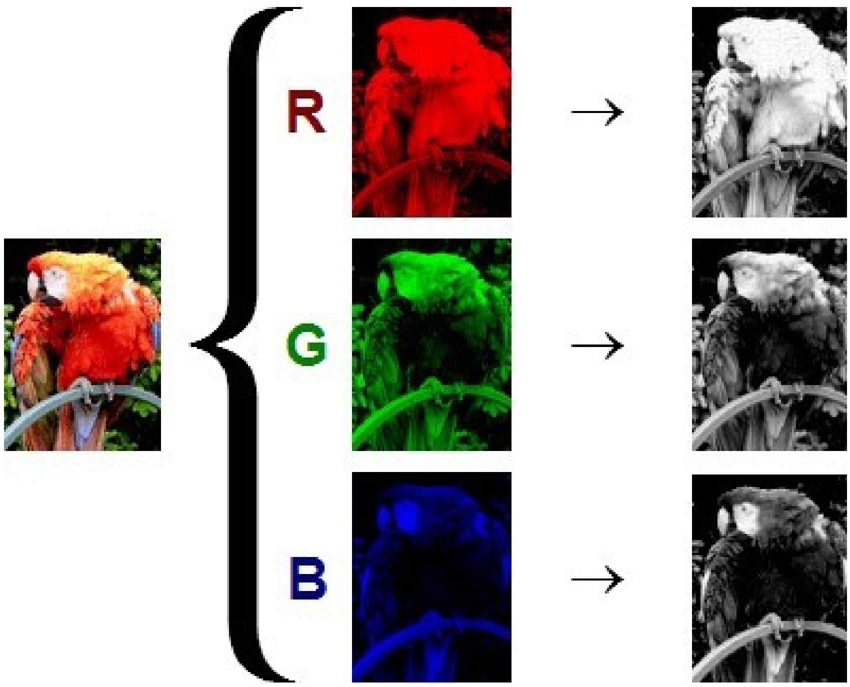


Figure 2.15: How machines understand colored images.

The red, green, and blue channels are stacked on top of each other, making the matrix a three-dimensional shape.

For example, the shape of a matrix representing a 500px by 500px color image will be $(500, 500, 3)$. Each pixel in this color image has three numbers (ranging from 0 to 255) associated with it. These numbers represent the intensity of red, green, and blue colors in that particular pixel.

After understanding how picture representation works, let's dive into how the convolutional layer extracts features from images. A convolution is a transformation applied pixel by pixel by using a set of weights, also known as a filter or kernel.

Let S be a set of source pixels and W be a set of weights. A pixel y is transformed as follows:

$$y = \sum_{i=0}^{n} S_i \cdot W_i \tag{2.8}$$

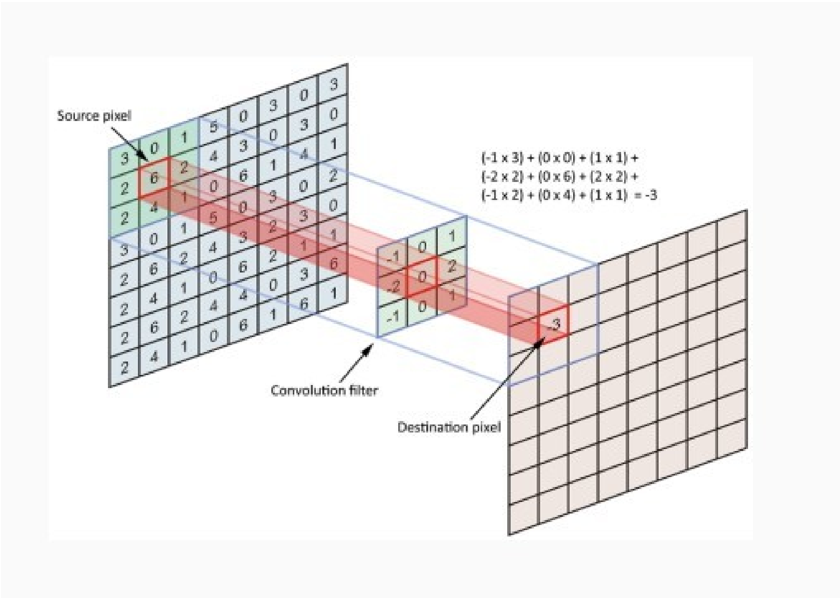


Figure 2.16: Convolution operation [88].

We are talking about things like straight lines, plain colors, and curves when we say features. Consider the most basic traits that all images share. These features are recovered by shifting the kernel over the image and applying a matrix multiplication between the kernel and the area of the image where the kernel is hovering at any given time.

The filter moves to the right by a determined step value or step size (Stride) until the entire width is resolved. It continues to the beginning of the frame (left) with the same step value and repeats the process until the entire frame is traversed.

2.6.2 Pooling layer:

Pooling layers, like convolutional layers, are responsible for minimizing and reducing the size of convolutional features. The computational power required to process the data is reduced through dimensionality reduction. It also facilitates the extraction of rotation and position invariant main features, which allows the model's training process to run smoothly. Pooling can be divided into two types: maximum pooling and medium or average pooling. Maximum pooling returns the maximum feature or value of the piece of the picture surrounded by the kernel. On the other hand, average pooling returns the average of all values from the kernel piece of the picture. Max pooling can also act as a noise canceler. It removes all sound activation and performs well.

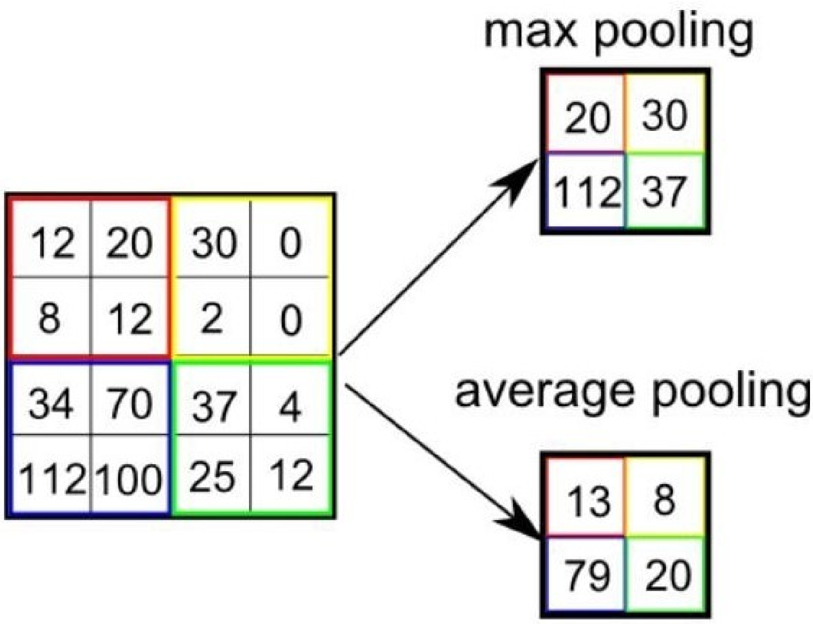


Figure 2.17: Types of polling operation [94].

2.6.3 Fully connected layer (dense layer):

We haven't done anything yet to categorize the various images. We have just highlighted specific characteristics in an image while substantially reducing its size. These high-level features can now be detected by connecting fully connected layers (dense layers) to the network's end.

This layer takes an input (the result of the convolution, ReLU, or pooling layer) and returns an N-dimensional vector, where N represents the number of classes from which the program must choose.

For example, if we are dealing with a digit recognition problem, N would be ten because there are ten digits. The likelihood of each class is represented by a number in this N-dimensional vector.

For instance, if the output is:

$$[0, 0.1, 0.1, 0.75, 0, 0, 0, 0, 0, 0.05]$$

It means that there is a 10% chance that the number is a 1, a 10% chance that the image is a 2, a 75% chance that the image is a 3, and a 5% chance that the image is a 9. (As a side note, there are numerous ways to describe the output; this example demonstrates the softmax method).

This fully connected layer analyzes the output of the preceding layer (which should represent the activation maps of high-level features) and determines which features are most associated with a specific class.

For example, if the computer predicts that a certain image is a mouse, the activation maps will contain high weights or values that reflect high-level traits such as a foot or four legs, and so on.

Similarly, if the algorithm predicts that a given image is a bird, the activation maps that reflect high-level traits such as wings or a beak will have high values. A fully connected layer considers which high-level traits are most strongly associated with a specific class and assigns weights to them, resulting in the correct probabilities for the various classes when the products of the weights and the preceding layer are computed [27].

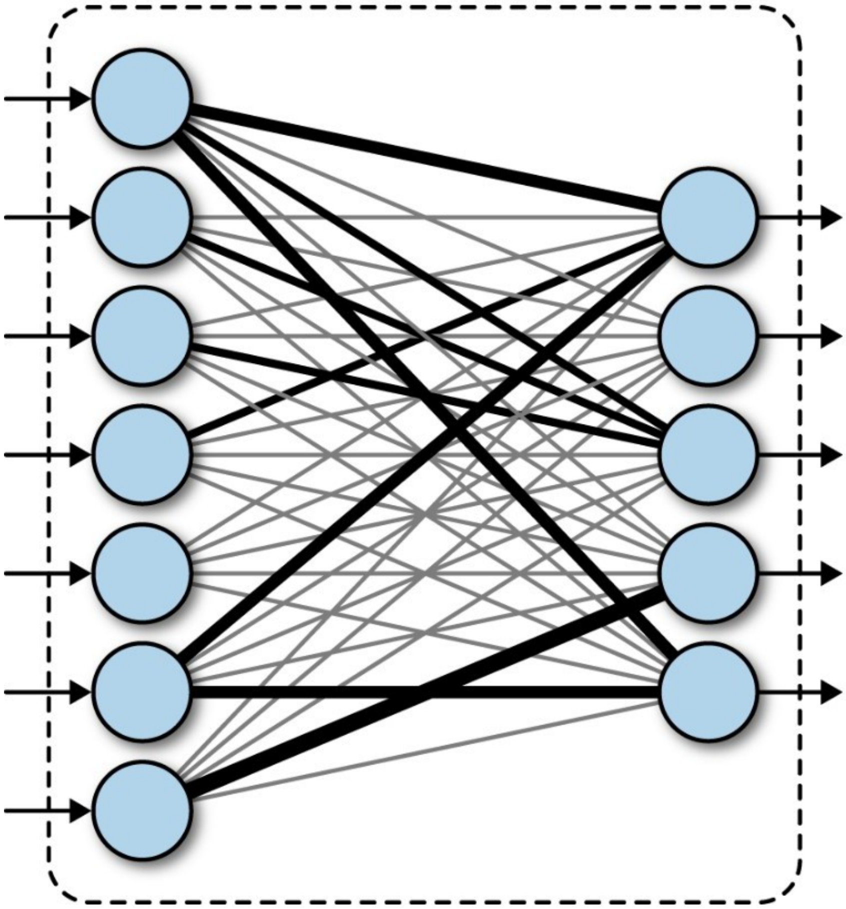


Figure 2.18: Fully connected layer [1].

2.7 Popular CNN Architectures

There is an infinite number of architecture, we will describe some of the most popular architectures like VGGNet, LeNet, and EfficientNet family.

2.7.1 VGGNet

Background • VGGNet This architecture, which was one of the first to appear, was introduced by Simonyan and Zisserman in 2014. The VGGNet architecture in the figure reffig:Basic architecture of VGGNet is composed of convolutional layers, ReLU as an activation function, Maxpooling layers inserted between convolutional layers in order to reduce the size of feature parameters, the classification the block is composed of dense layers that use ReLU also, and the final layer uses softmax for classification [3].

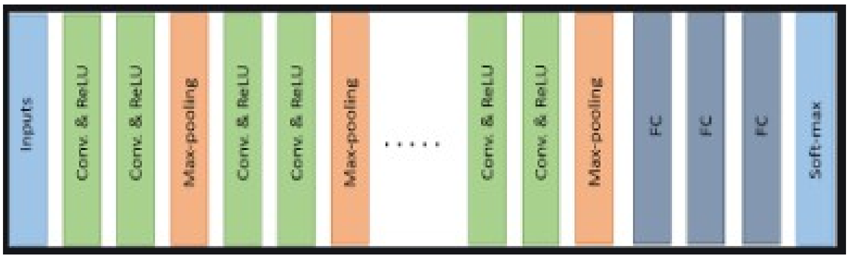


Figure 2.19: Basic architecture of VGGNet .

2.7.2 EfficientNet

This architecture of this family of architecture (B0,B1,..,B7), were realised by google in 2019. The EfficientNet-B0 architecture in Figure 2.20 wasn't developed by engineers but by the neural network itself. They developed this model using a multi-objective neural architecture search that optimizes both accuracy and floating-point operations. Taking B0 as a baseline model, the authors developed a full family of EfficientNets from B1 to B7 which achieved state-of-the-art accuracy on ImageNet while being very efficient to its competitors [98]

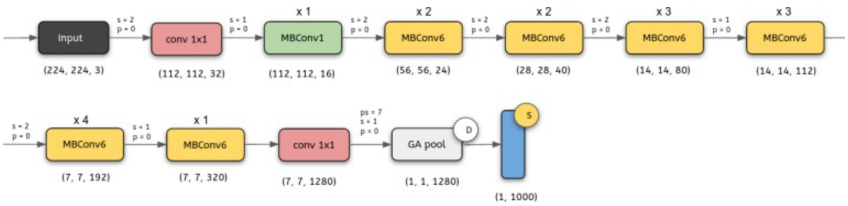


Figure 2.20: EfficientnetB0 architecture .

2.7.3 LeNet

LeNet was introduced by Yan LeCun for digit recognition. The basic configuration of LeNet-5 is in the figure 2.21, 2 convolutions layers, 2 sub-sampling layers, 2 fully connected layers, and an output layer with the Gaussian connection. The total number of weights and Multiply and Accumulates (MACs) are 431k and 2.3M respectively [64].

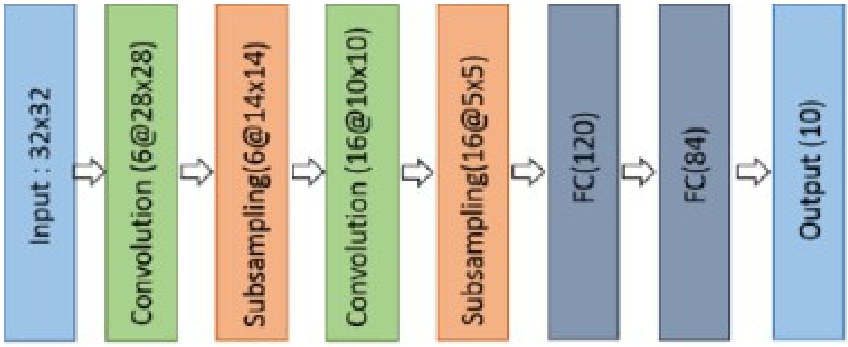


Figure 2.21: Basic architecture LeNet-5 .

2.7.4 MobileNet

MobileNet refers to a family of compact neural network architectures designed to work efficiently on devices with limited resources, like smartphones. Instead of using standard convolution layers, it splits them into smaller, more efficient operations: one that processes each channel separately (depthwise), and another that combines them (pointwise). This results in faster models with less memory usage [52].

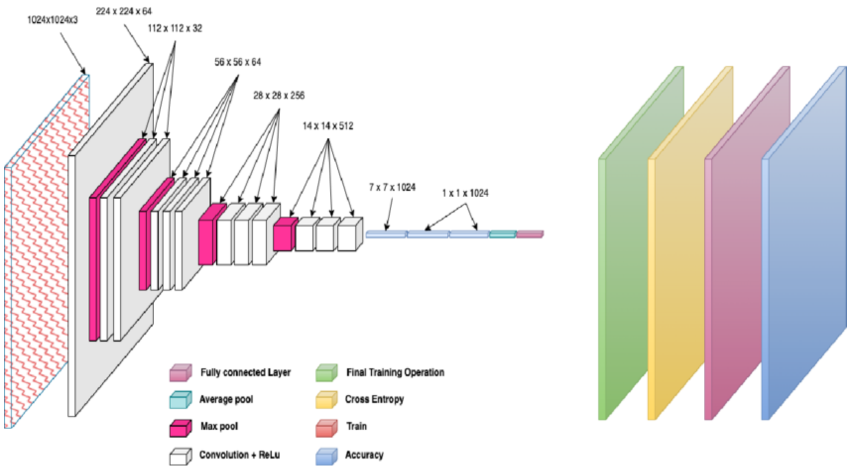


Figure 2.22: MobileNet architecture .

2.7.5 ResNet

ResNet stands for Residual Network, a deep learning structure built to overcome the problem of degraded performance in very deep networks. It introduces shortcuts that let information skip one or more layers. These shortcut paths help maintain gradient strength during training, making it possible to stack dozens or even hundreds of layers without the network's accuracy dropping [51].

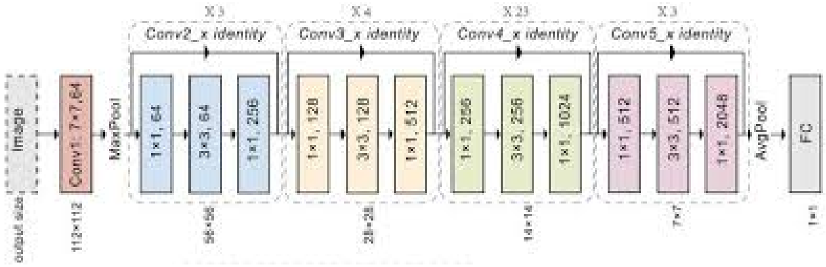


Figure 2.23: ResNet architecture .

2.8 Conclusion

Artificial intelligence, though still developing, has made remarkable progress despite early criticisms. Its potential to simplify daily life and push technological boundaries is immense. This chapter provided a brief overview of AI, leading into the next section on deep learning—one of the most impaction fields, unlocking opportunities once thought impossible. And has examined the theoretical foundations and real-world applications of deep learning methods for the identification of honey. The study examined the use of neural network techniques and their diverse applications in enhancing the precision and velocity of detection of pollen grains in honey. In the upcoming chapter, the focus will be on the techniques used to achieve Electron Microscopy (EM) detection of pollen grains. Key models to be developed, data preparation methods, and performance testing will be emphasized to ensure accurate and reliable results.

Chapter 3 :

System Design

Implementation and results

3.1 Introduction

The significant advancements in the field of artificial intelligence and its applications in various sectors, including environmental monitoring, have paved the way for innovative systems capable of predicting trends and aiding in decision-making processes. This chapter provides an overview of the system designed for the identification and classification of EM using microscopic images. Firstly, we will discuss the general architecture of our classification model. Following this, we will delve into the specific functionalities of our model, including dataset preparation, model training, performance evaluation, and model deployment.

3.2 Foctionalty architecture

The global architecture describes the microscopic image classification process. This process begins with the collection of a dataset comprising electron microscope images, which then undergo pre-processing to enhance their quality. The pre-processed data is subsequently split into training, testing, and validation sets. The training set is used to train the model to detect patterns and extract significant features. After training, the model's performance is analyzed using the testing and validation sets to ensure accuracy and generalizability. Based on this analysis, a decision is made: if the model meets the performance criteria, it is accepted and used to create an application for the detection of honey. If the model does not meet the criteria, it undergoes further training and refinement. This iterative process ensures the development of a robust and effective system.

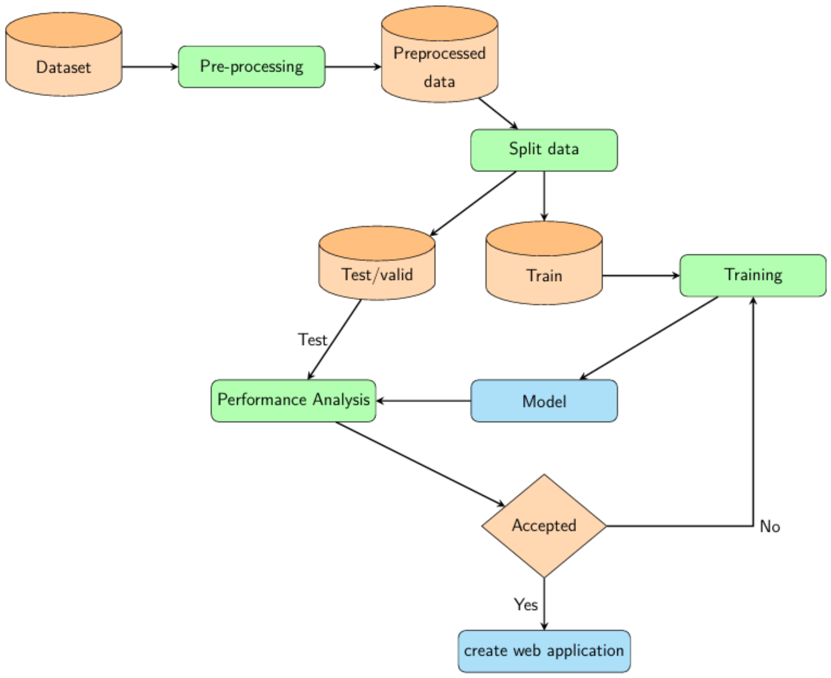


Figure 3.1: Global architecture.

3.3 Material and methods

3.3.1 Detailed CNN architecture

In this work, we constructed a tailored Convolutional Neural Network (CNN) specifically for the task of classifying pollen grain images into 23 distinct categories. The model architecture was carefully designed to maintain an optimal trade-off between accuracy and computational complexity, ensuring robust performance while minimizing the risk of overfitting due to the limited size of the available dataset.

The network structure features a sequence of convolutional and pooling layers to extract spatial features, followed by fully connected layers that handle the classification process. A detailed breakdown of the architecture is provided below, outlining each layer's function and configuration.

- **Convolutional layers**: These layers learn spatial hierarchies in the input images by detecting important visual patterns such as edges, textures, and shapes that help differentiate pollen grain types.
- **Pooling layers**: These layers reduce the spatial dimensions of the feature maps, lowering computational requirements while preserving essential features.
- **Fully connected (dense) layers**: Positioned after the convolutional and pooling blocks, these layers perform high-level reasoning on the extracted features to output class predictions.

- **Dropout regularization**: Introduced during training to reduce overfitting by randomly disabling a fraction of neurons, encouraging the model to build more robust representations.
- **Softmax activation**: Used in the final output layer to convert raw scores into probabilities across the 23 target classes, supporting confident and interpretable predictions.
- **Training strategy**: The model was trained using the Adam optimizer and categorical cross-entropy loss. Early stopping and learning rate scheduling were applied to improve generalization and dynamically adapt the learning process.

Layer (type)	Output Shape	Param #
conv2d_6 (Conv2D)	(None, 128, 128, 16)	448
max_pooling2d_6 (MaxPooling2D)	(None, 64, 64, 16)	0
conv2d_7 (Conv2D)	(None, 64, 64, 32)	2,080
max_pooling2d_7 (MaxPooling2D)	(None, 32, 32, 32)	0
conv2d_8 (Conv2D)	(None, 32, 32, 64)	8,256
max_pooling2d_8 (MaxPooling2D)	(None, 16, 16, 64)	0
conv2d_9 (Conv2D)	(None, 16, 16, 128)	32,896
max_pooling2d_9 (MaxPooling2D)	(None, 8, 8, 128)	0
conv2d_10 (Conv2D)	(None, 8, 8, 256)	131,328
max_pooling2d_10 (MaxPooling2D)	(None, 4, 4, 256)	0
conv2d_11 (Conv2D)	(None, 4, 4, 512)	524,800
max_pooling2d_11 (MaxPooling2D)	(None, 2, 2, 512)	0
flatten_1 (Flatten)	(None, 2048)	0
dense_8 (Dense)	(None, 1024)	2,098,176
dense_9 (Dense)	(None, 512)	524,800
dense_10 (Dense)	(None, 256)	131,328
dense_11 (Dense)	(None, 23)	5,911

Figure 3.2: Global CNN architecture.

3.3.2 Data description

The dataset referenced is a comprehensive collection of over 800 microscope images of pollen grains. It is designed to support research in automatic pollen classification and is particularly valuable in fields such as medicine, biology, agronomy, and environmental monitoring [43].

- **Total Images:** Approximately 800 segmented pollen grain images.
- **Image Resolution:** Each image is standardized to **128×128 pixels** in color.

- **Download :** ”https://zenodo.org/records/14950305/files/Pollen
- **Categories:** The dataset includes 23 classes:
- **Download :**

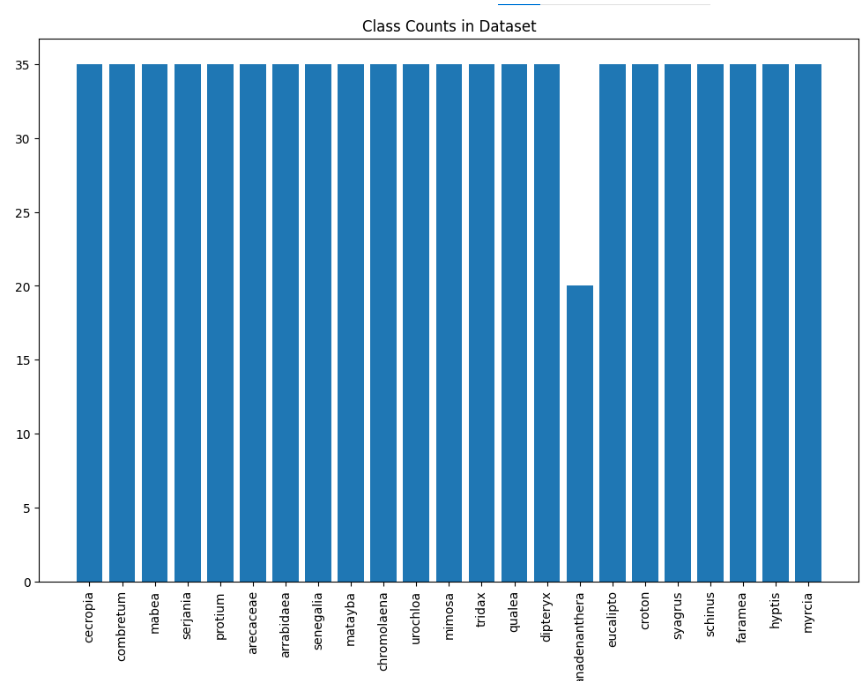


Figure 3.3: Pollen classes.

3.3.3 Data preprocessing and augmentation

- Pre-processing

  The pre-processing phase is a critical stage in image classification tasks, where we modify the data through processes such as resizing, applying filters, and removing noise before passing the data to the training model. The steps of pre-processing:

  - **Enumerate & label files:** Traverse the dataset directory, normalize each filename, extract its pollen class, and tally per-class image counts.
  - **Group file paths by class:** Build a dictionary mapping each class label to the list of all image paths in that class.
  - **Image resizing Goal:** Ensure all input images have the same dimensions.

    Reason: Deep learning models require a consistent input shape. However, images in a dataset often vary in size (e.g., 200×300, 512×512, etc.). To standardize the input for the model, each image is resized to a fixed resolution, such as 128×128 pixels. This resizing step guarantees compatibility with the architecture of the neural network.
  - **Image normalization:**

Goal: Convert pixel intensity values from the range $[0, 255]$ to the normalized range $[0.0, 1.0]$.

Reason: Machine learning models typically perform better when inputs are scaled to a smaller, consistent range. Pixel values in images are originally 8-bit unsigned integers ranging from 0 to 255. By dividing each pixel value by 255.0, we transform the data into floating-point numbers between 0.0 and 1.0. This normalization process improves training stability and speeds up convergence during model optimization.

  - **Load into image and label arrays:** Read each processed image into the features array and its class label into a parallel list.
  - **Convert labels to categorical format:** Map string labels to integer indices and then to one-hot vectors so they can be used with categorical loss functions.
  - **Compute class-balance weights:** Derive a weight for each class that inversely reflects its frequency, ensuring the model pays appropriate attention to underrepresented categories.

- Image Resizing

  Goal: Ensure all input images have the same dimensions.

  Reason: Deep learning models require a consistent input shape. However, images in a dataset often vary in size (e.g., 200×300, 512×512, etc.). To standardize the input for the model, each image is resized to a fixed resolution, such as 128×128 pixels. This resizing step guarantees compatibility with the architecture of the neural network.

- Image Normalization

  Goal: Convert pixel intensity values from the range $[0, 255]$ to the normalized range $[0.0, 1.0]$.

  Reason: Machine learning models typically perform better when inputs are scaled to a smaller, consistent range. Pixel values in images are originally 8-bit unsigned integers ranging from 0 to 255. By dividing each pixel value by 255.0, we transform the data into floating-point numbers between 0.0 and 1.0. This normalization process improves training stability and speeds up convergence during model optimization.

- Data Augmentation

  Data augmentation is a critical technique for increasing the size and diversity of training datasets in deep learning applications. By artificially expanding the dataset, it helps reduce overfitting and improves the generalization ability of neural networks. In our work, several augmentation methods were employed to enhance the quality and variability of the training data (see Figure **??**). These include traditional geometric transformations, pixel-level modifications, and advanced generative methods. The main techniques used are described below:

- Horizontal and vertical flip

  This augmentation method mirrors the image either across the vertical axis (left to right) or the horizontal axis (top to bottom), effectively altering its orientation.

  Implementation Details:

  - **Horizontal Flip:** The image is reversed along its vertical axis, swapping left and right sides.
  - **Vertical Flip:** The image is inverted along its horizontal axis, flipping the top and bottom sections.

  These flips are generally performed at random during training to introduce a wider range of visual variations into the dataset.

  Purpose:

  In many cases, especially when dealing with natural or biological subjects like pollen grains, the object may appear in different flipped positions. Since such objects often exhibit symmetry or random orientation in real-world scenarios, flipping helps the model become less sensitive to directionality.

  Effect:

  By learning from flipped versions of the same image, the model becomes more adaptable and capable of correctly identifying features regardless of how the object is oriented. This enhances its ability to generalize when processing new or unseen images.

- Random Rotations

  Images are rotated randomly by multiples of 90 degrees (90°, 180°, 270°) to train the model to recognize objects from different orientations. This increases the robustness of the model to rotational variations.

- Width and height shift This technique involves slightly moving the image horizontally (left or right) or vertically (up or down) by a small portion of its total size. The goal is to replicate real-world conditions where the subject in an image is not perfectly centered.

  Implementation Details:

  The shift value is generally defined as a decimal fraction. For instance, a value of 0.1 permits shifting the image by up to 10% of its width or height. When the image is shifted, empty regions may appear. These regions are typically filled using padding strategies, such as duplicating the edge pixels or using a constant background.
  Why It's Used:
  In practical scenarios, objects rarely appear in the exact center of the frame. Training a model with shifted versions of an image helps it recognize the object even when it's located in various parts of the image.

Resulting Benefit:
This augmentation method enhances the model's spatial awareness, allowing it to detect and classify objects irrespective of their exact position within the frame, which contributes to improved generalization performance.

- Splitting After all images and labels have been prepared and encoded, the dataset is split into training and testing sets in a single, reproducible step. Thirty-five percent of the samples are set aside for the test set, while the remaining sixty-five percent form the training set. This split is stratified on the original class labels to ensure that each subset maintains the same class proportions as the full dataset. The random shuffling of the samples provides a uniform distribution, and fixing a random seed guarantees that the exact same partition can be reproduced in future runs. Once the split is complete, class representation in the test set is verified by counting examples per label, and both training and testing arrays are saved to disk so they can be reloaded without needing to repeat the split.
- Train In deep learning, the training phase is essential for developing a model that can make accurate predictions on new data. This phase involves multiple steps, starting with data preparation and ending with model fitting. Data augmentation is a crucial technique in this process, used to artificially expand the size of a training dataset by creating modified versions of existing images. This enhances the model's performance by making it more robust to variations in the data. The typical architecture of the training phase includes collecting training data, applying data augmentation techniques, using the augmented data for training, and ultimately obtaining a fitted model

3.3.4 Test

The testing phase is when the trained model is assessed using the test set to gauge its performance. This step is crucial as it is the last one before the model is deployed in a real-world setting. To get a class prediction from the trained model, an image of the same dimensions as the training image is fed into the network

3.3.5 Transfer learning

Transfer learning refers to the process of adapting a model that was originally trained for one task to solve a different, yet related problem. Instead of building a model from the ground up—which often requires a large amount of labeled data and time—this approach makes use of an existing model that has already captured valuable patterns and features from a large dataset, such as ImageNet.

For example, in the field of image recognition, models like MobileNetV2, EfficientNet, ResNet50, and VGG16 are widely used due to their ability to detect visual elements like edges, shapes, and textures. These pre-trained models can be adapted to new tasks by reusing their learned representations.

There are generally two approaches:

- **Feature Extraction**: The original model's layers are kept unchanged, and only the final layers responsible for classification are modified and trained on the new dataset.
- **Fine-Tuning**: Some or all of the layers in the pre-trained model are retrained along with the new layers to better align with the target task.

This method is particularly helpful when working with small or specialized datasets. It speeds up training, improves performance, and lowers the risk of overfitting by making use of pre-existing knowledge. It has proven effective across many domains, including healthcare imaging, environmental analysis, and biological classification.

3.4 Model training and results

- Model training hyperparameters

  For the hyperparameters that we used to train our CNN model, we summarised them in the table 3.1.

Hyperparameter	**Value/Description**
Input Image Size	Resized to (128, 128)
Normalization	`/255` (scales pixel values to [0, 1])
Batch Size	32
Epochs	150
Learning Rate	0.01
Optimizer	Not explicitly defined, used Keras default `Adam`
Loss Function	`categorical_crossentropy` (multi-class classification)
Metric	`accuracy`
Callbacks	`EarlyStopping`: Stops if no improvement in `val_loss` for 10 epochs `ReduceLROnPlateau`: Reduces LR by a factor of 0.1 if there is no improvement for 4 epochs `ModelCheckpoint`: Saves best model by `val_loss`

Table 3.1: Hyperparameters used for model training

- Evaluation metrics

  We've put in place a number of indicators to make sure our system is sturdy and dependable. By comparing expected and actual results, these measures generate scores that indicate how effective the model is. They are used in a traditional data classification procedure in both the training and testing stages. Generally speaking, scores fall between 0 and 1, where 0 denotes subpar performance and 1 (or 100%) denotes perfect performance.

- Confusion Matrix

  A table known as a **confusion matrix** is frequently used to explain how well a classification model performs when applied to a set of test data for which the true values are known. It makes it possible to visualize how well an algorithm performs.

In the matrix, the examples in a **predicted class** are represented by each column, and the occurrences in an **actual class** are represented by each row.

There are four parts to the confusion matrix:

- **True Positives (TP)**: The number of observations correctly predicted as positive.
- **True Negatives (TN)**: The number of observations correctly predicted as negative.
- **False Positives (FP)**: The number of observations incorrectly predicted as positive.
- **False Negatives (FN)**: The number of observations incorrectly predicted as negative.

The confusion matrix is often presented in the following format:

Actual\Predicted	**Positive**	**Negative**
Positive	True Positive (TP)	False Negative (FN)
Negative	False Positive (FP)	True Negative (TN)

- Precision

The positive patterns that are successfully predicted from all of the projected patterns in a positive class are measured using precision. The equation for precision is given by:

$$\text{Precision (P)} = \frac{TP}{TP + FP} \tag{3.2}$$

- Recall

The percentage of positive patterns that are correctly categorized is measured by **recall**. The recall equation is provided by:

$$\text{Recall (R)} = \frac{TP}{TP + FN} \tag{3.3}$$

- F1-Score

The harmonic mean of recall and precision yields the **F1-score**, which strikes a balance between the two measures. It is computed with the following formula:

$$\text{F1-Score (F1)} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{3.4}$$

- Accuracy

  The ratio of accurate predictions to all instances analyzed is the measure of **accuracy**. It is computed with the following formula:

$$\text{Accuracy (Acc)} = \frac{TP + TN}{TP + FP + FN + TN} \tag{3.5}$$

3.4.1 Results

The following table 3.2 shows the results of the metrics (accuracy, precision, recall, F1 score) of our CNN model.

Class	Precision	Recall	F1-Score	Support
0	0.88	1.00	0.93	42
1	0.72	0.99	0.84	72
2	0.99	0.99	0.99	72
3	0.94	0.93	0.94	72
4	0.93	0.97	0.95	78
5	1.00	1.00	1.00	78
6	0.87	0.99	0.92	72
7	0.75	0.88	0.81	72
8	0.96	0.90	0.93	78
9	0.91	0.86	0.88	78
10	0.97	0.96	0.97	72
11	0.94	0.88	0.91	72
12	0.91	0.93	0.92	72
13	0.93	0.96	0.95	72
14	0.89	0.99	0.93	72
15	0.97	0.85	0.90	78
16	0.97	0.74	0.84	78
17	0.87	0.92	0.89	72
18	0.95	0.83	0.89	72
19	0.99	0.92	0.95	72
20	0.89	0.65	0.75	72
21	0.91	1.00	0.95	72
22	0.93	0.88	0.90	72
Accuracy			**0.91**	**1662**
Macro Avg	0.91	0.91	0.91	1662
Weighted Avg	0.92	0.91	0.91	1662

Table 3.2: Classification Report

3.5 Implementation

We discuss the implementation of deep learning models for detecting natural honey and evaluate the performance of these models using a specific dataset. The study focuses on comparing several advanced models, including pre-trained models and models combined with traditional machine-learning algorithms. In this chapter, we will review the implementation steps, from data collection and preprocessing to model training and evaluation. Additionally, we will discuss the performance of different models based on various metrics such as precision, recall, and classification accuracy.

subsectionDeployment of the Model using fastAPI and Render To deploy our deep learning model using FastAPI and host it on Render, we did these steps:

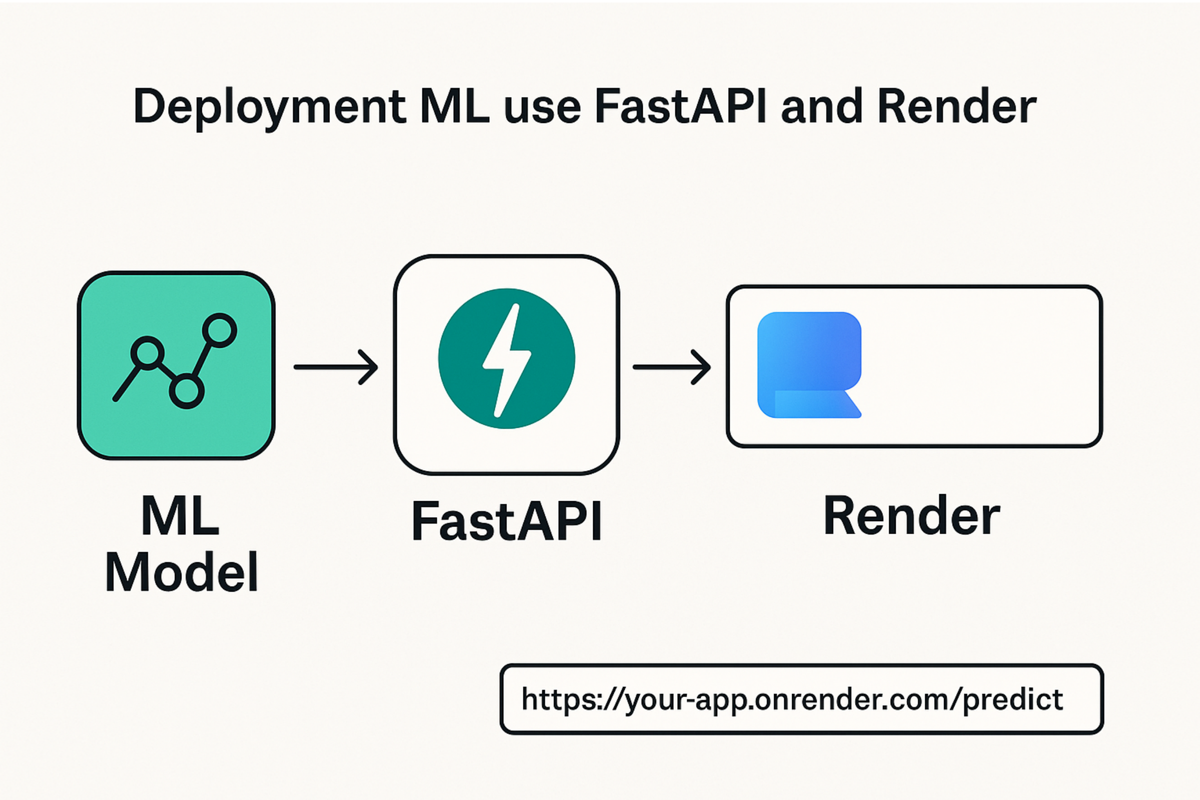


Figure 3.4: Deployment Model use fastAPI and Render

- Train and Save Our Model

  First, train our deep learning model and save it to a local file. Depending on the framework we using, the file format: is h5 for TensorFlow.

- Create the FastAPI

  Develop a FastAPI application that loads your saved model and defines an endpoint to receive input and return predictions.

- Add Dependencies

  Create a requirements.txt file with all the necessary Python packages: fastapi uvicorn NumPy TensorFlow pillow gunicorn python-multipart opencv-python Flask

- Version Control and Hosting

  Push the following files to a GitHub repository:

  - app.py.
  - model.h5.
  - requirements.txt.

- Deploy on Render

  Create a new Web Service and link your GitHub repository. n the Render configuration:

  Build Command: pip install -r requirements.txt

  Start Command: uvicorn main:app –host 0.0.0.0 –port PORT

  Render will automatically install dependencies, build the app, and make it publicly accessible.

- Test the Live API

  Once deployed, Render will give me a URL : https://application-6i19.onrender.com/predict
  Now we can send a POST request to this URL with JSON data to get predictions.

3.6 Mobile Application Development

3.6.1 Definition Of Mobile Applications :

Mobile applications are software programs designed specifically for use on mobile devices such as smartphones, tablets, or some computers that work with a Windows Phone or Chrome OS operating system. These applications can be downloaded and installed directly onto a mobile device, typically from an application store or other distribution platforms such as Google Play, Apple Application Store, or Microsoft Store, for free or for a fee. The ease with which they offer customers a variety of services, such as games, social networking, productivity tools, and many other kinds of content, makes mobile applications quite popular. They have become an essential part of our daily lives, providing us with quick and easy access to information and services whenever and wherever we need them. Mobile applications have the ability to integrate specialized features for users due to the hardware features found in mobile devices like cameras, GPS, and gyroscopes. This enhances the functional capabilities of the applications and allows for novel uses, including geolocation, Quick Response (QR) code scanning, augmented reality, and mobile commerce, which were previously not possible with information systems [102]

3.6.2 Why Mobile Applications Choice ?

This process enables:

- streamlined laboratory procedures without requiring the usage of complicated equipment or the transmission of physical samples.
- An intuitive interface provides quick and precise analysis findings.

Figure 3.5: app mobile for pollen detection.

3.6.3 The Pattern Design Used

In mobile applications, There are some software design patterns that are widely used, most notably the model-view-controller (MVC) pattern. Also, there are some other versions of the design patterns that are variations of the MVC [78]

- The Model-View-Controller (MVC)

  The Model-View-Controller (MVC) architecture, a cornerstone of modern software engineering, offers a robust framework for developing dynamic user interfaces. Recently, MVC become the most popular pattern for designing web applications and mobile apps. It was adopted in software development as a solution to separate presentation from application logic and data storage. Such separation allows multiple views of the same data. This eases the implementation, testing, and maintain multiple clients. [78]

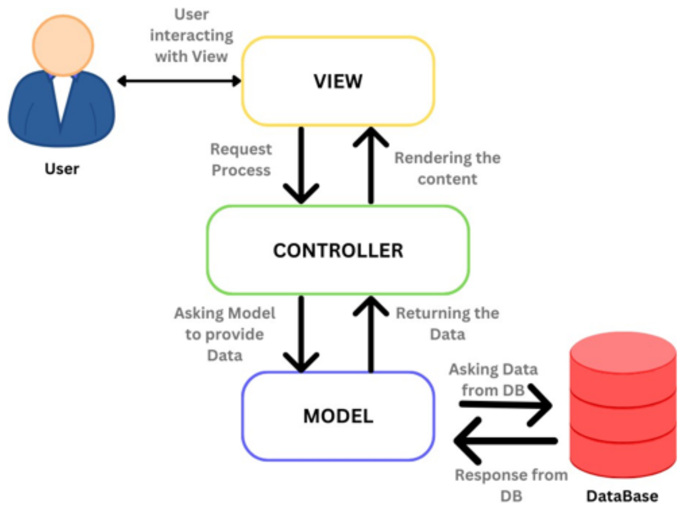


Figure 3.6: Model-View-Controller pattern.

Figure 3.8 demonstrates the division of the MVC pattern into three principal elements that are:

- **Model:** : component contains the data manipulated by the program. It ensures the management of this data and guarantees its integrity. In the typical case of a database, it is the model that contains it. The model provides methods for updating this data (insertion, deletion, value change). It also provides methods for retrieving its data.[78]
- **View**: The primary task of view is to display the data it has retrieved from the model on all the UI logic of the application. It also receives all user actions (mouse clicks, selection of entries, buttons,...). Its various events are sent to the controller. The view can provide multiple views, partial or complete, of the same data.[78]
- **Controller** : This component is responsible for synchronizing the model and the view. It receives all user events and triggers the actions to be performed. If an action Necessitates a change in the data, the controller requests to modify the data and then Reports the view that the data has changed so that it can update itself. There are certain events that do not concern the data but the controller requests to modify the view. The controller is often Split into several parts, and each one receives events from a subset of components. If the same object receives events from all components, it must determine the origin of each event. This sorting of events can lead to an Ungraceful code. To avoid this problem, the controller is divided into several objects[78]

• Advantages of MVC There are several advantages of MVC, we can cite [78]

  - MVC architecture helps us to regulate the complexness of application by dividing it into 3 parts i.e. model, view, and controller

- The application is modular, scalable, and easily reusable
- Allows multiple developers to work on the same project in parallel.
- Easy maintenance in terms of design and code

• Overview of Flutter with Firebase Platform for Mobile App Development :

  This diagram shows how the Model-View-Controller (MVC) architecture is used in a mobile application that was created with Firebase and Flutter.

  The user interface created with Flutter widgets is represented by the View at the front end. This is where users interact with the application, such as when they input their email address and password on the login screen. These inputs are sent to the Controller, which is made up of Firebase SDKs like cloud Firestore and Firebase auths, as requests.

  By managing communication between the View and the backend and handling business logic, the Controller serves as an intermediate. It makes queries to the Firebase NoSQL database (Firestore or Realtime Database), which is the Model, in order to store or retrieve data. Following the Model's response with data (such as the user profile or authentication result), the Controller processes it and sends a response back to the View to update the UI accordingly.

  The quick and scalable development of mobile apps is made possible by this MVC framework, which provides for a clear separation of concerns: Firebase SDK handles functionality and API calls, Flutter handles the user interface, and Firebase Database acts as the backend.

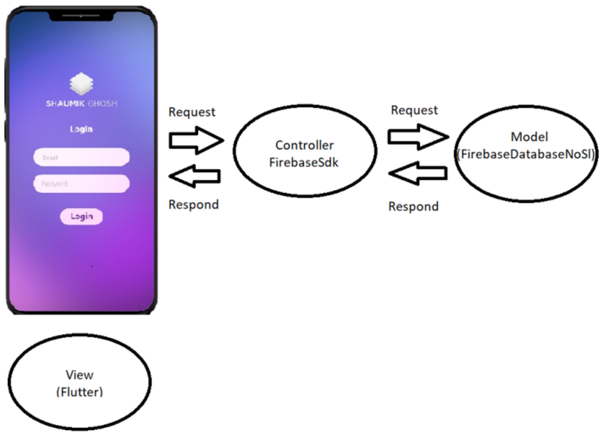


Figure 3.7: Architecture of Flutter-Firebase platform

3.6.4 Implementation tools and languages

We were able to familiarize ourselves with a variety of development methodologies and tools during the project's implementation phase, which required the use of specific programming domains. These are listed below:

- Python

  Python is a high-level, interpreted, object-oriented language with dynamic semantics. Its dynamic typing and dynamic binding, along with its high-level built-in data structures, make it an appealing language for Rapid Application Development and for usage as a scripting or glue language to join existing components. Because of its straightforward, basic syntax, Python promotes readability, which lowers software maintenance costs. Python's support for packages and modules promotes codere use and program modularity. The largest library and the Python interpreter are freely distributable and accessible for free on all major platforms in source or binary form [106].

Figure 3.8: python

- OpenCV

  A comprehensive open-source software library for computer vision and machine learning applications is called OpenCV, or Open Source Computer Vision Library. It was established in 2010 and offers a shared infrastructure to hasten the application of machine perception in items for sale. OpenCV provides both traditional and cutting-edge methods for applications like object identification, face recognition, human activity classification in films, and much more. It has over 2500 optimized algorithms. With more than 47,000 users and an estimated 18 million downloads, it is widely used by businesses, academic institutions, and governments all around the world. The library leverages specific hardware instructions and supports a variety of operating systems and programming languages, with a bias toward real-time vision applications. Furthermore, it fosters community involvement and offers multiple channels for assistance and cooperation, guaranteeing ongoing advancements and novelty in the domain of computer vision and machine learning [81].

Figure 3.9: OpenCV [81]

- Matplotlib

Matplotlib is a comprehensive library for creating static, animated, and interactive visualizations in Python. It makes easy things easy and hard things possible. It allows you to create publication-quality plots, make interactive figures that can zoom, pan, and update, customize visual styles and layouts, export to many file formats, and embed plots in JupyterLab and graphical user interfaces. You can also use a rich array of third-party packages built on Matplotlib. Matplotlib was created by neurobiologist John Hunter to work with EEG data, and it has grown to be used in many other fields [71]

Figure 3.10: Matplotlib [71]

- NumPy

NumPy is a fundamental Python open-source project that provides robust numerical computing capabilities. Established in 2005, it extends the functionality of earlier libraries while maintaining a commitment to open accessibility. Governed by a Steering Council, NumPy promotes community collaboration via GitHub, with various teams dedicated to development, documentation, and optimization. Funding from foundations and institutional partners ensures its continued growth. NumPy's significance lies in its pivotal role in scientific computing, offering users a vast array of manipulation functions and mathematical operations, rendering it an indispensable tool for numerical analysis and data manipulation tasks [79]

Figure 3.11: NumPy [79]

- Scikit-learn

An open-source machine learning library for the Python programming language is called Scikit-learn. David Cournapeau started the idea in 2007 as a Google Summer of Code project, and Matthieu Brucher expanded on it for his thesis. The first public release occurred on February 1st, 2010, after INRIA's Fabian Pedregosa, Gael Varoquaux, Alexandre Gramfort, and Vincent Michel assumed leadership in 2010. Numerous supervised and unsupervised learning methods for tasks like classification, statistical analysis, dimensionality reduction, and model selection are included in the collection. The NumFOCUS group provides financing for Scikit-learn, which is also funded by INRIA, Microsoft, and Nvidia. Donations are used to pay for organizational budget and other costs, such as workshop travel [84]

Figure 3.12: Scikit-learn [84]

- TensorFlow

  TensorFlow is an open-source machine learning platform that provides a flexible ecosystem of tools used to develop Artificial intelligence and deep learning applications, we have used that platform to generate our models [17].

Figure 3.13: TensorFlow [17]

- Keras

  Keras is a Python-based deep learning API that runs on top of the TensorFlow machine learning framework. It was designed to facilitate rapid experimentation, which is essential for successful research. As noted, the primary aim is to quickly transition from ideas to results [24].

  Keras is characterized by the following features:

  - **Simple:** While it is simple, it is not simplistic. Keras reduces the cognitive load on developers, enabling them to concentrate on the most critical parts of the problem.
  - **Flexible:** Based on the principle of *progressive disclosure of complexity*, Keras makes basic tasks quick and intuitive, while still allowing for advanced workflows through a logical and extendable path.
  - **Powerful:** Keras offers top-tier performance and scalability. It is trusted and utilized by major organizations such as NASA, YouTube, and Waymo.

Figure 3.14: Keras [24]

- Pandas

  Pandas (styled as pandas) is a software library written for the Python programming language for data manipulation and analysis. In particular, it offers data structures and operations for manipulating numerical tables and time series. It is free software released under the three-clause BSD license. The name is derived from the term "panel data", an econometrics term for data sets that include observations over multiple time periods for the same individuals, as well as a play on the phrase "Python data analysis" [72].

Figure 3.15: Pandas [72]

- Android Studio

  Android Studio is the official Integrated Development Environment (IDE) for Android App development. It is a powerful tool that allows developers to build high-quality applications for the Android platform. It has complete tools for the process of Android App development. From writing code to testing and deployment, ANdroid studio has all the functionalities for developers to develop an Android App [44].

Figure 3.16: Android Studio [44]

- Flutter

  Flutter is an open-source UI software development toolkit created by Google. It is used to build natively compiled applications for mobile, web, and desktop from a single codebase using the Dart programming language. Flutter is known for its fast development, expressive UIs, and hot-reload feature [47].

Figure 3.17: Flutter [47]

- Dart

  Dart is a programming language designed by Lars Bak and Kasper Lund and developed by Google. It can be used to develop web and mobile apps as well as server and desktop applications.

  Dart is an object-oriented, class-based, garbage-collected language with C-style syntax. It can compile to machine code, JavaScript, or WebAssembly. It supports interfaces, mixins, abstract classes, reified generics, and type inference [45].

Figure 3.18: Dart [45]

- Firebase

  Firebase was a company that developed backend software. It was founded in San Francisco in 2011 and was incorporated in Delaware.

  In 2014, Firebase was bought by Google. Its name continues as a set of backend cloud computing services and application development platforms provided by Google. It hosts databases, services, authentication, and integration for a variety of applications, including Android, iOS, JavaScript, Node.js, Java, Unity, PHP, and C++ [46].

Figure 3.19: Firebase

- FastAPI

  FastAPI is a modern web framework that is relatively fast and used for building APIs with Python 3.7+ based on standard Python-type hints. FastAPI also assists us in automatically producing documentation for our web service so that other developers can quickly understand how to use it. This documentation simplifies testing web services to understand what data they require and what they offer. FastAPI has many features like it offers significant speed for development and also reduces human errors in the code. It is easy to learn and is completely production-ready. FastAPI is fully compatible with well-known standards of APIs (i.e. OpenAPI and JSON schema) [91].

Figure 3.20: FastAPI [91]

- render

A render platform is a combination of software and hardware used to create visual content. This involves turning raw data—such as 3D models, textures, and lighting information—into finished images, animations, or graphics that can be viewed [12].

Figure 3.21: render [12]

- DrawSQl :

  Users can manage and visually construct database systems using easy-to-use entity-relationship diagrams (ERDs) with an online tool called DrawSQL. It allows developers and teams to create tables, define columns, create associations, and organize database structures without having to write raw SQL code. The software's ability to provide real-time collaboration makes it ideal for group tasks and system design. For instant use in their development process, users can choose to export their diagrams as SQL scripts or images. DrawSQL is especially useful in the early stages of application design, when a clear database structure is essential for successful implementation [29].

Figure 3.22: DrawSQl [29]

- Google Colab

  Google Colab is a free tool from Google that lets you write and run Python code right from your web browser—no need to install anything. It's based on Jupyter notebooks and is especially handy for data science, machine learning, and deep learning projects [48].

Figure 3.23: Google Colab [48]

- OverLeaf

  Overleaf is an online LaTeX editor that makes writing scientific documents, research papers, theses, and technical reports much easier. It runs entirely in your browser—so there's no need to install LaTeX on your computer [82].

Figure 3.24: OverLeaf [82]

- Github

  A web-based tool for collaborative software development and version management is called GitHub. It enables developers to collaborate on projects, track changes over time, save and manage their code using Git, and publish code online [42].

Figure 3.25: Github [42]

- Figma

  Figma is a powerful design tool that lets you create and collaborate on user interfaces and prototypes for websites, apps, and much more. Unlike traditional design software that requires installation, Figma runs entirely in your web browser, making it accessible on any platform, whether you're using Windows, Mac, or Linux. This platform's independence allows everyone to access and use it easily.[37].

Figure 3.26: Github [37]

3.6.5 Conception Of The Application

- Representation of Use Case Diagram:

  A use case diagram is a UML diagram used to provide an overview of all or part of the system requirements. It is also used to communicate the scope of a development project. It shows the relationships among actors and use cases within a system [100]

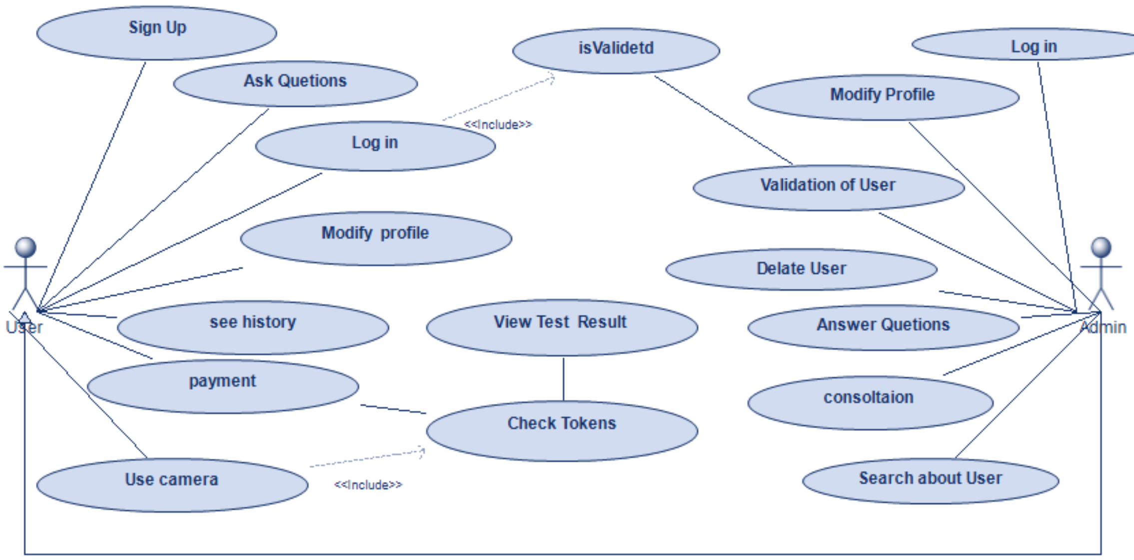


Figure 3.27: Usecases of the system

  - **User:** An individual accessing the platform to analyze honey, manage profile settings, and review results.
  - **Administrator:** A privileged operator responsible for maintaining the integrity of the system by managing user accounts and overseeing interactions.

  Functional Scenarios for Users

  Participants under the user category have access to the following set of actions:

  - Account creation to initiate access rights.
  - Entry authentication to access personalized services.
  - Submission of inquiries to system administrators.

- Updating of profile attributes when needed.
- Ability to track historical records and past test activities.
- Executing financial transactions related to system services.
- Utilizing the device camera to capture sample images.
- Viewing outcomes related to honey purity assessments.
- Monitoring token balances or usage status for available features.

Functional Dependencies

Some tasks involve preconditions or shared logic:

- Logging in is connected to an internal validation process, ensuring credentials are correct.
- Camera access is dependent on a token verification step before proceeding.

Administrator Responsibilities

For system administrators, the following capabilities are defined:

- Authentication to enter the administration dashboard.
- Approving or confirming user identities.
- Removing access rights from selected accounts.
- Replying to user messages and questions.
- Providing supportive feedback or guidance.
- Investigating user activity or information.
- Reviewing the test outcomes available in the system.
- Checking and evaluating token data relevant to services.

• Representation Sequence diagram :

A sequence diagram can map scenarios described by a use case in step-by-step detail to define how objects collaborate to achieve the application's goals.[100]

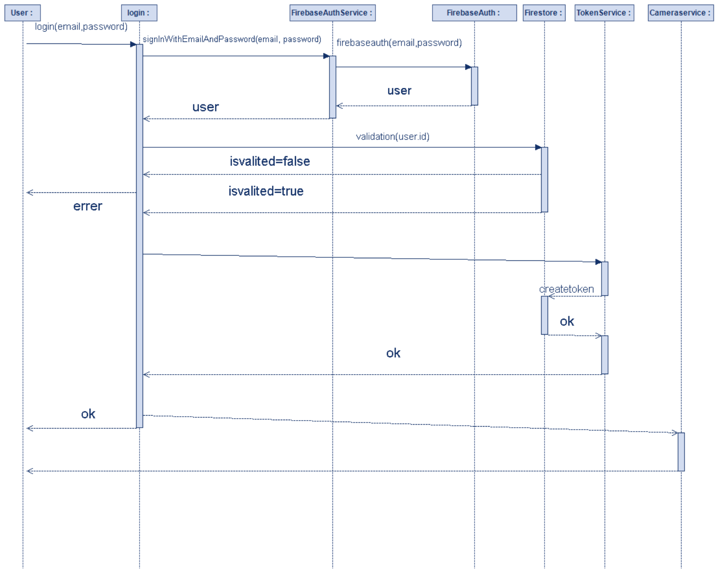


Figure 3.28: Sequence diagram of login.

Login Sequence Flow :

1. **User → login:** The user initiates a login request using their email and password.
2. **login → FirebaseAuthService:** The login method forwards these credentials to the authentication handler.
3. **FirebaseAuthService → FirebaseAuth:** The authentication handler passes the request to Firebase's backend.
4. **FirebaseAuth → FirebaseAuthService:** Firebase returns a user object indicating successful login.
5. **FirebaseAuthService → Firestore:** A query is sent to Firestore to verify the status of the user.
6. **Firestore → FirebaseAuthService:** First, the user is marked as not validated. Then, a second validation attempt confirms approval.
7. **FirebaseAuthService → TokenService:** Upon confirmation, a request to generate a session token is made.
8. **TokenService → FirebaseAuthService:** A token is generated and returned to the authentication handler.
9. **FirebaseAuthService → login:** A successful response is sent back to the login module.
10. **login → User:** The user receives confirmation of successful login and can access to camera.

- Representation of database using Conceptual Data Model (CDM) :

A conceptual data model (CDM) is a high-level representation of an organization's data. It focuses on identifying key business concepts, system entities, and their relationships.[80]

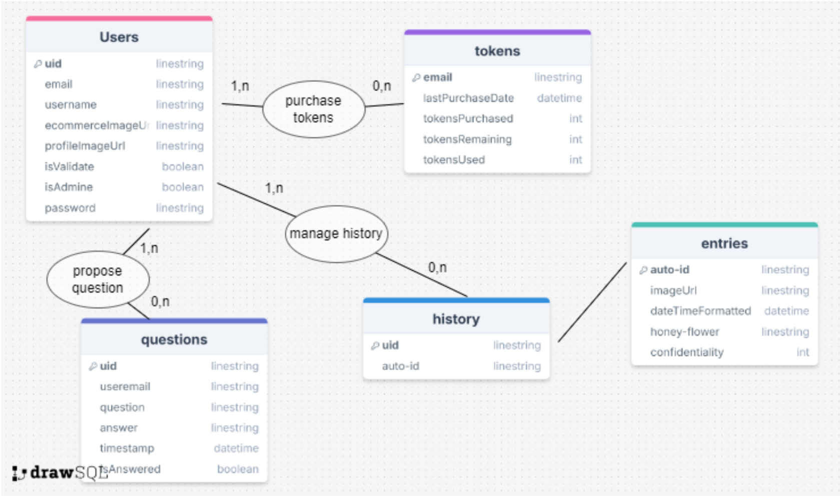


Figure 3.29: Database CDM.

- **User** (<u>uid</u>, email, username, ecommerceImageUrl, profileImageUrl, isValidate, isAdmin, password)
- **question** (<u>uid</u>, username, question, answer, timestamp, isAnswered)
- **tokens** (<u>email</u>, lastPurchasedData, tokensPurchased, tokensRemainding, tokensUsed)
- **history** (<u>uid,#auto-id</u>)
- **entries** (<u>auto-id</u>, imageUrl, dateTimeFormatted, honey-flower, confidentiality)

Remarque:

in User collection

- admin : "isAdmin" attribute must be true and "isValidate" attribute true
- user not validate : "isAdmin" attribute false and "isValidate" attribute false
- user validate : "isAdmin" attribute false and "isValidate" attribute true

In the history collection:

- uid is the primary key – it means that each record in history is directly linked to a specific user and is uniquely identified by this field.
- auto-id is a foreign key that refers to a record in the entries collection.

In the entries collection:

- auto-id is the primary key that uniquely identifies each analysis.

3.7 Application graphical interfaces

In this section, we provide some screenshots of our application to show the graphical user interface of the different parts of the application.

3.7.1 Splash Screen :

The splash screen represents the initial screen displayed when launching the application. It consists of our logo and app name, This screen appears for 5 seconds before automatically navigating to the home page.

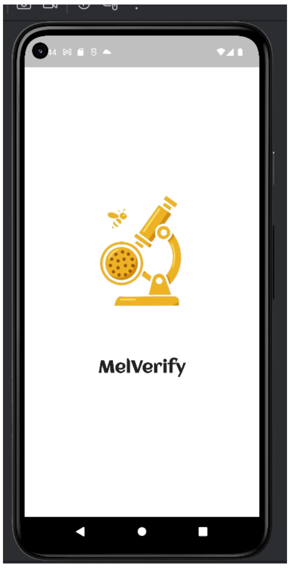


Figure 3.30: splash screen

3.7.2 Home Page :

On this page, the client can create a new account or just sign in to his account.

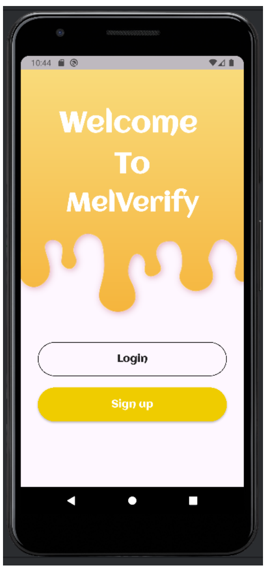


Figure 3.31: home page

3.7.3 Sing up :

In this interface, the user can create a new account by filling in his/her name, email, password, and a certificate (commerce proof) that proves that he/she works at a laboratory or owns one.

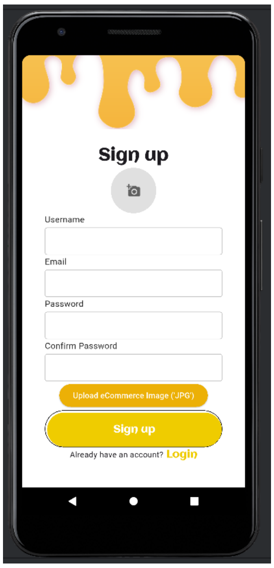


Figure 3.32: Sign-up page

- SingupExeption : Email address is already in use If the new user enters an existing email an exception will occur and a message will show and tell him that the email exists.

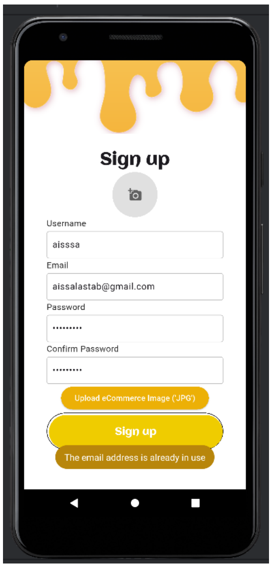


Figure 3.33: already used email

- SingupExeption : badly formatted email When the user enters a badly formatted email the message will occur.

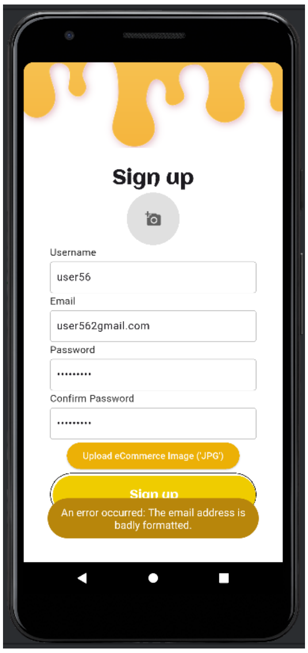


Figure 3.34: Bad formatted email

- Fields Empty : If the fields are empty the signing up will not be done unless they are all filled.

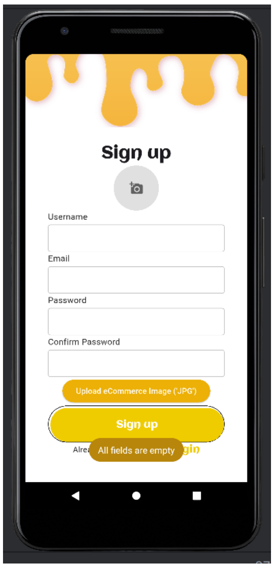


Figure 3.35: empty fields

- Some Fields Empty : Here if one of the fields (email, Password, or commerce file) is missing, the sign-up will be not done.

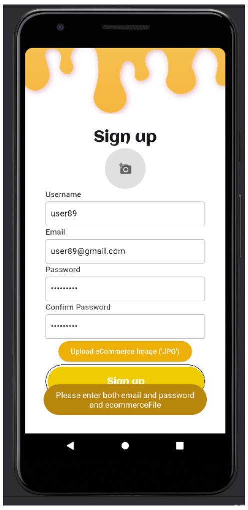


Figure 3.36: some empty fields

- Password do not much : This message (password do not much) appears when the user makes a mistake in entering the password and its confirmation (a miss-match).

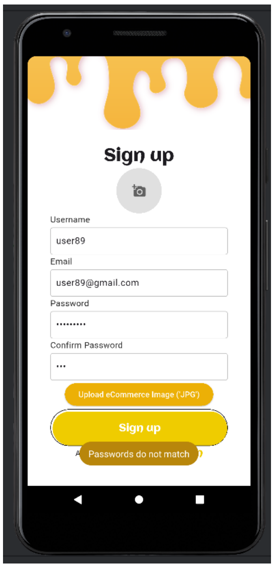


Figure 3.37: password not much

- Password is too weak : Here the user must enter a password that consists of at least 6 characters to sign up, to make the password more strong and secure.

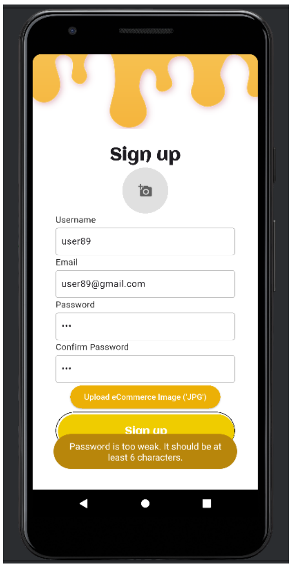


Figure 3.38: weak password

- Account created successfully interface Here, this page appears when the user successfully creates his account where it's saved in our system, and he can download the user guide to view how our application works.

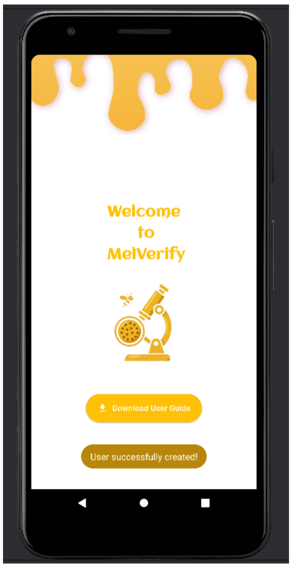


Figure 3.39: Account created successfully interface

- User guide interface Our user-guide documents will show the user, where he can understand how the application works(PDF file could be downloaded in the application browser)

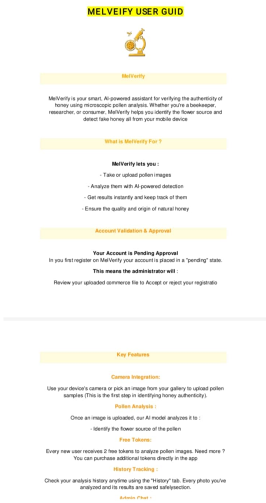


Figure 3.40: User guide interface

3.7.4 Login Page :

In this interface, the user can sign in to our application by his/her account by filling in (email, password), if the admin already accepted his request.

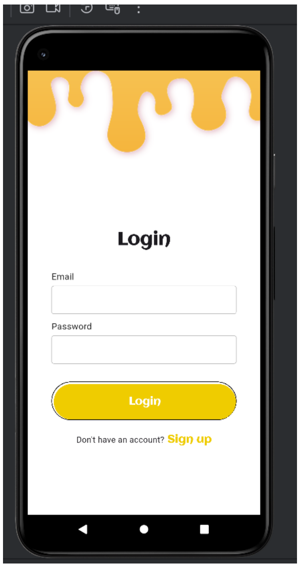


Figure 3.41: login page

- Login Exception: Incorrect Email When the user enters a non-existent email (hasn't signed yet) the message will show that the account doesn't exist.

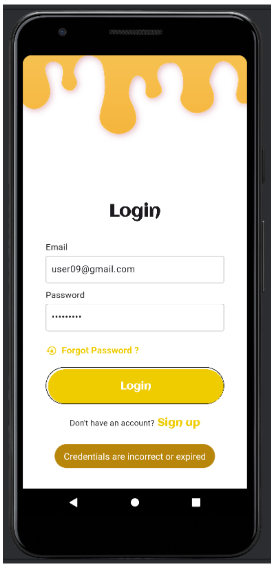


Figure 3.42: incorrect email

- Login Exception: Forgot Password Here when the user forgot his/her password he can get another password by email.

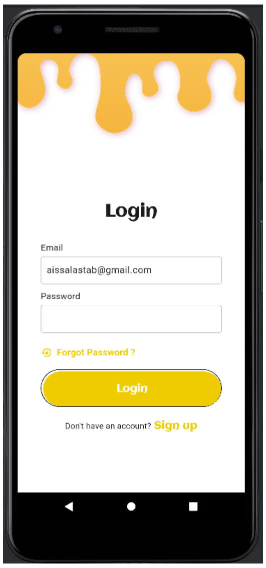


Figure 3.43: Forgotten password interface

* Here our system sends the user an email so he can change the password.

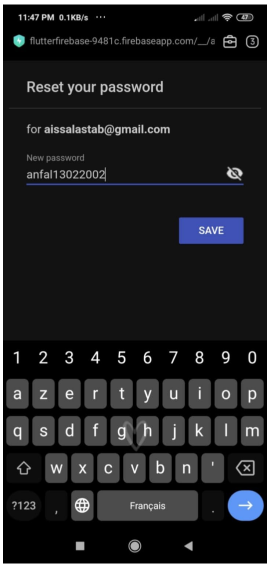


Figure 3.44: Creating a new password interface

* So here, the user has to change the password and now he can go back to login page in the application.

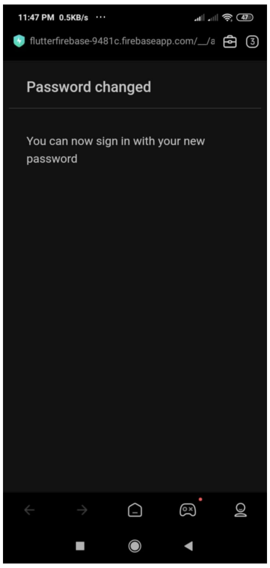


Figure 3.45: Successfully password changed interface

- Login Exception: Pending Account This interface appears when the user is not validated by the admin so he can not login in.

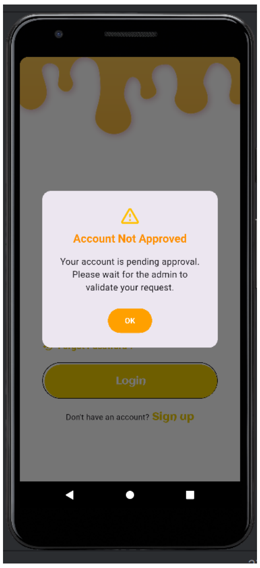
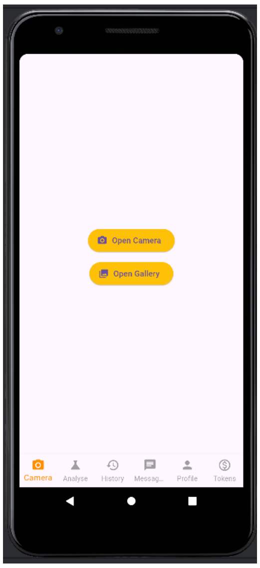


Figure 3.46: account not approved

3.7.5 CameraPage

Here the user can select a photo from Gallery or take a photo by camera.



- Camera Details: Cropping Image Here the user can use the cropping image technique to crop the pollen image that he wants to analyze.

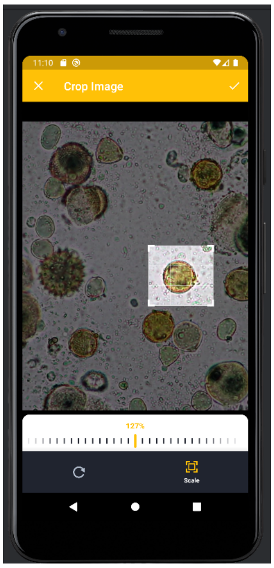


Figure 3.47: cropping image

- Camera Details: Display cropping image so here after cropping, the image appears here in this interface to confirm if the user wants to analyze this pollen image.

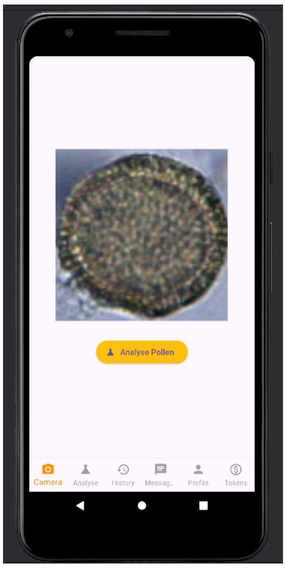


Figure 3.48: Analyzing pollens interface

3.7.6 Analyzing Result Page

In this interface, the application displays the result of pollen analysis.

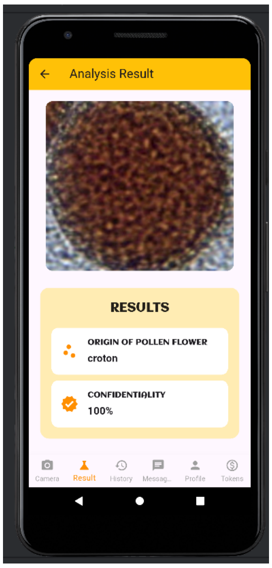


Figure 3.49: pollen analyzing result

3.7.7 History Page :

Here the user can see his/her history of pollen analysis results.

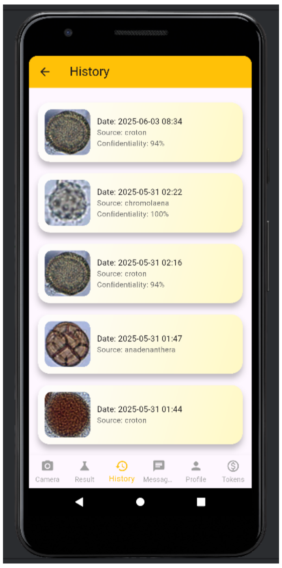


Figure 3.50: user history

3.7.8 Chat Page :

A page to contact the admin and ask him any questions about the application.

Figure 3.51: user chatting with admin

3.7.9 Profile Page

On this page, the user can view or edit his profile.

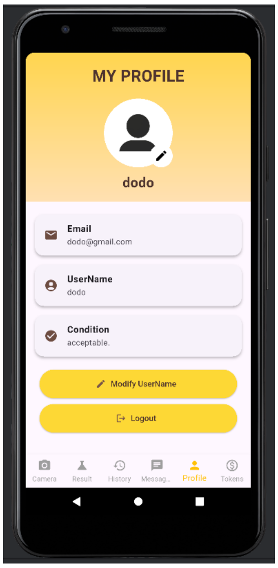


Figure 3.52: user profile

3.7.10 Tokens Page :

In this interface, the user can buy tokens to access the camera and gallery for analysis later.

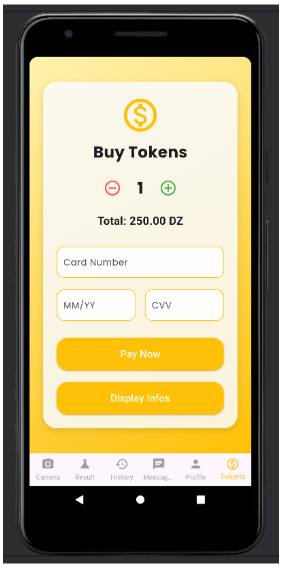


Figure 3.53: purchase tokens page

- Tokens Details: Incomplete Cart Information : An error message will appear if some fields of the cart are empty.

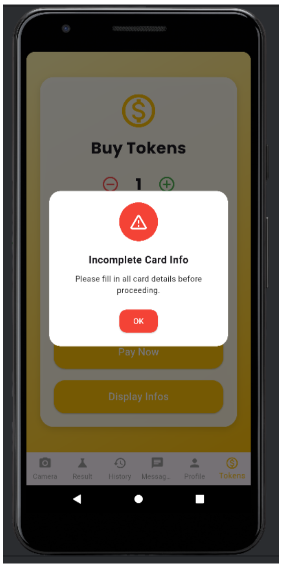


Figure 3.54: incomplete cart

- Tokens Details: Confirm tokens purchase A message appears here in this interface to confirm a token purchase.

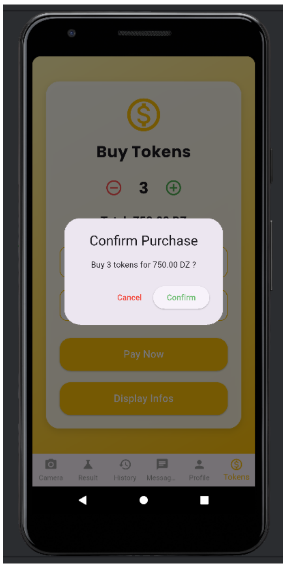


Figure 3.55: confirm purchase

- Tokens Details: Display user tokens information Here the user can see the statistics about his tokens: the number of purchased tokens, remaining, and the used tokens.

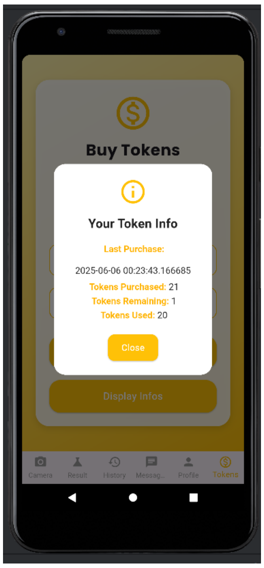


Figure 3.56: confirm purchase

3.7.11 Admin :

The admin can manage users through a web browser interface. The tasks that could be performed include:

- Admin Login : This admin login page is where he can sign in with an email address and password.

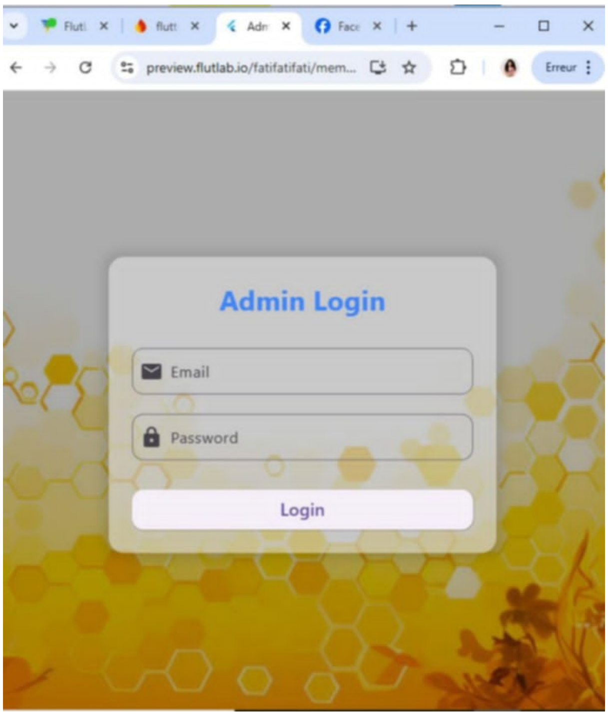


Figure 3.57: admin login

- Admin Home page : This admin panel is designed to help administrators manage users, validate accounts, track conversations, monitor feedback, and view their personal profiles in an organized, interactive interface.
  - **Sidebar Navigation:** A vertical menu allows switching between the main admin tasks:
    * Viewing user profiles.
    * Validating new users.
    * Accessing chat conversations.
    * Checking user's reviews and financial stats.
  - **Search Functionality:** The admin can search for users by their username in real-time and instantly access detailed user profiles.
  - **Live Notifications:**
    * Displays the number of pending user validations.
    * Shows the count of unanswered chat questions, both updated in real-time via Firestore streams.
  - **Profile Quick Access:** Admin information, including profile picture, name, and email, is shown at the top of the sidebar. Tapping it redirects the admin to their personal profile page.
  - **Responsive Content Display:** The main screen dynamically updates based on the selected menu item or search result.
  - **Secure Logout:** An easy-to-access logout button signs the admin out and redirects them to the login screen.

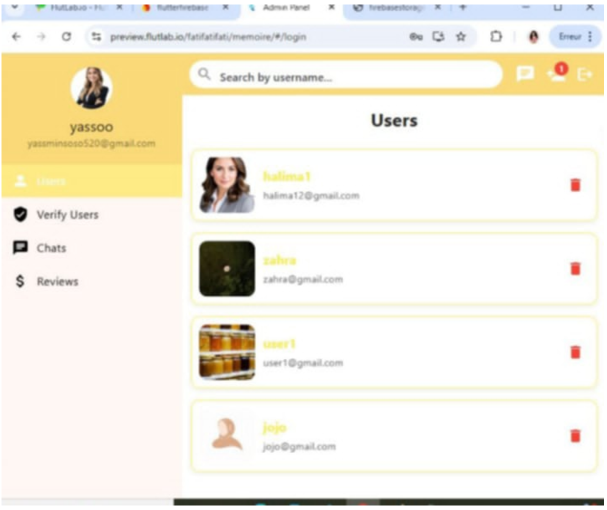


Figure 3.58: admin page

- Admin Search bar : The admin can search for users by their username in real time and instantly access detailed user profiles.

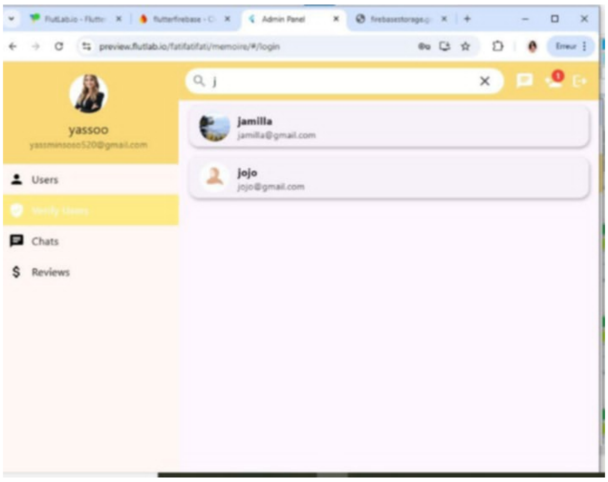


Figure 3.59: search bar

- Users Details here in this page the admin can see the users information(user tokens information, user history, email, profile photo, validation state)

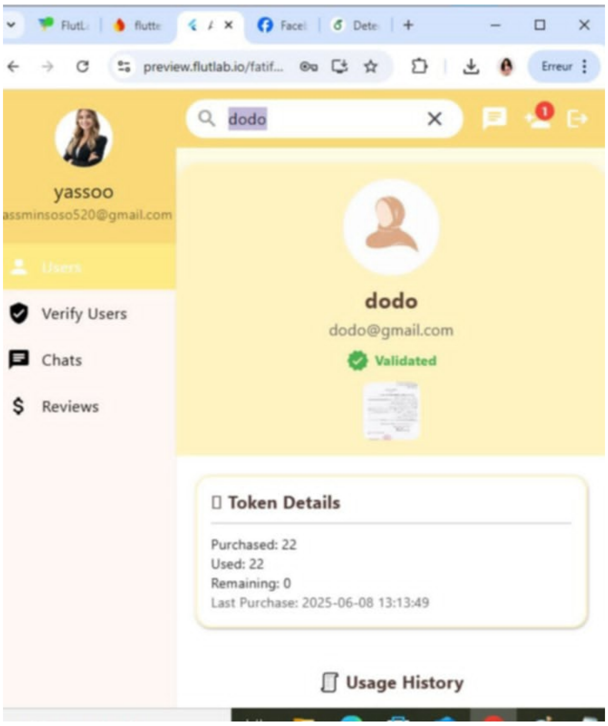


Figure 3.60: User details interface

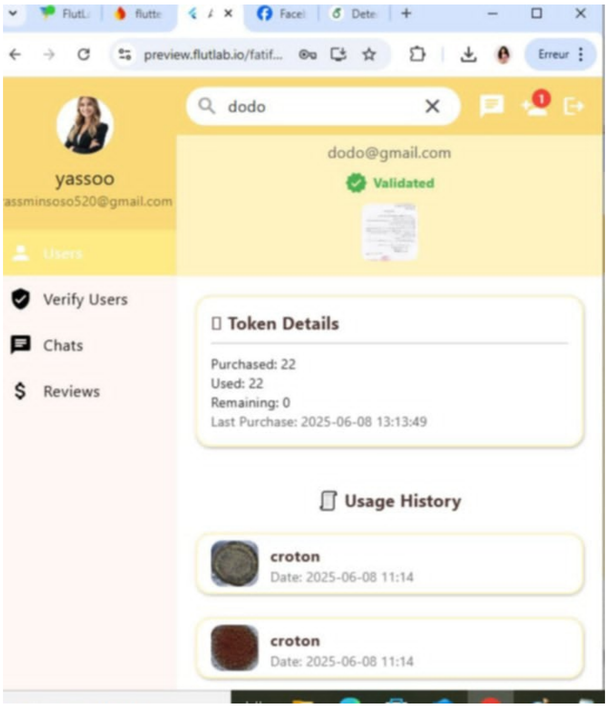


Figure 3.61: User details interface

- List of validated Users So here, is the validated user list by the admin and he can delete any one of them from our system after checking the commerce certificate.

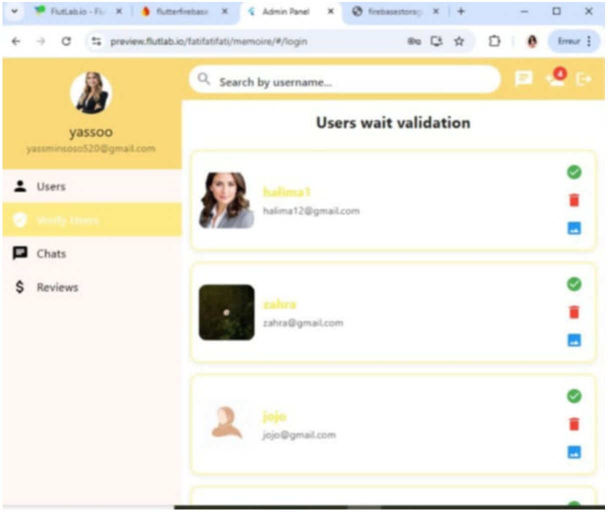


Figure 3.62: validated users list by admin

- Profile admin page The Admin profile provides essential personal details of the administrator, including their profile picture, full name, and email address. It allows the admin to view and update their information.

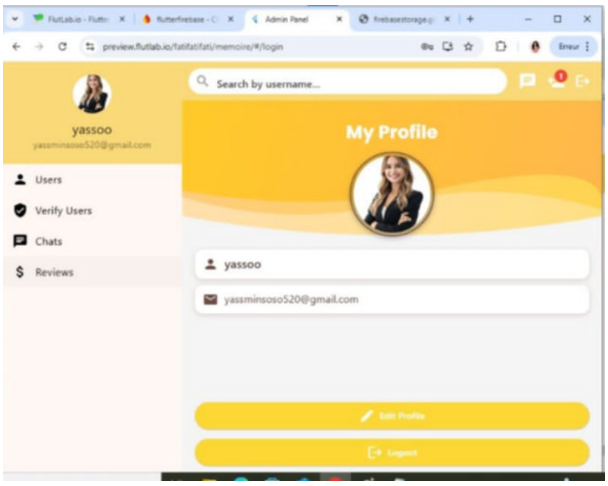


Figure 3.63: profile admin page

- Profits monitoring page : Here the admin can see the statistics of profits yearly, weekly, or daily.

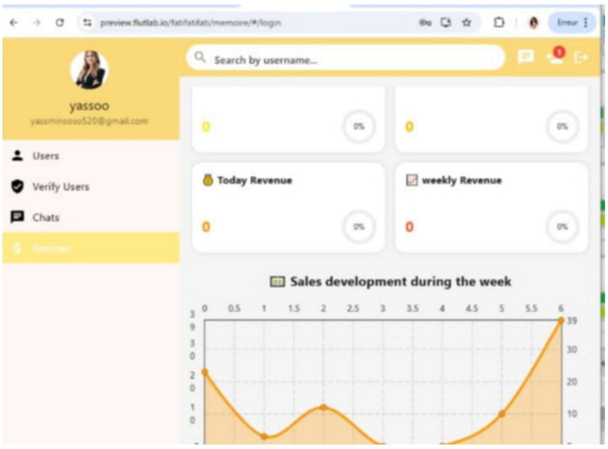


Figure 3.64: Profits monitoring

- Chat With User : Here the admin can contact the users if he has any questions or queries.

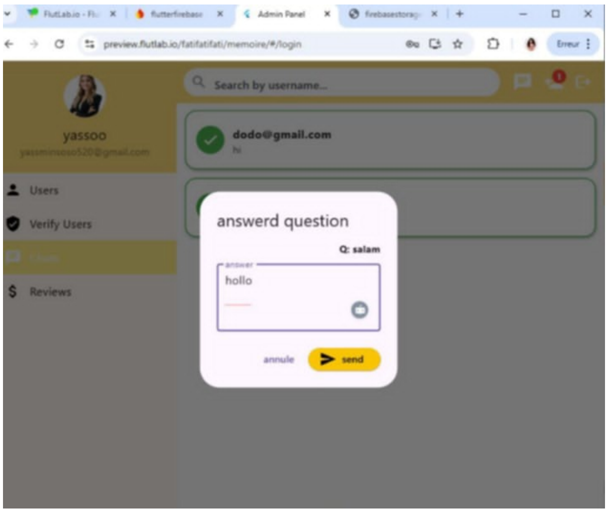


Figure 3.65: chat page

3.8 Conclusion:

We presented the various functionalities and user interfaces offered by our mobile application, which is designed to manage users effectively. Our software functions as a comprehensive system for the detection and identification of natural honey by analyzing pollen content. It provides a user-friendly platform that facilitates the management of honey analysis workflows, making it easier for laboratories and certified users to interact with the system.

Beyond basic user management, the application also streamlines the process of uploading, analyzing, and retrieving results, all within an intuitive interface. Integrating AI-powered pollen classification, supports accurate identification of the botanical origin of honey, helping to detect adulteration and ensure product authenticity. In the future, additional features such as segmentation and automated pollen counting will be incorporated to enhance the precision of the assessments.

General Conclusion

In this project, we created the clever application **MelVerify** with the goal of utilizing artificial intelligence methods to confirm the quality of natural honey. The development process was arranged using the **Model-View-Controller (MVC)** architecture, which separates the interface, data, and control logic.

Our project's **Model** is Firebase, which uses cloud databases and storage services to store user information, image analysis findings, and interaction history.

The **View** was created using Flutter, which offers slick, appealing, and intuitive user interfaces that let users upload photographs, log in, and view results.

Firebase packages for Flutter perform the **Controller** role by handling the communication between the user interface and the backend database, including result retrieval, image upload, account creation, and login.

In order to facilitate the scheduling and payment of analysis services directly within the app, our long-term goal is to incorporate an electronic payment system for client-certified laboratory interactions.

The **AI model** was trained by analyzing tiny photos of pollen in honey using a **Convolutional Neural Network (CNN)**. By examining the characteristics of individual pollen grains, our approach currently aids in determining the honey's floral source.

Based on the variety and distribution of pollen sources present in the image, we hope to improve the model in the future to evaluate a single image containing multiple pollen grains and identify whether the honey is natural or adulterated.

Future perspectives include the integration of pollen segmentation and automatic pollen counting within the system. This enhancement will allow the determination of the percentage of natural content in honey by estimating the proportion and diversity of pollen types. Such features will provide a more comprehensive and quantitative assessment of honey authenticity, further supporting laboratories in their quality control and certification processes.

Bibliography

[1] 4. fully connected deep networks - tensorflow for deep learning [book]. Accessed: 04/11/2022.

[2] Activation function. Accessed: 04/04/2022.

[3] Image classification with efficientnet: Better performance with computational efficiency.

[4] Overview of neuron structure and function. Accessed: 2025-06-09.

[5] The relation between artificial and biological neuron. Accessed: 2025-06-09.

[6] Softmax function. Accessed: 04/04/2022.

[7] Biological vs artificial neural networks — a comparison — neural network, 2020. Accessed: 2022-04-05.

[8] Automatic log analysis using deep learning and ai, 2022. Accessed: 2022-04-04.

[9] A complete history of artificial intelligence, 2022. Accessed: 2022-03-24.

[10] A complete history of artificial intelligence, 2022. Accessed: 2022-03-24.

[11] How do computers see an image? — commonlounge, 2022. Accessed: 04/09/2022.

[12] Rendering (computer graphics), 2025. Accessed: June 9, 2025.

[13] *4. Fully Connected Deep Networks - TensorFlow for Deep Learning [Book]*, n.d. Accessed: 2022-04-11.

[14] Automatic log analysis using deep learning and ai, n.d. Accessed: 2022-04-04.

[15] *Getting Started with R-CNN, Fast R-CNN, and Faster R-CNN - MATLAB & Simulink*, n.d. Accessed: 2022-04-11.

[16] How do computers see an image? — commonlounge, n.d. Accessed: 2022-04-09.

[17] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, Manjunath Kudlur, Josh Levenberg, Rajat Monga, Sherry Moore, Derek G Murray, Benoit Steiner, Paul Tucker, Vijay Vasudevan, Pete Warden, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. Tensorflow: A system for large-scale machine learning. In *12th USENIX Symposium on Operating Systems Design and Implementation (OSDI)*, pages 265–283, 2016.

[18] Peter Aspin. Navigating economic challenges: Global honey prices and the future of beekeeping in 2025. 2025.

[19] A. Author and B. Coauthor. Ultrafiltration of honey: A review of techniques and effects. *Journal of Food Engineering*, Year/Volume info:Page numbers, 2003.

[20] Various Authors. Published studies in food science and analytical chemistry journals, 2025. Accessed: February 26, 2025.

[21] L. Boukraa and A. Benarfa. Physicochemical and melissopalynological study of some honey samples from the algerian east region. *Turkish Journal of Agriculture - Food Science and Technology*, 11(5):1–9, 2023.

[22] C. Camacho. Convolutional neural networks, 2022. Accessed: 04/08/2022.

[23] C. Camacho. Convolutional neural networks, n.d. Accessed: 2022-04-08.

[24] François Chollet et al. Keras, 2015. Accessed: June 19, 2024.

[25] Codex Alimentarius Commission. *Standard for Honey*. FAO & WHO (International Food Standards), 2001.

[26] de L'Ever. To debunk this test, we investigated if it could detect crystallized honey that has been adulterated with water or high fructose corn syrup (hfcs). *The University of British Columbia Journal*, 2021.

[27] A. Deshpande. A beginner's guide to understanding convolutional neural networks. Accessed: 04/10/2022.

[28] A. Deshpande. A beginner's guide to understanding convolutional neural networks, n.d. Accessed: 2022-04-10.

[29] DrawSQL Team. Drawsql: Collaborative online erd tool with real-time editing and export capabilities, 2025. Accessed: June 9, 2025.

[30] K.-L. Du and M. Swamy. Reinforcement learning. In *Neural Networks and Statistical Learning*, pages 547–561. Springer, 2014.

[31] K.-L. Du and M. Swamy. Reinforcement learning. pages 547–561. December 2014.

[32] Fera. Hydroxymethylfurfural (hmf) test for honey. 2025. Accessed: 2025-04-10.

[33] Food and Agriculture Organization (FAO). Food authentication research reports, 2025.

[34] Food, Agriculture Organization of the United Nations, and World Health Organization. Codex alimentarius. 2025.

[35] Food and Agriculture Organization (FAO). Honey authentication and market challenges, 2025. Accessed: February 26, 2025.

[36] Food and Agriculture Organization (FAO). Reports on honey fraud and its impact on global markets, 2025. Accessed: February 26, 2025.

[37] GeeksforGeeks. What is figma?, 2023.

[38] J. P. Adam Gibson. *Deep Learning: A Practitioner's Approach*. O'Reilly Media, Inc., Sebastopol, CA, 2017.

[39] J. P. Adam Gibson. *Deep Learning: A Practitioner's Approach*. O'Reilly Media, Inc., 1005 Gravenstein Highway North, Sebastopol, CA 95472, first edition edition, July 2017.

[40] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. *arXiv preprint arXiv:1311.2524*, 2014.

[41] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. *arXiv:1311.2524 [cs]*, October 2014. arXiv: 1311.2524.

[42] GitHub. Github: Web-based platform for version control and collaboration, 2025. Accessed: June 9, 2025.

[43] Ariadne Barbosa Gonçalves, Junior Silva Souza, Gercina Gonçalves da Silva, Marney Pascoli Cereda, Arnildo Pott, Marco Hiroshi Naka, and Hemerson Pistori. Feature extraction and machine learning for the classification of brazilian savannah pollen grains. *PloS one*, 11(6):e0157044, 2016.

[44] Google. Android studio: The official ide for android, 2024. Accessed: June 19, 2024.

[45] Google. Dart programming language, 2024. Accessed: June 19, 2024.

[46] Google. Firebase: Backend-as-a-service platform, 2024. Accessed: June 19, 2024.

[47] Google. Flutter - beautiful native apps in record time, 2024. Accessed: June 19, 2024.

[48] Google Colab Team. Google colab: Free cloud-based jupyter notebook environment with python and gpu/tpu support, 2025. Accessed: June 9, 2025.

[49] Mokhtar Guerzou. Title of the pdf document, 2025.

[50] Laouar Hadia. Analyse des caractéristiques physico-chimiques du miel en algérie, 2019.

[51] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, pages 770–778, 2016.

[52] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017.

[53] IEEE. Advancements in ai for food quality control, 2025.

[54] International Honey Commission (IHC). *Scientific Methods for Honey Purity Testing*. 2025. Accessed: February 26, 2025.

[55] Shinji Itoh, Mika Terakawa, Takao Masaki, and Masaki Yamazaki. Characterization of honey ultrafiltration process. *Nippon Shokuhin Kagaku Kogaku Kaishi*, 47(6):431–438, 2000.

[56] Ukrainian Food Journal. Spectroscopic analysis of honey. *Ukrainian Food Journal*, 5(3):438–450, 2016.

[57] Journal of Agricultural and Food Chemistry. *Studies on Honey's Antioxidant and Medicinal Properties*. Accessed: February 26, 2025.

[58] R. K. Machine learning : Apprentissage supervisé ou non supervisé, mar 2018. Accessed: 2022-03-29.

[59] R. K. Machine learning : Apprentissage supervisé ou non supervisé, March 2018. Accessed: 2022-03-29.

[60] D. Karunakaran. Deep learning series 1: Intro to deep learning, oct 2019. Accessed: 2022-04-04.

[61] D. Karunakaran. Deep learning series 1: Intro to deep learning, October 2019. Accessed: 2022-04-04.

[62] T. R. Kurfess, editor. *Robotics and Automation Handbook*. CRC Press, Boca Raton, Fla., 2005.

[63] T. R. Kurfess, editor. *Robotics and Automation Handbook*. CRC Press, Boca Raton, Fla., 2005.

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

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

[66] luisquintanilla. *Tâches d'apprentissage automatique - ML.NET*, 2022. Accessed: 2022-04-03.

[67] B. Mahesh. Machine learning algorithms - a review. January 2019.

[68] B. Mahesh. Machine learning algorithms-a review. *International Journal of Scientific and Research Publications*, jan 2019.

[69] V. Maini and S. Sabri. Machine learning for humans, 2017. p. 97.

[70] V. Maini and S. Sabri. Machine learning for humans, n.d. p. 97.

[71] Matplotlib Development Team. Matplotlib: Visualization with python, 2024. Accessed: June 19, 2024.

[72] Wes McKinney et al. Data structures for statistical computing in python. *SciPy*, 445(1):51–56, 2010.

[73] A. Mohanty. Multi layer perceptron (mlp) models on real world banking data, May 2019. Accessed: 2022-04-05.

[74] H. Momtaz et al. Honey authentication: A review of the issues and challenges. *Food Control*, 132:108520, 2023.

[75] H. Momtaz et al. Honey authentication: A review of the issues and methods. *Trends in Food Science & Technology*, 139:123–135, 2023.

[76] National Honey Board. *Nutritional Composition and Health Benefits of Honey.* 2024. Accessed: February 26, 2025.

[77] National Honey Board (USA). Insights on honey purity and common adulteration methods, 2025. Accessed: February 26, 2025.

[78] Sabina Necula. Exploring the model-view-controller (mvc) architecture: A broad analysis of market and technological applications. 2024. preprint.

[79] NumPy Contributors. Numpy project, 2024. Accessed: June 4, 2024.

[80] OpenClassrooms. Design a conceptual data model. *OpenClassrooms*, 2023.

[81] OpenCV Team. Opencv about page, 2024. Accessed: June 19, 2024.

[82] Overleaf Team. Overleaf: Online collaborative latex editor, 2025. Accessed: June 9, 2025.

[83] The pandas development team. pandas: Powerful python data analysis toolkit, 2024. Accessed: June 19, 2024.

[84] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[85] M. Pirmoradi, M. Mostafaei, L. Naderloo, and H. Javadikia. Modeling honey adulteration by processing microscopic images using artificial intelligence methods. *Journal of Agricultural Science and Technology*, 24(2):365–378, 2022.

[86] P. Prasad. Machine learning process on social media. *Journal of Media Studies*, aug 2020.

[87] P. Prasad. Machine learning process on social media. August 2020.

[88] D. H. C. Pérez. A practical approach to convolutional neural networks.

[89] D. H. C. Pérez. A practical approach to convolutional neural networks, n.d. p. 70.

[90] Luis Quintanilla. Tâches d'apprentissage automatique- ml.net. Technical report, Microsoft, 2022. Accessed: 2022-04-03.

[91] Sebastián Ramírez. Fastapi: The modern, fast (high-performance), web framework for building apis with python 3.7+, 2024. Accessed: June 19, 2024.

[92] B. Rokad. Machine learning approaches and its applications, feb 2020. Accessed: 2022-04-05.

[93] B. Rokad. Machine learning approaches and its applications, February 2020. Accessed: 2022-04-05.

[94] S. Saha. A comprehensive guide to convolutional neural networks — the eli5 way, Dec. 2018. Accessed: 04/09/2022.

[95] S. Saha. A comprehensive guide to convolutional neural networks — the eli5 way, December 2018. Accessed: 2022-04-09.

[96] L. Salsabil and S. Amaria. Présenté et soutenu publiquement par, 2022. p. 63.

[97] L. Salsabil and S. Amaria. Présenté et soutenu publiquement par, n.d. p. 63.

[98] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[99] Honey Source. Quality honey: How to check the purity of your honey. 2025.

[100] Stack Overflow user "user1591666". Sequence diagram vs use case diagram. *Stack Overflow*, 2015.

[101] Ess Team. Regulatory pressures and adulteration risks in the global honey industry. 2025.

[102] Techopedia. What is a mobile application (mobile app)?, 2024. Accessed: 2025-06-08.

[103] A. Thind. A brief introduction to artificial intelligence, nov 2021. Accessed: 2022-03-24.

[104] A. Thind. A brief introduction to artificial intelligence, November 2021. Accessed: 2022-03-24.

[105] Claude Touzet. *Les réseaux de neurones artificiels, introduction au connexionnisme*. EC2, 1992.

[106] Guido van Rossum. Python: A programming language for readability and simplicity, 1996. Accessed: 2024-06-01.

[107] B. Wang. Research on pedestrian detection algorithm based on image. *Journal of Physics: Conference Series*, 1345:062023, nov 2019.

[108] World Health Organization (WHO). Health risks associated with adulterated food products, 2025. Accessed: February 26, 2025.