

République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université 08 Mai 1945 Guelma
Faculté des Mathématiques, de l'Informatique et des Sciences de la Matière
Département d'Informatique



Mémoire de Fin d'études master

Filière : informatique

Option : Système Informatique

Thème :

Un Outil d'Annotation pour le Big Data Agricole

Présenté Par : Meguellatni Aya

N	Nom	Qualité
1	Pr. Farou Brahim	Président
2	Dr. Benhamza Karima	Examineur
3	Dr. Djakhdjakha Lynda	Ecadreur

Juin 2023

REMERCIEMENTS

Tout d'abord j'exprime mes remerciements au bon DIEU de m'avoir donné la santé, le courage, et la force pour réaliser ce travail.

Je tiens à exprimer ma profonde gratitude à mon encadrante Madame DJAKHDJAKHA Lynda, pour sa patience, ses encouragements, et surtout pour ses sacrifices.

Je tiens à exprimer ma gratitude envers Monsieur Farou Brahim et à le remercier. Mille mercis à Imane Bouacida.

Je voudrais également remercier les membres du jury pour avoir consacré leur temps et leurs efforts à l'évaluation de ce travail. Leurs commentaires constructifs et leurs suggestions ont contribué à améliorer la qualité de cette thèse.

Mes sincères sentiments vont à mes parents qui ont sacrifié jusqu'à aujourd'hui et leurs encouragements tout le long de mon parcours.

DÉDICACE

Je dédie ce modest travail :

À mes chers parents, pour tout leurs sacrifices,leur amour,leur tendresse,leur soutien et surtout pour leurs prières tout au long de mes études,

*À mes chères frères "**Abd nour**" et "**Mouhamed Ridha**" pour leur apui et leur encouragement, À ma chère soeur "**Nardjess**",pour son soutien moral,*

*À mon mari "**Adel**" pour ses encouragements inspirants et son soutien moral tout au long de cette période.*

Ces dernières années, le domaine d'agriculture génère de plus en plus un volume considérable de données, notamment dans le domaine de la surveillance des maladies des cultures. Cependant, l'exploitation efficace de ces données massives nécessite des outils d'annotation précis et automatisés.

L'objectif principal de cette recherche est de concevoir une solution capable d'extraire et d'annoter automatiquement les informations pertinentes sur les maladies du blé à partir des données recueillies dans les entrepôts de grains. Pour atteindre cet objectif, nous proposons une approche basée sur les techniques du web sémantique et les algorithmes de traitement d'image.

Dans un premier temps, nous collectons les informations du domaine pour le développement d'une ontologie qui permet de structurer et représenter les connaissances. Ensuite, nous proposons un processus pour l'extraction et l'annotation des différents symptômes des maladies du blé.

En parallèle, nous concevons une interface conviviale permettant aux utilisateurs d'annoter manuellement les données non étiquetées pour améliorer la précision et la qualité des résultats. Cette interface fournira également des fonctionnalités de visualisation et de gestion des données annotées, facilitant ainsi la collaboration entre les experts agricoles et les scientifiques des données.

Les résultats de cette étude fourniront une solution pour l'annotation relatives aux données sur les maladies du blé dans les entrepôts de grains, facilitant ainsi la la visualisation et la surveillance des symptômes, et contribuant ainsi à l'amélioration de la prévention des maladies.

MOTS-CLÉS : Big Data agricole, Annotation sémantique, Ontologie de domaine, Annotation automatique, Annotation manuelle, Traitement d'image, Maladies de blé, entropot de stockage des grains.

In recent years, the field of agriculture has been generating a considerable volume of data, particularly in the area of crop disease monitoring. However, the efficient utilization of this massive data requires accurate and automated annotation tools.

The main objective of this research is to design a solution capable of automatically extracting and annotating relevant information on wheat diseases from the data collected in grain warehouses. To achieve this goal, we propose an approach based on semantic web techniques and image processing algorithms.

Firstly, we gather domain information to develop an ontology that enables the structuring and representation of knowledge. Next, we propose a process for extracting and annotating various symptoms of wheat diseases.

Simultaneously, we design a user-friendly interface that allows users to manually annotate unlabeled data to improve the accuracy and quality of results. This interface will also provide visualization and management features for annotated data, facilitating collaboration between agricultural experts and data scientists.

KEYWORDS : Agricultural Big Data, Semantic Annotation, Domain Ontology, Automatic Annotation, Manual Annotation, Image Processing, Wheat Diseases, Grain Warehouses.

Résumé		4
Abstract		
Introduction Générale		1
1 État de l'art		4
1.1 Introduction		4
1.2 Le Big Data Agricole		5
1.2.1 Définition		5
1.2.2 Les 6 « V » du Big Data agricole		5
1.3 Capteurs Agricoles utilisés dans les Entrepôts de Grains		6
1.4 Annotation de Données		10
1.4.1 Définitions		10
1.4.2 Type d'Annotation		11
1.5 Ontologie et Annotation Sémantique		12
1.6 Avantages de l'Annotation Sémantique		12
1.7 Annotation des Images en Agriculture		13
1.8 Synthèse et Conclusion		17
2 Conception		20
2.1 Introduction		20
2.2 Processus d'annotation :		20
2.3 Développement de l'ontologie		21
2.3.1 Question de compétence		22
2.3.2 Collecte d'information		23

2.3.3	Un meta-modèle pour la représentation des connaissances du domaines . . .	24
2.3.4	Conceptualisation	24
2.4	Extraction des caractéristiques	25
2.4.1	Les descripteur de couleur	25
2.4.2	Les descripteurs de texture	25
2.4.3	Les descripteurs de taille	28
2.4.4	Les descripteurs de trou	28
2.5	Extension de l'Ontologie	29
2.6	Conclusion	31
3	Implémentation	32
3.1	Introduction	32
3.2	Langages et Outils d'implémentation	32
3.2.1	Anacoda	32
3.2.2	Spyder	33
3.2.3	Python	34
3.2.4	OpenCv-Python	34
3.2.5	Os	35
3.2.6	Numpy	35
3.2.7	PyQt	35
3.3	Implémentation de l'Ontologie	36
3.4	Processus d'Annotation	37
3.4.1	Etape de Pré-Traitement	37
3.4.2	Ètape d'Extraction des Caractéristiques :	38
3.5	Un scénario d'exécution	45
3.6	Conclusion	49
	Conclusion Générale	50
	Traveaux réalisés	57

1.1 Exemple d'annotation	10
2.1 Processus d'annotation	21
2.2 Méta-Modèle représentant les concepts du domaine	24
2.3 Extension du méta-modèle	30
3.1 Anaconda Python	33
3.2 Logo spyder	34
3.3 Logo Python	34
3.4 Partie du Code de Création de l'Ontologie	36
3.5 Les Etapes de Pré-Traitement	38
3.6 Code pour l'Extraction des Caractéristiques de Couleur	39
3.7 Exemple de Résultat de la Fonction Extract_Color_Features()	40
3.8 Code pour l'extraction des caractéristiques de texture	40
3.9 Code pour l'extraction des caractéristiques de taille	41
3.10 Exemple de sortie du caractéristiques taille	42
3.11 Code détection des trous	43
3.12 Visualisation d'ontologie avant l'instanciation avec Protégé	44
3.13 Visualisation d'ontologie après l'instanciation avec Protégé	44
3.14 Visualisation du contenu d'un dataset	45
3.15 Page d'accueil du système d'annotation	45
3.16 Message d'erreur en cas de chemin invalide	46
3.17 Message d'erreur pour un chemin de dataset ne contenant pas de fichiers d'images	46
3.18 Erreur d'instanciation : Instance existante dataset	47
3.19 La création des instances en cours	47

3.20 Visualisation du dataset avec symptômes et identification des trous.	48
3.21 Détails d'annotation manuelle	48
3.22 Détails des caractéristiques de l'image	49

1.1	Les capteurs utilisés dans les entrepôts de grains	9
1.2	Récapitulatif des travaux analysés	18
2.1	Liste des concepts identifiés	24
2.2	Liste des propriétés	25
2.3	Liste des relations	25
2.4	Liste des concepts identifiés	30
2.5	Liste des propriétés	31
3.1	Liste des caractéristiques de Couleur Extraites des Images à l'aide de la Fonction de Bibliothèque OpenCV	40
3.2	Liste des caractéristiques texture extraites des images à l'aide de la fonction de bi- bliothèque OpenCV	41

Le domaine d'agriculture connaît une transformation radicale grâce à l'émergence du big data. Avec l'avènement des capteurs agricoles, les exploitants agricoles peuvent collecter une quantité massive de données provenant de leurs champs et de leurs exploitations[9]. Ces capteurs, tels que les capteurs de sol, de climatologie, d'humidité et de rendement des cultures, fournissent des informations précieuses sur les conditions de croissance, la santé des plantes et les besoins en irrigation[33], [58]. Cependant, l'impact du big data ne se limite pas aux champs agricoles. Un autre domaine où le big data agricole joue un rôle important est celui des entrepôts de stockage. Ces infrastructures jouent un rôle crucial dans le stockage et la préservation des produits agricoles, tels que les céréales, les légumes, les fruits, et bien d'autres. Les entrepôts de stockage sont équipés de capteurs spécifiques qui permettent de surveiller en temps réel les conditions de stockage, comme la température, l'humidité, la qualité de l'air et les niveaux de CO₂ [81, 41, 7, 53] and [32].

La présence de ces capteurs dans les entrepôts de stockage génère une quantité massive de données qui peuvent être exploitées pour optimiser les processus de stockage et de gestion des produits agricoles. Ces données peuvent fournir des informations précieuses sur les conditions de stockage optimales, détecter les problèmes de qualité ou de sécurité alimentaire, et aider à prendre des décisions éclairées.

Le véritable défi du Big Data réside dans la valeur qu'il peut générer [9]. Cette valeur dépend de la capacité à structurer les données massives, ce qui nous permet de les réutiliser, de les agréger de différentes manières et de les lier à d'autres données. En d'autres termes, l'innovation dans le domaine du Big Data agricole est essentielle pour créer de la valeur, et cela implique de structurer les données et de développer de nouvelles approches d'analyse [9]. Dans cette perspective, il devient crucial de développer des techniques et des outils permettant d'exploiter efficacement ces données massives provenant des capteurs agricoles dans les entrepôts de stockage. L'analyse de ces données peut fournir des informations précieuses pour améliorer la qualité des produits,

réduire les pertes post-récolte, optimiser les processus de gestion des stocks et garantir la sécurité alimentaire.

Dans ce contexte, cette étude vise à explorer l'importance du big data agricole dans les entrepôts de stockage et en particulier les données relatives aux maladies affectant les grains de blé en stock. Ces maladies posent un défi majeur tant pour les agriculteurs que pour les acteurs de la chaîne d'approvisionnement alimentaire, en raison de la nature internationale de cet aliment essentiel. Cependant, la collecte et l'analyse de données fiables et complètes dans un contexte opérationnel peut être complexe. De plus, l'intégration de différentes sources de données, telles que les relevés météorologiques, les données de surveillance et les enregistrements des entrepôts, nécessite des efforts de gestion et de traitement des données.

Pour remédier à cette problématique, nous proposons de mettre en place un outil pour l'annotation de ces données basé sur les ontologies. Les ontologies, en tant que structures de connaissances, offrent la possibilité de représenter de manière sémantique les concepts, les relations et les propriétés propres à un domaine particulier. L'annotation est une information qui peut être associée à différentes entités telles qu'un ensemble de documents, un document, un passage, une phrase, un terme, un mot, une image, etc. Son objectif principal est de donner généralement un sens à l'une de ces entités [52]. L'annotation sémantique implique une représentation formelle du contenu en utilisant des concepts, des relations et des instances, éventuellement dans une ontologie liée à la ressource documentaire source[2]. L'annotation sémantique a déjà été utilisée dans diverses applications, telles que la classification, l'extraction d'information, le résumé automatique et l'interopérabilité. Elle offre une approche précieuse pour l'identification des symptômes de maladies dans le domaine médical [1], [63]. Elle permettra de lier des informations sémantiques précises aux données massives, ce qui facilitera leur compréhension, leur analyse et leur utilisation.

Notre mémoire se concentre sur le développement d'un outil d'annotation sémantique de big data visant à faciliter la détection des symptômes des maladies des blés dans les entrepôts de grain. En combinant des techniques du Web sémantique et de traitement d'image, notre outil d'annotation sémantique sera en mesure d'extraire des connaissances spécifiques liées aux maladies des blés. Il identifiera les symptômes caractéristiques, ainsi que d'autres informations pertinentes pour une gestion optimale de ces maladies.

Ce mémoire sera structuré comme suit :

- Le premier chapitre présente une vision d'ensemble complète de la définition du big data

agricole, de ses caractéristiques ainsi que des capteurs utilisés dans les entrepôts de stockage. Il aborde également la notion d'annotation des données et les avancées réalisées dans ce domaine, notamment en ce qui concerne l'annotation des images en agriculture. Cette partie constitue une introduction essentielle pour appréhender les enjeux et les opportunités liés à l'utilisation des données agricoles, ainsi que pour comprendre comment l'annotation des images peut contribuer à l'amélioration des processus agricoles.

- Le deuxième chapitre présentera en détail l'approche adoptée pour le développement de l'outil d'annotation sémantique. Il décrira les différentes étapes du processus d'annotation, avec une focalisation particulière sur le développement de l'ontologie. Les choix de conception ainsi que les fonctionnalités spécifiques pour l'annotation des données liées aux maladies des blés dans les entrepôts de grain seront expliqués de manière approfondie.
- Le troisième chapitre présentera en détail l'implémentation de l'outil d'annotation, en mettant l'accent sur les outils et langages utilisés pour sa réalisation. De plus, il illustrera l'utilisation de l'outil à travers des scénarios concrets, démontrant ainsi son utilité dans ce domaine spécifique des maladies des blés dans les entrepôts de grain.
- Enfin, la conclusion viendra clore ce mémoire en récapitulant les principales contributions, en soulignant les avantages et les limites de l'outil d'annotation développé, et en proposant des pistes pour des travaux futurs dans ce domaine passionnant de l'agriculture et de l'annotation des données.

1.1 Introduction

Le domaine de l'agriculture génère un volume considérable de données grâce à l'utilisation croissante de capteurs agricoles. Ces données, regroupées sous le terme de "big data agricole", offrent un potentiel significatif pour améliorer la productivité et la durabilité de l'agriculture. Cependant, l'analyse efficace des données relatives aux entrepôts de stockage des grains représente un défi majeur, car il est crucial de garantir la qualité des grains et de minimiser les pertes.

Dans ce contexte, l'annotation des données agricoles joue un rôle essentiel. Elle permet de catégoriser, d'organiser et d'analyser les données de manière plus efficace. L'annotation des images revêt une importance particulière en agriculture, car elle facilite la détection des cultures, la surveillance des parasites et des maladies, ainsi que l'évaluation de la qualité des produits agricoles.

Dans ce chapitre, nous aborderons la notion de "big data agricole" en examinant ses caractéristiques clés. Ensuite, nous nous pencherons sur les sources de big data agricole relatives aux entrepôts de stockage des grains. Nous aborderons ensuite la notion d'annotation des données et les avantages de l'annotation sémantique. Enfin, nous explorerons les avancées récentes dans le domaine de l'annotation des images en agriculture, mettant en évidence son potentiel et ses applications concrètes.

1.2 Le Big Data Agricole

1.2.1 Définition

Les Big Data agricoles désignent la collecte, le stockage et l'analyse massifs de données générées dans le domaine de l'agriculture. Ces données proviennent de diverses sources telles que les capteurs, les drones, les satellites, les réseaux sociaux et les machines agricoles. Elles offrent des perspectives uniques pour améliorer la productivité, l'efficacité et la durabilité de l'agriculture [8].

1.2.2 Les 6 « V » du Big Data agricole

Le concept des fameux "V du Big Data" est souvent utilisé pour décrire les caractéristiques essentielles des données volumineuses (Big Data) dans le domaine de l'agriculture [9]. Ces "V" représentent les principaux défis et caractéristiques associés aux données agricoles massives. Les trois "V" les plus couramment mentionnés sont les suivants :

- Volume : Le premier "V" fait référence au volume des données agricoles. Avec les progrès technologiques tels que l'utilisation de capteurs, de drones, de satellites et de machines agricoles connectées, une énorme quantité de données est collectée dans l'agriculture moderne. Ces données proviennent de diverses sources telles que les machines agricoles, les capteurs environnementaux, les drones agricoles, les systèmes de gestion agricole, etc.
- Vitesse : Le deuxième "V" concerne la vitesse ou la rapidité à laquelle les données agricoles sont générées, collectées et traitées. Dans l'agriculture de précision et la gestion des cultures, il est essentiel de prendre des décisions rapides basées sur les données en temps réel. Les capteurs, les drones et d'autres dispositifs génèrent continuellement des données agricoles, ce qui nécessite des capacités de traitement et d'analyse en temps réel pour une prise de décision efficace.
- Variété : Le troisième "V" représente la variété des données agricoles. Les données agricoles proviennent de sources diverses et variées, telles que les capteurs de sol, les images satellites, les données météorologiques, les données d'irrigation, les données sur les cultures, etc. Ces données peuvent être structurées (bases de données traditionnelles), semi-structurées (formats JSON, XML) ou non structurées (images, vidéos). La gestion et l'intégration de cette variété de données pour en extraire des informations significatives sont des défis importants. Au-delà des données elles-mêmes, la diversité des acteurs qui les produisent et les manipulent constitue un défi en soi dans le domaine agricole. Ces acteurs, tels que les agronomes, les généticiens, les biologistes, les statisticiens, les industriels, les distributeurs, possèdent naturellement des vocabulaires, des sémantiques et des méthodes de production de don-

nées très hétérogènes. Cette grande diversité rend complexe l'intégration des données, ce qui nécessite de nouvelles approches pour gérer et analyser les données en prenant en compte non seulement les dimensions spatiales et temporelles, mais aussi les aspects organisationnels tels que les filières, les circuits courts et les normes sanitaires.

Il convient de noter que certains experts ajoutent d'autres "V" pour décrire davantage les caractéristiques du Big Data agricole, tels que [9] :

- Véracité : Ce "V" concerne la qualité et la fiabilité des données agricoles. Les données provenant de différentes sources peuvent varier en termes de précision, de cohérence et de qualité. Il est essentiel de disposer de mécanismes pour évaluer et garantir la véracité des données agricoles utilisées pour la prise de décision.
- Valeur : Ce "V" représente la capacité à extraire de la valeur à partir des données agricoles. La collecte et le stockage massifs de données ne sont utiles que s'ils conduisent à des informations exploitables pour les agriculteurs et les décideurs. L'analyse des données agricoles doit permettre de générer des connaissances précieuses, d'améliorer les rendements, de prendre des décisions éclairées et d'optimiser les processus agricoles.

1.3 Capteurs Agricoles utilisés dans les Entrepôts de Grains

Afin d'identifier les différents capteurs agricole utilisés dans les entrepôts de Grains, nous avons basé sur la méthodologie Prisma¹. En utilisant les mots-clés "Grain Warehouse IoT" dans la base de données Mendeley, nous avons obtenu un total de 14 papiers. Parmi ceux-ci, nous avons sélectionné 13 articles provenant de revues et de conférences scientifiques. Nous avons ensuite restreint notre sélection à des papiers publiés entre 2018 et 2023, ce qui nous a permis d'obtenir 12 articles pertinents. Dans le but de faciliter l'accès à l'information, nous avons également privilégié les papiers en accès libre (open access). Après cette étape, nous avons retenu 12 papiers accessibles gratuitement. En procédant à une lecture complète et détaillée de ces papiers, nous avons identifié 8 articles pertinents qui correspondaient spécifiquement à notre objectif :

Le modèle proposé dans [62] s'est avéré capable d'acquérir avec précision des mesures de température et d'humidité. Une alarme est déclenchée et envoyée à la personne responsable ou à l'observateur si la valeur reçue dépasse la limite prédéfinie. Ainsi, il est possible de surveiller facilement les articles stockés dans l'entrepôt et de prévenir les dommages aux grains, ce qui permet de maintenir leur qualité et d'augmenter leur durée de vie.

1. <http://www.prisma-statement.org/PRISMAStatement/?AspxAutoDetectCookieSupport=1>

Les auteurs dans l'article [37] proposent un système intelligent basé sur l'IoT pour surveiller les grains dans les entrepôts. Le système utilise des capteurs de température, d'humidité et de CO2 pour observer les paramètres de l'entrepôt. La détection des mouvements des rongeurs est réalisée grâce à la technique de détection des contours et les résultats sont affichés à l'aide d'un projecteur. Un dispositif de mesure de la force de morsure (BFM) est développé pour évaluer la force de morsure et est interfacé avec un microcontrôleur et programmé en LabVIEW. Lorsqu'un mouvement de rongeur est détecté, un capteur infrarouge envoie un message au module GSM, qui envoie ensuite un SMS au client pour l'informer de la présence de rongeurs. Le client peut observer les rongeurs via une caméra IP et prendre des mesures pour les éloigner.

L'article [69] décrit un système intelligent avancé conçu pour surveiller et contrôler l'état des grains dans de grands entrepôts. Le système utilise la technologie IoT et la technologie cloud pour offrir des performances, une évolutivité, une portabilité et une sécurité de transmission des données optimales. Les dispositifs matériels utilisés dans l'expérience comprennent le capteur DHT22, le capteur CDM7160 et un capteur de CO2. Ces capteurs sont utilisés pour observer les paramètres de l'entrepôt tels que la température, l'humidité relative et la concentration de dioxyde de carbone.

Dans [16], un prototype IoT est présenté pour la détection des intrusions dans les entrepôts et le suivi des grains. Le système utilise des capteurs PIR pour détecter les intrusions et des capteurs RFID pour suivre individuellement chaque sac de grains. Lorsqu'une intrusion est détectée, une alerte est envoyée à la direction via un module GSM. Les données de suivi des grains sont transmises à une base de données cloud via des modules de communication sans fil. Ce prototype vise à améliorer la sécurité des entrepôts et à faciliter le suivi précis des grains au sein de l'agence de réserve alimentaire.

Les auteurs de [32] proposent un système de surveillance basé sur des capteurs IoT pour surveiller en temps réel les conditions de stockage des céréales dans les entrepôts. Le système utilise des capteurs pour mesurer des paramètres tels que la température, l'humidité et la qualité de l'air, qui sont des facteurs critiques pour la préservation des céréales. Il collecte les données des capteurs et les transmet à une plateforme de surveillance centralisée. Les données sont ensuite analysées et traitées pour détecter les écarts par rapport aux conditions de stockage optimales. Des alertes sont générées en cas de conditions anormales, ce qui permet aux gestionnaires d'entrepôt de prendre des mesures correctives rapidement. Les auteurs mettent en avant les avantages de ce système automatisé de surveillance des céréales, tels que la réduction des pertes de stock

dues à des conditions de stockage inadéquates et la possibilité d'une intervention rapide en cas de problèmes détectés. De plus, le système permet une surveillance continue sans dépendre de l'intervention humaine, ce qui améliore l'efficacité et la précision de la surveillance.

Les auteurs de [53] proposent un système de surveillance d'entrepôt intelligent basé sur l'IoT. Le réseau de capteurs comprend des capteurs de vibration, d'humidité, de température et d'incendie. Le contrôleur Raspberry Pi adopte la technologie de l'IoT pour transmettre les messages. En fonction des données des capteurs, les informations appropriées sont capturées et traitées en fonction des limites définies dans le logiciel, puis des informations sont envoyées en temps opportun aux responsables concerné d'entrepôt via SMS pour des mesures de modération et des actions correctives en cas de conditions atmosphériques à l'intérieur de l'entrepôt. Le système développé présente de nombreux avantages par rapport au modèle traditionnel, notamment en termes de stockage des données d'entrepôt dans le cloud.

Le système de surveillance des entrepôts de stockage de grains de [38] utilise un capteur de température et d'humidité DHT11. Ce capteur est intégré à un dispositif de surveillance qui collecte les données de température et d'humidité dans les différentes étapes de la chaîne de production et de stockage des grains, notamment dans les fermes, les usines de transformation, les entrepôts de stockage et les supermarchés. Le capteur DHT11 mesure à la fois la température et l'humidité de l'environnement dans lequel il est placé. Ces données sont ensuite utilisées pour surveiller les conditions de stockage des grains et prendre des mesures appropriées pour maintenir des niveaux optimaux de température et d'humidité.

L'article [81] présente la conception et l'implémentation d'un système de surveillance. Le système utilise la technologie du capteur de température et d'humidité DHT11 ainsi que la technologie du transmetteur sans fil nRF24L01 pour former un système de surveillance sans fil des capteurs de température et d'humidité basé sur un réseau de capteurs sans fil (WSN) afin d'assurer la surveillance des entrepôts de grains. La partie émettrice est basée sur STC89C52, et la puce de conversion de température et d'humidité DHT11 est utilisée pour collecter la température en temps réel. La température collectée est transmise sans fil à la partie réceptrice par le biais de la puce de fréquence radio nRF24L01. Enfin, elle est affichée sur le tube numérique à l'extrémité réceptrice.

L'analyse de travaux existants (Table 1.1) permet de montrer les types de BIG Data agricole qui peuvent être collectés dans les entrepôts de stockage :

Le papier	Objectif	Les capteurs utilisés dans les entrepôts de grain
[32]	Un système de surveillance des céréales	Capteur de température, d'humidité et de qualité de l'air
[53]	Un système de surveillance d'entrepôts	Capteurs de vibration, d'humidité, de température et d'incendie
[81]	Un système de surveillance	Capteur de température et d'humidité DHT11
[16]	Un prototype IoT pour la détection des intrusions dans les entrepôts et un modèle de suivi des grains (e-périmètre)	Capteurs PIR et capteurs RFID
[69]	Un système intelligent pour surveiller et contrôler les conditions des grains dans de grands dépôts.	Capteur DHT22 de température, CDM7160 d'humidité et le capteur de CO2
[37]	Un Système de surveillance et d'évitement des rongeurs basé sur l'IoT dans les silos de stockage agricole	Capteur PIR /Capteur IR SENSORS (image)
[38]	Un système de surveillance des entrepôts de stockage de grains	Capteur DHT11 de température et humidité
[62]	Une méthode pour la surveillance de qualité des grains et fruits dans les entrepôts	Capteur DHT11 de température et humidité

TABLE 1.1 – Les capteurs utilisés dans les entrepôts de grains

- Données de surveillance environnementale : Les entrepôts de stockage de grains utilisent des capteurs pour surveiller les conditions environnementales telles que la température, l'humidité, etc. Ces données permettent de s'assurer que les grains sont conservés dans des conditions optimales pour éviter la détérioration, la croissance de moisissures ou l'infestation d'insectes.
- Données de qualité des grains : Les entrepôts de stockage collectent des données sur la qualité des grains, telles que la teneur en humidité, la teneur en protéines, le taux de germination, etc. Ces données aident à évaluer la qualité des grains stockés, à prendre des décisions sur le moment optimal pour la vente ou l'utilisation des grains, et à éviter les pertes dues à la détérioration.
- Données de surveillance des rongeurs : Les capteurs installés dans les entrepôts de stockage peuvent détecter la présence de rongeurs. Ces données permettent de prendre des mesures préventives ou curatives rapidement pour minimiser les pertes et garantir la qualité des grains stockés.

En utilisant ces types de BIG Data agricole dans les entrepôts de stockage de grains, il est possible d'améliorer la gestion, la qualité et la traçabilité des grains, réduisant ainsi les pertes et assurant un approvisionnement alimentaire plus sûr et plus efficace.

Cependant, malgré le potentiel prometteur des données d'images, leur utilisation dans les travaux existants est encore limitée. Des recherches supplémentaires sont nécessaires pour développer

des méthodes d'analyse des données de type images adaptés à la détection des maladies des grains dans les entrepôts de stockage.

1.4 Annotation de Données

1.4.1 Définitions

Selon Larousse², l'annotation est l'action de faire des remarques sur un texte pour l'expliquer ou le commenter. En science de données, l'annotation de données est le processus consistant à attribuer des étiquettes ou des marques à des données pour aider les algorithmes d'apprentissage automatique à comprendre et à classer les informations qu'ils traitent³. Selon [79], l'annotation est le processus qui consiste à ajouter des métadonnées ou des informations sémantiques à un objet, comme une image ou un document, pour améliorer l'extraction et la récupération des informations.

Les annotations peuvent inclure des mots-clés, des descriptions, des classifications ou des catégories, des notes, des balises, des liens et d'autres types de données qui aident à décrire et à organiser l'objet annoté. La figure 1.1 illustre un exemple d'annotation⁴.

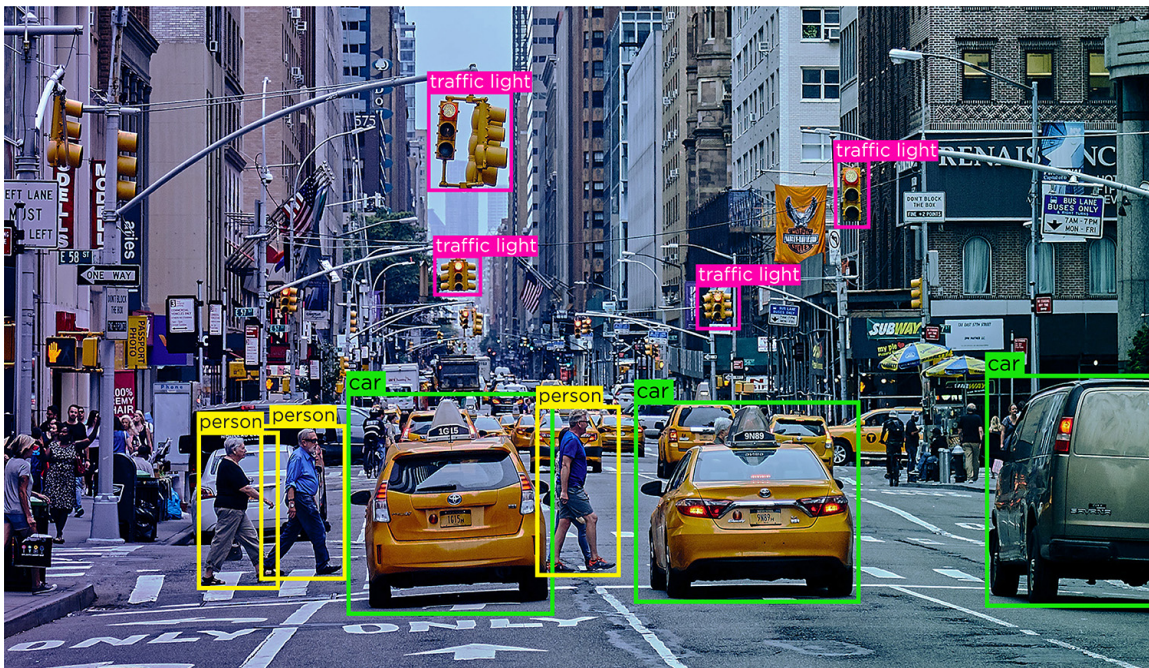


FIGURE 1.1 – Exemple d'annotation

2. <https://www.larousse.fr/dictionnaires/francais/annotation/3676>

3. <https://fr.shaip.com/blog/the-a-to-z-of-data-annotation/>

4. <https://fr.shaip.com/blog/the-a-to-z-of-data-annotation/>

1.4.2 Type d'Annotation

Annotation Sémantique

L'annotation sémantique consiste à ajouter des informations sémantiques à un contenu, tel que le sens d'un mot, la signification d'une phrase, ou le contexte d'un texte. Elle est utilisée pour aider les machines à comprendre la signification du contenu, et faciliter la recherche, l'indexation et l'analyse de données. Selon [77], l'annotation sémantique peut être effectuée en utilisant différentes techniques telles que l'analyse syntaxique, l'analyse sémantique, la représentation des connaissances et l'alignement d'ontologies. Elle peut être appliquée à différents types de données, y compris les textes, les images et les vidéos.

Annotation d'Objet

Elle consiste à marquer des régions spécifiques d'une image ou d'une vidéo avec des étiquettes ou des tags qui indiquent le type ou la classe d'objet présent dans cette région [82]. Cette forme d'annotation est couramment utilisée dans le domaine de la vision par ordinateur et de l'apprentissage automatique pour entraîner des modèles à reconnaître et à classer des objets dans des images et des vidéos.

Annotation Temporelle

L'annotation temporelle est le processus d'ajout de marques de temps (c'est-à-dire des informations temporelles) à des événements, des actions ou des entités dans un texte, une vidéo, un enregistrement audio ou tout autre type de données temporelles [60]. L'annotation temporelle permet de localiser les événements dans le temps, de mesurer leur durée et de décrire les relations temporelles entre eux.

Annotation Spatiale

L'annotation spatiale consiste à ajouter des informations supplémentaires aux entités géographiques, telles que des attributs, des métadonnées, des relations topologiques ou des liens avec des sources de données externes [45]. L'objectif de l'annotation spatiale est de fournir des informations plus riches et plus utiles sur les entités géographiques, permettant ainsi des analyses plus avancées et une meilleure prise de décision. En d'autres termes, l'annotation spatiale permet d'enrichir les données géographiques en ajoutant des informations contextuelles et en les reliant à d'autres sources de données. Cette définition met également en évidence les avantages de l'annotation spatiale pour l'analyse et la prise de décision.

Annotation de Sentiments

L'annotation de sentiment est la tâche de déterminer si une opinion exprimée dans un texte est positive, négative ou neutre, ou de déterminer l'intensité de cette opinion. L'annotation de sentiment peut être effectuée à différents niveaux, allant du niveau de la phrase au niveau du document, et peut impliquer l'utilisation de différentes techniques d'apprentissage automatique pour identifier les mots et les expressions qui indiquent un sentiment [55].

1.5 Ontologie et Annotation Sémantique

Les ontologies sont un élément clé de l'annotation sémantique. Une ontologie est une représentation formelle et structurée des concepts et de leurs relations dans un domaine spécifique [48]. Elle fournit un vocabulaire commun et une structure pour décrire les concepts et les relations entre eux. L'utilisation d'une ontologie peut aider à améliorer la précision et la cohérence de l'annotation sémantique, en fournissant un cadre de référence commun pour l'interprétation des données[6]. Les ontologies peuvent être utilisées pour annoter des textes, des images, des vidéos et d'autres types de données[3].

Les ontologies sont utilisées dans l'annotation sémantique de plusieurs manières, notamment :

- Comme cadre de référence : les ontologies fournissent un vocabulaire commun et une structure pour décrire les concepts et les relations entre eux, ce qui permet d'assurer la cohérence et la précision de l'annotation sémantique [47].
- Pour l'alignement de l'annotation : les ontologies peuvent être utilisées pour aligner les annotations sémantiques avec les concepts de l'ontologie, ce qui permet de créer des liens entre les données et de faciliter leur recherche et leur analyse [48].
- Pour l'extension de l'ontologie : l'annotation sémantique peut être utilisée pour enrichir l'ontologie en ajoutant de nouveaux concepts ou en améliorant les relations entre les concepts existants [5].

1.6 Avantages de l'Annotation Sémantique

L'annotation sémantique présente plusieurs avantages par rapport à d'autres méthodes d'annotation :

- Précision : l'annotation sémantique permet une annotation plus précise des données, car elle relie le contenu d'un texte, d'une image ou d'un objet à des entités dans une ontologie,

ce qui permet de créer des liens entre les données et de faciliter leur recherche et leur analyse [6].

- Cohérence : l'utilisation d'une ontologie fournit un vocabulaire commun et une structure pour décrire les concepts et les relations entre eux, ce qui permet d'assurer la cohérence et la précision de l'annotation sémantique⁵.
- Extensibilité : l'annotation sémantique peut être utilisée pour enrichir l'ontologie en ajoutant de nouveaux concepts ou en améliorant les relations entre les concepts existants [1].
- Automatisation : l'annotation sémantique peut être automatisée à l'aide de techniques d'apprentissage automatique, ce qui permet de traiter de grandes quantités de données rapidement et efficacement[23].
- Recherche et analyse : les annotations sémantiques permettent une recherche et une analyse plus précises des données, car elles permettent de créer des liens entre les données et les concepts de l'ontologie[28].

1.7 Annotation des Images en Agriculture

Pour réaliser une recherche exhaustive sur l'annotation d'images en agriculture, nous avons basé sur la méthodologie Prisma⁶. En utilisant les mots-clés "image annotation agriculture" dans la base de données Mendelej, nous avons obtenu un total de 134 papiers. Parmi ceux-ci, nous avons sélectionné 130 articles provenant de revues et de conférences scientifiques. Nous avons ensuite restreint notre sélection à des papiers publiés entre 2019 et 2023, ce qui nous a permis d'obtenir 99 articles pertinents. Dans le but de faciliter l'accès à l'information, nous avons également privilégié les papiers en accès libre (open access). Après cette étape, nous avons retenu 42 papiers accessibles gratuitement. En procédant à une lecture complète et détaillée de ces 42 papiers, nous avons identifié 11 articles pertinents qui correspondaient spécifiquement à notre domaine d'intérêt :

Les auteurs de [70] proposent un ensemble de données spécifique qui a été créé pour soutenir la vision informatique robotique dans le domaine de l'agriculture. L'ensemble de données est composé d'images annotées de cultures alimentaires et de mauvaises herbes. Les auteurs soulignent l'importance de disposer d'un ensemble de données précis et étiqueté pour former et évaluer les modèles d'apprentissage automatique utilisés dans les systèmes de vision des robots agricoles. Ces modèles permettent aux robots d'identifier et de différencier les cultures alimentaires

5. <https://lipn.univ-paris13.fr/lannotation-semantique-et-exploration-textuelle/>

6. <http://www.prisma-statement.org/PRISMAStatement/?AspxAutoDetectCookieSupport=1>

des mauvaises herbes, ce qui est crucial pour les tâches de contrôle et de gestion des cultures. L'ensemble de données décrit dans l'article comprend un large éventail d'images couvrant différentes variétés de cultures alimentaires et de mauvaises herbes. Chaque image est annotée avec des étiquettes précises indiquant la classe de la culture ou de la mauvaise herbe qu'elle représente. Les auteurs expliquent également les protocoles et les méthodes utilisés pour l'acquisition des images, l'annotation des étiquettes et la validation de la qualité de l'ensemble de données.

Le papier [50] présente une approche avancée d'annotation automatique d'images pour l'agriculture. L'approche utilise des tâches d'annotation répétitives pour annoter automatiquement un objet. Le jeu de données de l'étude comprend des fruits de palmier à huile (FFB) et une variété de fruits, avec un grand nombre de données. Le modèle YOLOv5, une approche d'apprentissage en profondeur, est choisi pour annoter automatiquement les images en utilisant la technique des tâches d'annotation répétitives. La méthode développée a été testée sur un grand jeu de données pour évaluer ses performances et sa précision en matière d'annotation. Les résultats montrent que le réseau entraîné est capable de classifier correctement un objet dans une image. Deux autres versions de YOLO, YOLOv3 et YOLOv4, ont également été entraînées et évaluées sur le même jeu de données, afin de comparer les résultats avec ceux obtenus par l'approche proposée. Les résultats comparatifs démontrent l'efficacité et la supériorité de la méthode proposée pour la catégorisation des fruits.

Dans l'article [54], Eden Library, une nouvelle plateforme de données agricoles ouvertes, est présentée. Le schéma complet des métadonnées, le processus d'annotation et les principales fonctionnalités de la plateforme sont décrits. Grâce à la méthodologie agile suivie pour la construction de la plateforme, de nouvelles fonctionnalités et ensembles de données peuvent être ajoutées pour répondre à la demande croissante d'applications de vision sur le terrain. De plus, des améliorations de la méthodologie d'annotation, telles que l'utilisation de l'annotation semi-supervisée, seront explorées afin d'optimiser l'efficacité de l'ingestion des données. Par conséquent, davantage d'ensembles de données de qualité seront disponibles plus rapidement et la recherche communautaire dans les domaines de l'intelligence artificielle, de la vision par ordinateur et de l'agriculture de précision pourra améliorer les performances de ses technologies actuelles.

[87] présente CropDeep, un nouvel ensemble de données spécifique au domaine de l'agriculture de précision. CropDeep vise à fournir une référence pour la construction de modèles de classification et de détection basés sur l'apprentissage en profondeur, en prenant en compte les caractéristiques réalistes des tâches agricoles dans les serres. Les avantages de CropDeep par rapport

à d'autres ensembles de données de vision comprennent une collecte d'images dans des serres réelles, des échantillons d'annotations plus représentatifs et abondants, ainsi que des images présentant des changements périodiques et des caractéristiques de similarité. Le papier souligne également l'importance d'améliorer les performances des frameworks de classification et de détection basés sur l'apprentissage en profondeur pour des applications agricoles, et propose d'explorer des améliorations telles que l'annotation semi-supervisée.

Dans l'étude de [11], les auteurs ont proposé une approche basée sur la méthode ATSS DL pour détecter les fruits de pomme en utilisant une stratégie d'étiquetage des points centraux. La méthode a été comparée à d'autres réseaux basés sur l'apprentissage en profondeur de pointe, notamment RetinaNet, Libra R-CNN, Cascade R-CNN, Faster R-CNN, FSAF et HRNet. Les auteurs ont évalué différentes conditions d'occlusion et de corruption du bruit dans différents ensembles d'images. L'approche basée sur ATSS a surpassé toutes les autres méthodes, atteignant une précision moyenne maximale de 0,946 avec une taille de boîte englobante de 160×160 pixels. Lors de la recréation des conditions d'adversité au moment de l'acquisition des données, l'approche a fourni une réponse robuste à la plupart des données corrompues, à l'exception des conditions météorologiques de neige, de gel et de brouillard. Des études supplémentaires sont suggérées avec d'autres variétés de fruits dans lesquelles la couleur joue un rôle important pour les différencier des feuilles. Des attributs supplémentaires des fruits tels que la forme, le poids et la couleur sont également des informations importantes pour des recherches futures de ce travail.

Le papier [12] présente une étude sur l'utilisation de l'apprentissage actif avec MaskAL (Mask Active Learning) pour réduire l'effort d'annotation lors de l'entraînement d'un modèle Mask R-CNN sur un ensemble de données de brocoli contenant des classes visuellement similaires. Les chercheurs ont comparé les performances de MaskAL avec l'échantillonnage aléatoire. Les résultats de l'étude montrent que MaskAL a surpassé l'échantillonnage aléatoire, en obtenant de meilleures performances avec un nombre beaucoup moins important d'échantillons annotés. Par exemple, après avoir annoté seulement 900 images avec MaskAL, les performances étaient équivalentes à celles obtenues après avoir annoté 2300 images avec l'échantillonnage aléatoire. Cela signifie que l'utilisation de MaskAL aurait permis d'économiser 1400 annotations. Les chercheurs concluent que l'utilisation de MaskAL permet de réduire considérablement l'effort d'annotation nécessaire pour entraîner un modèle Mask R-CNN sur un ensemble de données de brocoli avec des classes visuellement similaires. De plus, ils suggèrent que MaskAL pourrait également être utilisé pour sélectionner des classes inconnues dans l'apprentissage à jeu ouvert et comme estimateur d'incertitude dans les cadres robotiques probabilistes.

Le papier [67] présente une nouvelle méthodologie pour générer des images fluorescentes de plantes malades avec une annotation automatique des lésions. Les chercheurs ont utilisé un modèle U-Net pour segmenter les lésions de maladies sur ces images. Ils ont démontré que ce modèle peut être efficacement entraîné en utilisant uniquement un ensemble de données synthétiques générées. La création et l'utilisation de telles données synthétiques constituent une technique puissante pour faciliter l'application des méthodes d'apprentissage profond dans la protection précise des cultures. De plus, la méthode de génération d'images fluorescentes synthétiques présentée dans l'étude contribue à améliorer la capacité de généralisation des modèles d'apprentissage profond dans ce domaine.

Le papier [40] introduit un ensemble de données comprenant 2 801 images de cultures maraîchères, notamment de maïs, de haricots et de poireaux, à un stade précoce de développement. Les annotations fournies comprennent des boîtes englobant les cultures d'intérêt et des graphes en étoile représentant les structures des cultures. Les images ont été capturées dans diverses parcelles en France, offrant une variété de conditions de sol, d'infestation de mauvaises herbes et de stades de croissance. Cet ensemble de données est utile pour les applications de binage de précision et de surveillance des cultures sur le terrain basées sur l'imagerie proximale.

Le CoFly-WeedDB [1] est une base de données contenant 201 images RVB prises dans un champ de coton en Grèce pendant les premiers stades de croissance des plantes. Les images ont été collectées à partir d'un UAV équipé d'une caméra, avec des annotations fournissant les limites précises de 3 types de mauvaises herbes courantes. Cette base de données peut être utilisée pour développer des méthodes basées sur l'IA pour la segmentation et la classification automatiques des mauvaises herbes, soit seule, soit en combinaison avec d'autres ensembles de données.

[64] met en évidence l'utilisation de réseaux de neurones convolutifs (CNN) pour estimer la fragmentation des grains de maïs dans les ensilages à partir d'images. Les deux approches de délimitation par boîte englobante et de segmentation d'instance ont été utilisées avec succès pour estimer la fragmentation de manière robuste. L'entraînement des modèles a été réalisé en utilisant des images provenant de différentes saisons de récolte grâce à l'apprentissage par transfert. L'évaluation de l'estimation de la fragmentation a été réalisée à l'aide de mesures de vision par ordinateur et en comparant les prédictions des modèles avec un score de traitement des grains. Les résultats ont démontré une forte corrélation entre les prédictions des modèles et le score de traitement des grains utilisé dans l'industrie. De plus, l'étude a également montré la capacité pro-

metteuse d'estimer la fragmentation des grains même dans des images où les grains et la tige ne sont pas séparés, ce qui pourrait avoir un impact significatif sur le temps de mesure. Cela pourrait potentiellement accélérer le processus d'évaluation de la fragmentation des grains de maïs dans les ensilages.

Dans [76], les auteurs décrivent leur ensemble de données, TomatOD, et leurs plans pour l'enrichir davantage en ajoutant des données provenant de différentes serres de culture sans sol. L'objectif est de permettre aux modèles entraînés sur cet ensemble de données de s'intégrer facilement aux systèmes opérationnels. Les auteurs soulignent également la nécessité d'améliorer les annotations en faisant appel à davantage d'experts, afin de pouvoir comparer statistiquement leurs annotations. Les auteurs mentionnent que les détecteurs de pointe actuels ne conviennent pas à un usage en production dans un scénario similaire à leur ensemble de données. Les détecteurs à deux étapes approchent d'un niveau de précision de détection acceptable, mais manquent de capacité d'inférence en temps réel. D'autre part, les détecteurs "rapides" à tir unique ne fonctionnent pas bien avec la distribution des tailles d'objets spécifiques de TomatOD, entraînant une faible précision de détection. Les auteurs soulignent que le principal défi réside dans la petite taille des objets présents dans leurs données. Ils suggèrent que la recherche en détection d'objets devrait se concentrer sur l'ajustement des échelles des modèles pour mieux correspondre à la distribution des tailles d'objets de l'ensemble de données, afin de développer des modèles offrant une précision acceptable tout en maintenant des performances d'inférence quasi temps réel.

1.8 Synthèse et Conclusion

Le papier	Titre	Type de culture annotée
[70]	Dataset of annotated food crops and weed images for robotic computer vision control	Ensemble de données d'images annotées de cultures alimentaires et de mauvaises herbes
[50]	Enhancing Image Annotation Technique of Fruit Classification Using a Deep Learning Approach	Annotation automatique d'images pour les fruits de palmier à huile et autres fruits
[87]	CropDeep : The crop vision dataset for deep-learning-based classification and detection in precision agriculture	CropDeep est un ensemble de données de vision comprenant une collection d'images dans des serres réelles
[11]	ATSS deep learning-based approach to detect apple fruits	L'approche basée sur la méthode ATSS DL utilisée pour détecter les fruits de pomme
[12]	Active learning with MaskAL reduces annotation effort for training Mask R-CNN on a broccoli dataset with visually similar classes	L'apprentissage actif avec MaskAL est utilisé pour réduire l'effort d'annotation dans l'entraînement d'un modèle Mask R-CNN pour le brocoli
[67]	Plant disease symptom segmentation in chlorophyll fluorescence imaging with a synthetic dataset	Générer des images fluorescentes de plantes malades avec une annotation automatique des lésions
[40]	An annotated image dataset of vegetable crops at an early stage of growth for proximal sensing applications	Ensemble de données comprenant 2 801 images de cultures maraîchères, notamment de maïs, de haricots et de poireaux, à un stade précoce de développement
[64]	Maize silage kernel fragment estimation using deep learning-based object recognition in non-separated kernel/stover rgb images	Estimer la fragmentation des grains de maïs dans les ensilages à partir d'images
[76]	TOMATOD : Evaluation of object detection algorithms on an real-word tomato dataset	TomatOD, Ensemble de données de différentes serres de culture sans sol
[39]	CoFly-WeedDB : A UAV image dataset for weed detection and species identification	Un ensemble de données d'images de drones pour la détection des mauvaises herbes et l'identification des espèces
[54]	Eden library : A long-term database for storing agricultural multi-sensor datasets from uav and proximal platforms	Eden Library, une nouvelle plateforme de données agricoles ouvertes

TABLE 1.2 – Récapitulatif des travaux analysés

Les articles mentionnés précédemment proposent des contributions intéressantes dans le domaine de l'annotation d'images pour l'agriculture intelligente. Cependant, il est important de souligner que notre recherche n'a pas abouti à la découverte de travaux spécifiquement dédiés à l'annotation des images de blé ou des grains en stockage. Cette absence peut être considérée comme une limite dans le cadre de l'analyse des grains en stockage.

Dans le chapitre suivant, nous présenterons un outil que nous avons développé spécifiquement pour l'annotation des images de blé en stock. Cet outil vise à combler la lacune identifiée dans la littérature en fournissant une solution pratique et efficace pour l'annotation des images de grains en stockage. Nous décrirons en détail les fonctionnalités de cet outil, son interface convi-

viale et sa capacité à faciliter le processus d'annotation en fournissant des fonctionnalités avancées telles que la détection automatique des grains, la segmentation précise et la détection des symptômes. En intégrant cet outil dans notre analyse des grains en stockage, nous espérons contribuer à l'avancement de la recherche dans ce domaine et fournir aux professionnels de l'agriculture un moyen précis et efficace d'annoter les images de blé en vue de l'analyse ultérieure.

2.1 Introduction

Dans ce chapitre, nous aborderons la conception de notre outil d'annotation en mettant l'accent sur le développement de l'ontologie. L'ontologie joue un rôle central dans notre système, car elle fournit une structure conceptuelle qui nous permet de représenter et de modéliser les connaissances spécifiques à notre domaine d'étude. Nous détaillerons le processus de développement de l'ontologie, en décrivant les différentes classes, propriétés et relations utilisées pour représenter les entités et les concepts clés. De plus, nous proposerons un processus d'annotation, qui utilise l'ontologie pour guider et faciliter l'annotation des données en leur attribuant une signification sémantique.

2.2 Processus d'annotation :

Le processus d'annotation est divisée en plusieurs étapes :

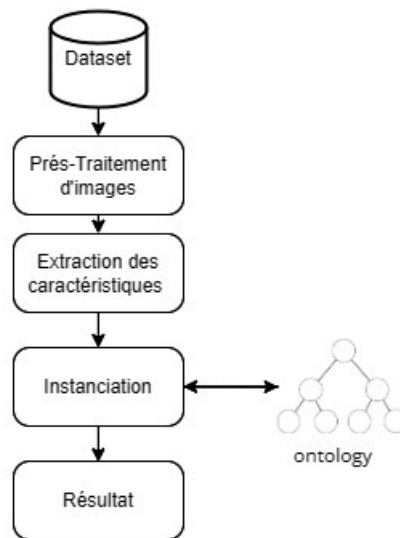


FIGURE 2.1 – Processus d’annotation

1. **Acquisition des images** : cette étape consiste à collecter des images de blé.
2. **Prétraitement des images** : Avant d’extraire des caractéristiques, il est souvent nécessaire de prétraiter les images pour les améliorer. Cette étape inclue plusieurs opérations.
3. **Extraction des caractéristiques** : Cette étape consiste à utiliser des algorithmes d’extraction de caractéristiques pour extraire des informations pertinentes des images basé sur les différents symptômes des maladies.
4. **Annotation sémantique** : Cette étape consiste à utiliser l’ontologie pour enregistrer les caractéristiques extraites, enrichir le processus de détection des maladies et utiliser des raisonnements basés sur l’ontologie pour effectuer des inférences et améliorer la précision de la détection en exploitant les connaissances représentées dans l’ontologie.

2.3 Développement de l’ontologie

La construction des ontologies implique de clarifier les questions conceptuelles et techniques, d’obtenir des définitions claires et d’atteindre un engagement ontologique pour construire des ontologies neutres [4]. Cependant, ce processus peut être difficile en raison du manque de consensus parmi les experts quant aux concepts et aux relations qui devraient être inclus dans les ontologies [4].

D’après [31], la construction de l’ontologie se fait selon un cycle composé de 8 étapes :

- Définition du domaine et des objectifs de l’ontologie : Notre objectif principal est d’atteindre l’annotation automatique sémantique des images pour la détection des symptômes des ma-

ladies de blé au niveau des entrepôts de stockage. En construisant une ontologie spécifiquement dédiée à ce domaine.

- Collecte et analyse des sources d'informations pertinentes.
- Identification des concepts clés et de leurs relations.
- Organisation des concepts en hiérarchies et création de classes.
- Définition des propriétés (slots) pour chaque classe.
- Création des instances pour chaque classe.
- Validation de l'ontologie à travers des exemples concrets.
- Révision et mise à jour de l'ontologie au besoin.

2.3.1 Question de compétence

Avant de commencer le processus de construction une pour le développement de l'ontologie, nous proposons un ensemble de questions de compétences : de stockage. :

- Quelles sont les connaissances agricoles nécessaires pour identifier et décrire les symptômes des maladies du blé?
- Quelles sont les différentes maladies du blé qui doivent être prises en compte dans l'ontologie?
- Quels sont les termes et concepts clés liés aux maladies du blé et à leur détection?
- Quelles sont les relations et les hiérarchies entre les différentes entités dans le domaine des maladies du blé?
- Quelles sont les caractéristiques visuelles spécifiques à prendre en compte lors de la détection des symptômes des maladies du blé dans les images?
- Quels sont les outils et les techniques utilisés pour l'analyse d'image et la segmentation des régions d'intérêt liées aux symptômes des maladies du blé?
- Comment représenter et structurer les connaissances dans l'ontologie de manière à faciliter l'annotation automatique et la recherche d'informations sur les symptômes des maladies du blé?

Répondre à ces questions de compétences aidera à orienter et à guider le développement de l'ontologie pour une annotation automatique sémantique efficace des images de détection des symptômes des maladies du blé dans les entrepôts.

2.3.2 Collecte d'information

Une ontologie est une représentation formelle des connaissances d'un domaine spécifique, et la collecte d'informations est une étape importante pour la création ou l'enrichissement d'une ontologie. Pour collecter des informations sur le domaine des maladies de blé dans les entrepôts de stockage, nous avons utilisé diverses sources afin de rassembler des données pertinentes et fiables. Nous avons consulté des articles scientifiques spécialisés dans l'agriculture et des rapports de recherche [83], [10], [65]. Ces sources nous ont permis d'obtenir des informations détaillées sur les types de maladies qui affectent le blé pendant le stockage et les symptômes associés. De plus, nous avons également consulté des experts dans le domaine d'agriculture pour bénéficier de leur expertise et de leurs connaissances pratiques sur les maladies spécifiques au blé dans les entrepôts de stockage. L'ensemble de ces sources nous a permis de recueillir des informations complètes et à jour pour enrichir notre ontologie sur les maladies de blé dans les entrepôts de stockage.

D'après l'analyse de ces ressources, il a été constaté que la présence des maladies de blé peut varier en fonction des conditions environnementales, de la qualité des grains et des pratiques de gestion des entrepôts. et que les symptômes les plus couramment observés sont liés à la couleur, à la texture, à la forme et à l'existence de trous. Ces caractéristiques visuelles ont été largement documentées et identifiées comme des indicateurs clés pour détecter et diagnostiquer les maladies de blé.

- **La couleur** : la couleur des grains peut présenter des variations anormales, telles que des taches, des décolorations ou des nuances inhabituelles, qui sont souvent associées à des maladies spécifiques. Ces changements de couleur peuvent être perceptibles à l'œil nu ou peuvent nécessiter des techniques d'imagerie spéciales pour être détectés.
- **La texture** : la texture des grains peut également être altérée par des maladies, se manifestant par des rugosités, des irrégularités de surface ou des formations de structures anormales. La texture des grains sains diffère de celle des grains affectés par des maladies, ce qui permet une distinction visuelle significative lors de l'inspection.
- **La forme** : la forme des grains peut également être affectée par les maladies de blé, avec des déformations, des asymétries ou des irrégularités observées. Ces changements de forme peuvent être caractéristiques de certaines maladies spécifiques et peuvent être utilisés comme un symptôme diagnostique important.
- Enfin, **L'existence de trous dans les grains** : est un symptôme particulièrement préoccupant et fréquemment observé dans plusieurs maladies de blé. Ces trous peuvent être causés par des parasites, des infections fongiques ou d'autres agents pathogènes, et leur présence est

souvent associée à des pertes de rendement significatives.

2.3.3 Un meta-modèle pour la représentation des connaissances du domaines

Une fois que nous avons collecté les informations pertinentes sur les maladies de blé dans les entrepôts de stockage, nous avons utilisé un diagramme de classe pour structurer ces informations de manière organisée. Le diagramme de classe est un outil de modélisation visuelle qui permet de représenter les classes, les propriétés et les relations entre les classes.



FIGURE 2.2 – Méta-Modèle représentant les concepts du domaine

2.3.4 Conceptualisation

Dans cette section, nous aborderons les principaux aspects de la conceptualisation de l'ontologie, y compris la définition des concepts, la spécification des attributs et la modélisation des relations.

Définir les concepts : cette étape consiste à définir la liste de concepts trouvés dans l'étape de collecte d'information.

Concept	Description
Image	Représente une représentation visuelle ou graphique d'un plant de blé ou d'une partie de celui-ci. Il capture les propriétés d'une image
Symptom	Représente les manifestations cliniques ou visuelles observées sur les grains de blé qui sont associées à des maladies spécifiques

TABLE 2.1 – Liste des concepts identifiés

Définir les propriétés : Cette étape consiste à définir pour chaque concept les propriétés qui décrivent les caractéristiques spécifiques.

Définir les relations : cette étape permet d'établir des relations entre les concepts et les propriétés pour refléter les liens logiques entre les différents concepts.

Propriété	Domain	Range
hasCouleur	Image	String
hasTexture	Image	String
hasForme	Image	String
hasHols	Segment	Boolean

TABLE 2.2 – Liste des propriétés

Relation	Domain	Range
hasImage	Dataset	Image

TABLE 2.3 – Liste des relations

2.4 Extraction des caractéristiques

Dans notre contexte nous allons baser sur les descripteurs suivants :

2.4.1 Les descripteur de couleur

Le descripteur de couleur joue un rôle essentiel dans la détection des maladies de blé dans les entrepôts de stockage en permettant une analyse précise et objective des variations de couleur des blé, ce qui facilite l'identification précoce des maladies et la prise de décisions appropriées.

Pour faciliter l'analyse des couleurs nous convertissons les images acquises en image HSV où Hue (teinte), Saturation et Value (valeur), pour extraire des informations plus significatives sur les variations de couleur dans les images. Cela permet d'effectuer des opérations de seuillage ou de filtrage plus précises, de détecter des objets basés sur leur teinte.

Une image HSV est une représentation de l'image où chaque pixel est codé en fonction de ces trois composantes. L'espace de couleur HSV est souvent utilisé pour représenter les images en fonction de leurs teintes qui représente la couleur dominante du pixel, de leur saturation qui indique l'intensité de la couleur et la valeur correspond à la luminosité ou à la brillance du pixel, ce qui facilite l'analyse des couleurs.

2.4.2 Les descripteurs de texture

Les descripteurs de texture jouent également un rôle crucial dans la détection des maladies de blé. Les maladies de blé peuvent souvent provoquer des changements dans la texture de la surface des blé, tels que des taches, des fissures ou des structures anormales. L'utilisation de descripteurs de texture permet de quantifier ces variations et d'identifier les blés affectés par des maladies spécifiques.

Pour l'extraction de la texture du grain de blé nous utilisons la matrice de occurrence de niveau de gris GLCM [91]. C'est une méthode statistique utilisée pour analyser les textures d'une

image en considérant la relation spatiale des pixels. En utilisant GLCM, nous extrayons des mesures statistiques et nous pouvons ensuite obtenir des informations sur les propriétés de texture d'une image.

Dans notre travail, nous allons utiliser les mesures suivantes : le contraste, la corrélation, l'énergie, l'homogénéité, corrélation et entropie.

1. Le contraste :

Dans les images, le contraste est une mesure de la différence de luminance ou de couleur entre les pixels adjacents.[57].

La formule mathématique pour calculer le contraste à partir de la matrice de co-occurrence des niveaux de gris (GLCM) est la suivante :

$$Contraste = \sum_{i=0}^{N_g-1} \sum_{j=0}^{N_g-1} (i-j)^2 \cdot g^2(i, j) \quad (1)$$

Note: N_g est le nombre de niveaux de gris, $g(i,j)$ est l'entrée (i,j) dans le GLCM[29].

où G est le nombre de niveaux de gris dans l'image et GLCM (i,j) représente l'élément de la matrice de co-occurrence des niveaux de gris correspondant aux niveaux de gris i et j .

2. Dissimilarité :

La dissimilarité dans le contexte de l'analyse de texture se réfère à la mesure de la différence entre deux pixels en termes de leurs caractéristiques de texture. Il est similaire à Contraste. Au lieu de pondérer les éléments de façon exponentielle, la dissimilarité augmente linéairement[56].

La formule mathématique pour calculer la dissimilarité à partir de la matrice de co-occurrence des niveaux de gris (GLCM) est la suivante :

$$Dissimilarité = \sum_{i=0}^{N_g-1} \sum_{j=0}^{N_g-1} |i-j| \cdot g(i, j) \quad (2)$$

où G est le nombre de niveaux de gris dans l'image et GLCM (i,j) représente l'élément de la matrice de co-occurrence des niveaux de gris correspondant aux niveaux de gris i et j .

3. Energy :

Nous allons utiliser ainsi l'énergie qui est une mesure statistique extraite de la matrice GLCM qui évalue l'uniformité de la texture de l'image de blé. Une valeur élevée d'énergie indique une texture plus uniforme et lisse, tandis qu'une valeur faible suggère une texture plus hétérogène avec des variations importantes. L'énergie permet d'analyser la répartition des niveaux de gris dans l'image et peut être utilisée pour détecter des motifs complexes ou des

structures texturales spécifiques liées aux maladies du blé [91]. La formule pour l'énergie est la suivante :

$$energy = \sqrt{\sum_{i=0}^{N_g-1} \sum_{j=0}^{N_g-1} g^2(i, j)} \quad (3)$$

Note: N_g est le nombre de niveaux de gris, $g(i,j)$ est l'entrée (i,j) dans le GLCM[29].

4. Homogénéité :

Nous allons utiliser l'homogénéité GLCM pour mesurer l'uniformité des niveaux de gris entre les pixels voisins dans une image de blé. Une valeur élevée d'homogénéité indique une texture plus uniforme et régulière, avec des variations progressives, tandis qu'une valeur plus faible révèle une texture plus hétérogène avec des variations prononcées.

L'homogénéité GLCM est calculée en prenant en compte la somme des éléments carrés de la matrice GLCM, qui représente la similarité des niveaux de gris entre les pixels adjacents. Cela permet d'évaluer la régularité de la texture dans l'image de blé. Une texture homogène peut indiquer une zone saine du blé, tandis qu'une texture hétérogène peut suggérer la présence de symptômes de maladies.

La formule mathématique pour calculer l'homogénéité à partir de la matrice de co-occurrence des niveaux de gris (GLCM) est donnée par :

$$homogeneity = \sum_{i=0}^{N_g-1} \sum_{j=0}^{N_g-1} \frac{1}{1 + (i - j)^2} g(i, j) \quad (4)$$

Note: N_g est le nombre de niveaux de gris, $g(i,j)$ est l'entrée (i,j) dans le GLCM[29].

5. Corrélation

L'analyse de la corrélation dans la matrice GLCM permet de caractériser la régularité ou l'irrégularité de la texture des grains de blé, en fournissant une mesure de la similarité des niveaux de gris entre les pixels adjacents. Cette information est utile pour détecter des anomalies ou des motifs spécifiques liés à certaines maladies de blé, contribuant ainsi à la détection et à la classification des maladies à partir des images de blé.

$$correlation = \sum_{i=0}^{N_g-1} \sum_{j=0}^{N_g-1} (i - \mu)(j - \mu)g(i, j) / \sigma^2 \quad (5)$$

Note: N_g est le nombre de niveaux de gris, $g(i,j)$ est l'entrée (i,j) dans le GLCM, μ est la moyenne GLCM, et σ^2 est la variance GLCM [29].

6. Entropie

Dans le contexte de la détection de la texture du blé, nous allons utiliser l'entropie de la matrice GLCM pour évaluer la complexité et la quantité d'information contenue dans la matrice de corrélation de niveau de gris. Cette entropie est utilisée comme un descripteur de texture pour caractériser la variabilité et la diversité des niveaux de gris dans les images de blé.

En analysant la matrice GLCM du blé, on peut calculer l'entropie en utilisant la formule :

$$entropy = \sum_{i=0}^{N_g-1} \sum_{j=0}^{N_g-1} \log(g(i, j)) \quad (6)$$

Note: N_g est le nombre de niveaux de gris, $g(i, j)$ est l'entrée (i, j) dans le GLCM [29].

Cette entropie permet d'évaluer la rugosité ou la complexité locale de la texture du blé. Une entropie plus élevée peut indiquer la présence de structures plus complexes ou de motifs variés, ce qui peut être lié à des irrégularités ou à des caractéristiques spécifiques des grains de blé. En revanche, une entropie plus faible peut correspondre à des régions de texture plus homogènes ou régulières, ce qui peut indiquer des zones de grains de blé plus uniformes.

2.4.3 Les descripteurs de taille

Les descripteurs de taille jouent un rôle essentiel dans l'analyse des grains de blé. Ces descripteurs permettent de mesurer et de caractériser les dimensions des grains, offrant ainsi des informations précieuses pour leur classification et leur évaluation. Dans notre projet, nous nous sommes intéressés à trois descripteurs de taille spécifiques : la région(area), le rapport d'aspect et le périmètre.

La région est un indicateur de la taille relative d'un grain de blé. Elle permet de quantifier la surface occupée par le grain dans l'image. Un grain de blé de grande taille aura une aire plus élevée par rapport à un grain de petite taille.

Le rapport d'aspect renseigne sur la forme du grain de blé. Il est calculé en comparant la longueur et la largeur du grain.

Le périmètre mesure la longueur de la ligne qui délimite le contour du grain de blé. Il fournit des informations sur la régularité ou l'irrégularité de la forme du grain. Un grain de blé régulier aura un périmètre plus uniforme, tandis qu'un grain de blé irrégulier présentera des variations dans son contour.

2.4.4 Les descripteurs de trou

Dans notre approche de détection des trous, nous utilisons les descripteurs de couleur basés sur l'analyse de l'image RGB. Notre objectif est de repérer la présence de trous en se basant sur des caractéristiques spécifiques de couleur.

En analysant l'image RGB, nous avons observé que les trous dans les grains de blé ont une couleur distincte. Plus précisément, nous avons remarqué que la différence entre les composantes rouge (R) et verte (G) est généralement inférieure à 10, tandis que la composante bleue (B) est souvent supérieure à 60.

En utilisant cette observation, nous avons pu définir des critères de détection simples pour les trous basés sur la couleur des pixels. Ainsi, en analysant les valeurs des composantes R, G et B de chaque pixel, nous pouvons déterminer si ce pixel correspond à un trou ou non.

Ces descripteurs de trou basés sur la couleur nous permettent d'identifier et de localiser les régions des grains de blé qui présentent des cavités. En détectant les trous de cette manière, nous facilitons l'analyse et l'annotation des grains de blé, en fournissant des informations utiles sur les caractéristiques structurelles des grains.

2.5 Extension de l'Ontologie

Après la description de chaque descripteur, nous avons pris la décision d'ajouter de nouvelles propriétés et relations à l'ontologie afin de la rendre plus utile, complète et capable de représenter de manière précise les connaissances du domaine.

Le concept "dataset" peut être décrit comme une classe qui représente un ensemble de données regroupées en fonction de certaines caractéristiques communes. Un dataset peut contenir des instances de différents types de données, tels que des images, des textes, des vidéos, etc. Il peut être utilisé pour organiser et structurer les données, facilitant ainsi leur manipulation et leur analyse. Dans notre contexte, un dataset regroupe les images de grains de blé.

Le concept "dataset" peut avoir des propriétés associées qui décrivent des informations spécifiques, telles que le nom du dataset, sa description, sa source, sa date de création, son format, sa taille, etc. Ces propriétés aident à fournir des métadonnées sur le dataset, ce qui peut être utile pour la recherche, la gestion et la compréhension des données.

Le concept "segment" d'une image peut être défini comme une représentation structurée d'une partie spécifique de l'image qui a été isolée ou extraite en utilisant des techniques de segmentation. Dans le contexte de l'ontologie, un segment peut être considéré comme une entité distincte ayant des caractéristiques et des propriétés spécifiques. Il peut être associé à des informations telles que sa forme, sa taille, sa couleur, etc. Le concept de segment dans notre ontologie permet de modéliser et de représenter les grains de blé identifiés dans une image, facilitant ainsi la re-

cherche, et l'analyse d'images basées sur des caractéristiques spécifiques des segments.

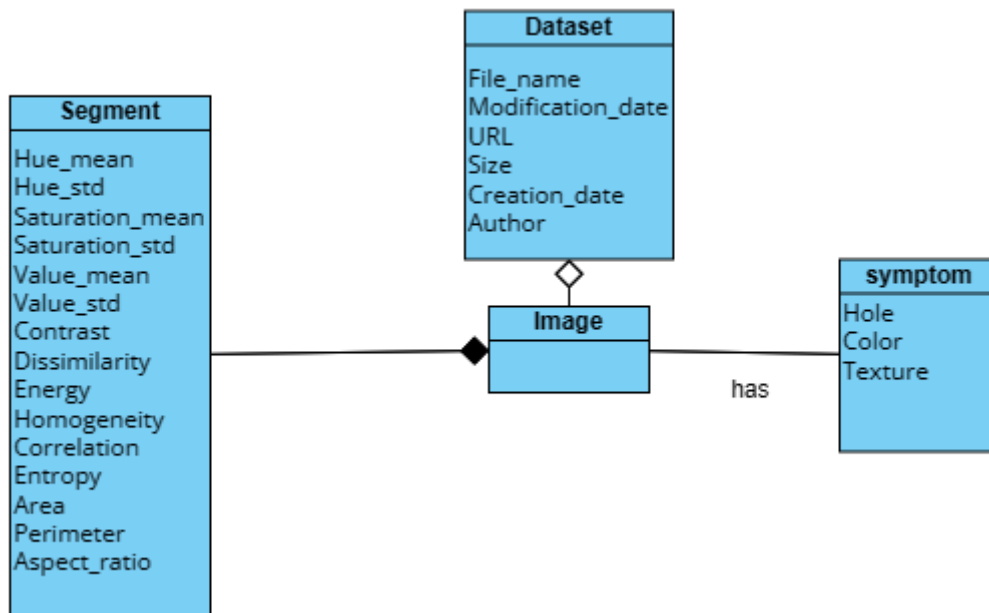


FIGURE 2.3 – Extension du méta-modèle

Ajout de nouveaux concepts : La table 2.4 illustre les nouveaux concepts ajoutés.

Concept	Description
Dataset	Est un concept qui permet de représenter un ensemble d'images
Segment	Il représente les régions ou les objets d'intérêt identifiés dans une image

TABLE 2.4 – Liste des concepts identifiés

Ajout des nouvelles propriétés : La table 2.5 illustre les nouveaux concepts ajoutés.

Propriété	Domain	Range
hasContrast	Segment	Float
hasDissimilarity	Segment	Float
hasEnergy	Segment	Float
hasHomogeneity	Segment	Float
hasCorrelation	Segment	Float
hasEntropy	Segment	Float
hasHueMean	Segment	Float
hasHueStd	Segment	Float
hasSaturationMean	Segment	Float
hasSaturationStd	Segment	Float
hasValueMean	Segment	Float
hasValueStd	Segment	Float
hasArea	Segment	Float
hasPerimeter	Segment	Float
hasAspectRatio	Segment	Float
hasHols	Segment	Boolean

TABLE 2.5 – Liste des propriétés

2.6 Conclusion

Ce chapitre a mis en évidence l'importance de l'ontologie dans le développement de notre outil d'annotation pour les données relatives aux entrepôts de grains. L'ontologie offre une structure conceptuelle solide permettant de représenter et de modéliser les connaissances spécifiques à notre domaine d'étude. Nous avons décrit en détail le processus de développement de l'ontologie, en identifiant les classes, les propriétés et les relations nécessaires pour représenter les entités et les concepts clés. Cette ontologie joue un rôle central dans notre système, car elle permet d'attribuer une signification sémantique aux données annotées. Grâce à l'utilisation de l'ontologie, notre processus d'annotation devient guidé et facilité, garantissant ainsi une annotation précise et cohérente des données.

3.1 Introduction

Dans ce chapitre, nous commencerons par présenter l'implémentation de l'ontologie à l'aide de la bibliothèque Owlready2 de Python. Ensuite, nous décrirons les différentes étapes de l'implémentation de notre outil d'annotation. Nous aborderons d'abord le prétraitement d'image, où nous expliquerons en détail les techniques et les bibliothèques utilisées, telles que la normalisation et le filtrage, qui ont permis une préparation optimale des images pour la segmentation ultérieure. Ensuite, nous nous concentrerons sur la deuxième étape, la segmentation, qui consiste à diviser l'image en régions d'intérêt afin de faciliter l'identification et l'annotation des grains de blé à analyser. Enfin, nous présenterons les différents algorithmes utilisés pour l'extraction de caractéristiques, permettant ainsi une analyse approfondie des données obtenues.

De plus, nous conclurons ce chapitre en présentant un scénario d'annotation détaillé. Ce scénario illustrera l'application pratique de notre système d'annotation automatisée.

3.2 Langages et Outils d'implémentation

Avant de nous lancer dans l'implémentation de notre système, il est crucial de mentionner les différents langages et outils que nous allons utiliser. Ces outils joueront un rôle essentiel dans la réussite de notre projet en fournissant les fonctionnalités nécessaires pour gérer et annoter efficacement les données.

3.2.1 Anaconda

Anaconda est une distribution libre et complète de logiciels qui est spécifiquement conçu pour l'informatique scientifique et la recherche [66]. Il fournit une gamme d'outils et d'environnements

pour le codage dans les langages de programmation Python et R. Ces environnements, souvent appelés environnements de développement intégrés (IDE), offrent une variété de fonctionnalités pour faciliter le développement de code, l'analyse de données, la visualisation et la collaboration.

Avec Anaconda, vous pouvez facilement gérer les paquets et les dépendances en utilisant son gestionnaire de paquets intégré appelé conda. Il simplifie l'installation, la mise à jour et la suppression des paquets, assurant que vous avez accès à une vaste collection de plus de 1500 paquets adaptés pour la science des données, les mathématiques, l'ingénierie et divers autres domaines[66].

En plus de la gestion de paquet, Anaconda fournit plusieurs IDEs populaires, y compris Jupyter Notebook... Ces IDE offrent des interfaces conviviales pour écrire, exécuter et déboguer le code, ainsi que de puissantes capacités pour visualiser les données et créer des documents interactifs.



FIGURE 3.1 – Anaconda Python

3.2.2 Spyder

Spyder sert d'environnement de développement intégré (IDE) spécialement conçu pour le calcul scientifique et l'analyse de données avec le langage de programmation Python. Il présente aux utilisateurs une interface conviviale, permettant un codage, une exécution et un débogage efficaces du code Python.

Spyder, étant un logiciel open-source, est disponible gratuitement et peut être personnalisé en fonction des préférences individuelles. De plus, Spyder favorise une collaboration transparente en facilitant l'intégration de diverses API.



FIGURE 3.2 – Logo spyder

3.2.3 Python

Python est un langage de programmation puissant, flexible, facile à lire et à comprendre. Il a une grande communauté de développeurs qui contribuent à son développement et à son entretien. En outre, Python dispose d'un large éventail de bibliothèques qui le rendent utile pour diverses applications telles que l'analyse de données, l'apprentissage automatique, le développement web, le calcul scientifique, et plus encore [61].

Grâce à sa richesse d'outils, il s'impose comme le langage idéal pour le développement de notre projet.

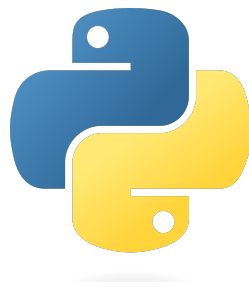


FIGURE 3.3 – Logo Python

3.2.4 OpenCv-Python

OpenCV-Python est un outil puissant pour le traitement d'image qui fournit un large éventail de fonctions et d'algorithmes pour les tâches de vision en temps réel par ordinateur. Il offre diverses transformations géométriques, telles que la translation, la rotation et l'échelle, permettant une manipulation flexible des images. De plus, OpenCV-Python inclut des fonctionnalités comme le seuillage, le filtrage et la détection d'objets, ce qui en fait une bibliothèque polyvalente pour les applications de vision par ordinateur. Connue pour sa rapidité et sa précision dans le traitement d'image, OpenCV-Python est devenu un choix populaire parmi les développeurs à la recherche de solutions efficaces dans le domaine de la vision par ordinateur[20].

3.2.5 Os

Le module Python "os", qui fait partie de la bibliothèque Python Standard, facilite l'interaction avec le système d'exploitation. Il permet aux programmes Python de réaliser un large éventail de tâches associées au système d'exploitation.

En utilisant le module "os", les développeurs Python accèdent aux fonctions qui permettent les opérations de fichiers et de répertoires, la gestion des processus, la manipulation des variables d'environnement, et plus encore. Cela inclut la réalisation d'actions telles que la création, la suppression et le renommage de fichiers et de répertoires, la navigation dans les chemins de fichiers, l'exécution de commandes système et l'accès à des fonctionnalités spécifiques au système[86].

En important et en utilisant le module "Os", les programmeurs Python peuvent écrire du code qui communique efficacement avec le système d'exploitation sous-jacent. Cela simplifie la mise en œuvre de la gestion des fichiers, de la gestion des processus et des opérations liées au système dans les applications Python.

3.2.6 Numpy

NumPy est une bibliothèque Python très utilisée qui excelle dans la réalisation de calculs numériques basés sur des tableaux. Il présente une approche robuste et efficace pour travailler avec des tableaux, qui sont des structures de données capables de contenir de multiples dimensions de données. NumPy comprend une vaste collection de fonctions et d'opérations conçues pour manipuler, analyser et effectuer des calculs mathématiques sur des tableaux. Son application couvre divers domaines, y compris la science des données, l'apprentissage automatique et l'informatique scientifique [8]. Notamment, NumPy est vénéré pour ses performances optimisées et sa capacité à manipuler efficacement des ensembles de données importants.

3.2.7 PyQt

D'après [84]PyQt est un choix populaire pour la programmation graphique en Python en raison des avantages clés suivants :

- Compatibilité multiplateforme : PyQt permet aux développeurs de créer des applications qui s'exécutent sur plusieurs plateformes, y compris Windows, macOS, Linux et des plateformes mobiles comme Android et iOS
- Bibliothèque Qt étendue : PyQt donne accès à la bibliothèque Qt complète, qui offre des composants et des fonctionnalités pré-construits pour le développement de l'interface graphique.

- Langage de programmation Python : PyQt est basé sur Python, connu pour sa simplicité, sa lisibilité et sa vaste bibliothèque standard, ce qui facilite la création d'applications GUI pour les développeurs.
- Communauté active et soutien : PyQt dispose d'une grande communauté active de développeurs, assurant des mises à jour, une maintenance et un soutien continu.
- Intégration avec d'autres bibliothèques Python : PyQt s'intègre parfaitement aux bibliothèques et cadres Python populaires, permettant aux développeurs de tirer parti de leurs capacités d'analyse de données, de visualisation et d'autres fonctionnalités avancées.

3.3 Implémentation de l'Ontologie

L'implémentation de l'ontologie dans notre projet est réalisée en utilisant la bibliothèque Owlready2 en Python. Owlready facilite la création, la manipulation et le raisonnement sur les ontologies. L'utilisation de owlready nous permet de définir les classes, les propriétés et les relations nécessaires pour représenter notre domaine d'étude. La figure (Figure 3.4) montre une partie de la création de l'ontologie proposée.

```
def create_ontology(self):
    # Define the SWRL-like rule as a DL query
    onto = get_ontology("http://test.org/onto.owl")
    with onto:
        class Images(Thing):pass
        class Dataset(Thing):pass
        class symptome(Thing):pass
        class segment(Thing):pass
        class hole(Images >> bool, FunctionalProperty):pass
        class color(Images >> str, FunctionalProperty):pass
        class texture(Images >> str, FunctionalProperty):pass
        class hue_mean(segment >> float, FunctionalProperty):pass
        class hue_std(segment >> float, FunctionalProperty):pass
        class saturation_mean(segment >> float, FunctionalProperty):pass
        class saturation_std(segment >> float, FunctionalProperty):pass
        class value_mean(segment >> float, FunctionalProperty):pass
        class value_std(segment >> float, FunctionalProperty):pass
        class contrast(segment >> float, FunctionalProperty):pass
        class dissimilarity(segment >> float, FunctionalProperty):pass
        class energy(segment >> float, FunctionalProperty): pass
        class homogeneity(segment >> float, FunctionalProperty):pass
        class correlation(segment >> float, FunctionalProperty):pass
        class entropy(segment >> float, FunctionalProperty): pass
        class entropy(segment >> float, FunctionalProperty): pass
        class area(segment >> float, FunctionalProperty): pass
        class perimeter (segment >> float, FunctionalProperty): pass
        class aspect_ratio(segment >> float, FunctionalProperty): pass
        class holes(segment >> bool, FunctionalProperty): pass
        AllDisjoint([Images, Dataset,segment,symptome])
```

FIGURE 3.4 – Partie du Code de Création de l'Ontologie

De plus, Owlready offre des fonctionnalités avancées telles que l'inférence automatique et la recherche de connaissances, ce qui nous permet d'enrichir notre système d'annotation avec des capacités sémantiques.

3.4 Processus d'Annotation

Dans cette section, nous aborderons le processus d'annotation, en mettant l'accent sur les étapes de pré-traitement et d'extraction des caractéristiques.

3.4.1 Etape de Pré-Traitement

Pour préparer nos images de manière optimale pour l'étape d'extraction des caractéristiques, nous avons implémenté les algorithmes suivants :

- Pour simplifier les données, nous avons commencé par convertir l'image en niveaux de gris. Les images en niveaux de gris ont une seule composante de couleur (luminosité) par pixel, ce qui peut faciliter les opérations d'analyse et de traitement ultérieures.
- Ensuite, nous avons utilisé un algorithme de filtrage pour éliminer le bruit et les artefacts indésirables, améliorant ainsi la qualité globale des images. Nous avons utilisé le filtre de Gaussian Blur [19] pour obtenir des résultats précis et adaptés à notre domaine d'étude :
- Pour conclure notre étape de pré-traitement, nous avons appliqué l'algorithme de segmentation de Canny [46] pour détecter les contours des grains de blé. Cela nous a permis de séparer chaque grain individuellement afin de pouvoir les traiter dans les prochaines étapes.

GaussianBlur

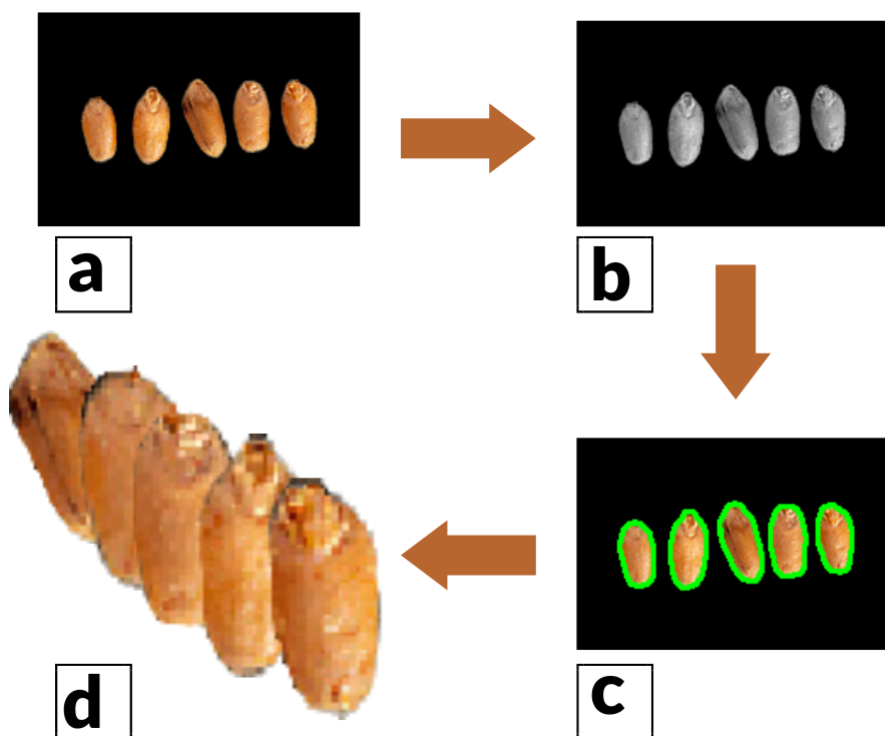
Le flou gaussien [19] est une technique de traitement d'image largement utilisée qui peut être appliquée aux images pour réduire le bruit et lisser les bords coupants. Il fonctionne en convoluant l'image avec une fonction gaussienne, qui est une courbe en forme de cloche qui représente la distribution des valeurs dans l'image. La quantité de flou appliquée à l'image dépend de la taille du noyau gaussien utilisé pour la convolution.

Dans OpenCV, le flou gaussien peut être appliqué en utilisant une fonction, qui prend une image d'entrée et lui applique un filtre gaussien. La fonction permet de spécifier la taille du noyau et l'écart-type de la distribution gaussienne.

Canny :

Canny [46] est une technique de détection de bord qui suit les critères de bonne détection, de bonne localisation et de réponse minimale aux résultats de détection de bord de haute qualité. Il utilise un double seuil pour la révélation des contours et applique le filtre gaussien pour supprimer tout bruit d'une image.

L'algorithme Canny peut être implémenté en Python en utilisant la bibliothèque OpenCV, qui a une fonction intégrée appelée «cv2.Canny()». Cette fonction prend une image d'entrée et deux valeurs de seuil comme arguments et retourne les bords détectés.



a :image RGB principale, **b** : Image en niveaux de gris filtrée, **c** :contourage et segmentation, **d** :Grain de blé unique.

FIGURE 3.5 – Les Etapes de Pré-Traitement

3.4.2 Ètape d'Extraction des Caractéristiques :

Couleur : Nous avons utilisé l'espace de couleur HSV pour extraire les caractéristiques de couleur de l'image. Cette approche nous a permis d'analyser les composantes teinte, saturation et valeur de chaque pixel, ce qui nous a permis de capturer et de quantifier efficacement les caractéristiques des couleurs.

Hue :

— La teinte représente la longueur d'onde dominante de la lumière réfléchie ou transmise par

un objet.

- Il est mesuré en degrés allant de 0 à 360.
- 0 et 360 représentent le rouge, 120 le vert et 240 le bleu.
- La teinte peut être utilisée pour distinguer les différentes couleurs d'une image.

Saturation :

- La saturation représente la pureté de la couleur.
- Il est mesuré en pourcentage de 0% (gris) à 100% (complètement saturé).
- Une couleur complètement saturée n'a pas de blanc ou de noir.
- La saturation peut être utilisée pour ajuster l'intensité d'une couleur dans une image.

Valeur :

- La valeur représente la luminosité ou l'intensité de la couleur.
- Il est mesuré en pourcentage de 0% (noir) à 100% (entièrement brillant).
- Une valeur élevée indique une couleur vive, tandis qu'une valeur faible indique une couleur foncée.
- La valeur peut être utilisée pour ajuster la luminosité ou l'obscurité d'une image.

```
#####  
#COLOR FEATURES EXTRACTION  
  
# Convert the image to HSV color space  
hsv_image = cv2.cvtColor(image, cv2.COLOR_BGR2HSV)  
  
# Extract the HSV components  
hue = hsv_image[:, :, 0]  
saturation = hsv_image[:, :, 1]  
value = hsv_image[:, :, 2]  
  
# Compute the mean and standard deviation of each component  
hue_mean = np.mean(hue)  
hue_std = np.std(hue)  
saturation_mean = np.mean(saturation)  
saturation_std = np.std(saturation)  
value_mean = np.mean(value)  
value_std = np.std(value)  
gray_image = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
```

FIGURE 3.6 – Code pour l'Extraction des Caractéristiques de Couleur

Caractéristiques Extraites	Fonction Utilisé
Teinte	hsv_image[:, :, 0]
Saturation	hsv_image[:, :, 1]
Valeur	hsv_image[:, :, 2]
le moyen	np.mean()
l'écart-type	np.std()

TABLE 3.1 – Liste des caractéristiques de Couleur Extraites des Images à l'aide de la Fonction de Bibliothèque OpenCV

```
Out[1]: {'hue_mean': 3.12835763545615,
        'hue_std': 7.011262593767486,
        'saturation_mean': 20.658855031800226,
        'saturation_std': 47.987103797202614,
        'value_mean': 28.35609847863241,
        'value_std': 63.85111394568266}
```

FIGURE 3.7 – Exemple de Résultat de la Fonction Extract_Color_Features()

Texture : Dans cette partie, nous avons opté pour la méthode de la Matrice de Co-occurrence des Niveaux de Gris (GLCM).L'utilisation de la méthode GLCM (Gray-Level Co-Occurrence Matrix) nous a permis d'extraire des caractéristiques de couleur significatives de l'image(contraste, dissimilarité, énergie, homogénéité, corrélation,et entropie), fournissant une base solide pour l'analyse et le traitement ultérieurs.

```
#####
#texture features extraction
gray_image = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
# Define the properties of the GLCM
distances = [1] # Specify the distance(s) between pixels
angles = [0, np.pi/4, np.pi/2, 3*np.pi/4] # Specify the angle(s) for texture measurement
levels = 256 # Number of gray levels

# Compute the GLCM
glcm = graycomatrix(gray_image, distances, angles, levels=levels, symmetric=True, normed=True)

# Extract texture features from the GLCM
contrast = graycoprops(glcm, 'contrast').mean()
dissimilarity = graycoprops(glcm, 'dissimilarity').mean()
energy = graycoprops(glcm, 'energy').mean()
homogeneity = graycoprops(glcm, 'homogeneity').mean()
correlation = graycoprops(glcm, 'correlation').mean()
entropy = -np.sum(glcm * np.log2(glcm + np.finfo(float).eps))
```

FIGURE 3.8 – Code pour l'extraction des caractéristiques de texture

caractéristiques extraites	Fonction Utilisé
GLCM	graycomatrix(gray_image, distances, angles, levels=levels, symmetric=True, normed=True)
Contraste	graycoprops(glcm, 'contrast').mean()
Dissimilarité	graycoprops(glcm, 'dissimilarity').mean()
énergie	graycoprops(glcm, 'energy').mean()
Homogénéité	graycoprops(glcm, 'homogeneity').mean()
Corrélation	graycoprops(glcm, 'correlation').mean()
Entropie	-np.sum(glcm * np.log2(glcm + np.finfo(float).eps))

TABLE 3.2 – Liste des caractéristiques texture extraites des images à l'aide de la fonction de bibliothèque OpenCV

Taille : Dans cette section, nous présentons la mise en œuvre de l'algorithme d'extraction des caractéristiques de taille pour les grains de blé dans une image. Le code extrait effectue les étapes suivantes :

```
# convert the image to grayscale
gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
up=1
low=255
# apply thresholding to convert grayscale to binary image
ret,thresh = cv2.threshold(gray,up,low,0)
# find the contours
contours,hierarchy = cv2.findContours(thresh, cv2.RETR_TREE,cv2.CHAIN_APPROX_SIMPLE)
largest_contour=contours[0]
max_area= cv2.contourArea(largest_contour)
# Iterate over each contour and calculate size features
for contour in contours:
    # Calculate the area of the contour
    area = cv2.contourArea(contour)
    # Check if the current contour has a larger area than the previous maximum area
    if area > max_area:
        max_area = area
        largest_contour = contour
contour1= largest_contour
# Calculate perimeter
area = cv2.contourArea(contour1)
perimeter = cv2.arcLength(contour1, True)
# Calculate the bounding box
x, y, w, h = cv2.boundingRect(contour1)
# Calculate width and height
width = w
height = h
# Normalize the size features by calculating ratios
h1, w1, channels = image.shape
area1 = area / (w1 * h1)
perimeter1 = perimeter / (w1 + h1)
# define function to compute aspect ratio
def aspect_ratio(cnt):
    x, y, w, h = cv2.boundingRect(cnt)
    ratio = float(w)/h
    return ratio
# select first contour
cnt = largest_contour
# find the aspect ratio
ar = aspect_ratio(cnt)
# round it to two decimal points
ar = round(ar, 2)
```

FIGURE 3.9 – Code pour l'extraction des caractéristiques de taille

Le code fourni est responsable de l'extraction des caractéristiques de taille des grains de blé. Il commence par convertir l'image d'entrée en niveaux de gris en utilisant la fonction `cvtColor()` de la bibliothèque OpenCV. Ensuite, un seuillage est appliqué pour convertir l'image en niveaux de gris en image binaire, ce qui permet une meilleure détection des contours.

En utilisant la fonction `findContours()`, les contours sont extraits de l'image binaire. Parmi ces contours, l'algorithme identifie le plus grand contour en fonction de sa zone. La surface et le péri-

mètre du plus grand contour sont calculés en utilisant les fonctions `contourArea()` et `arcLength()` respectivement.

L'algorithme calcule la boîte de délimitation, la largeur et la hauteur du plus grand contour à l'aide de la fonction `boundingRect()`. Ces caractéristiques de taille sont ensuite normalisées en les divisant avec les dimensions de l'image originale, ce qui permet des comparaisons significatives entre différentes tailles d'image.

Pour analyser davantage la morphologie du grain de blé, le code définit une fonction, `aspect_ratio()`, pour calculer le rapport d'aspect d'un contour. Le rapport d'aspect est calculé en divisant la largeur par la hauteur de la zone de délimitation du contour.

Enfin, le rapport d'aspect calculé est arrondi à deux décimales et stocké dans la variable `ar`.

L'extraction des caractéristiques de taille, y compris la superficie, le périmètre, la largeur, la hauteur, le rapport de surface, le rapport de périmètre et le rapport d'aspect, fournit des renseignements précieux sur les caractéristiques physiques et les proportions des grains de blé dans l'image.

```
Out[3]: {'Area': 0.11330406138303961,  
        'Perimeter': 1.0837353632001594,  
        'Aspect Ratio': 1.85}
```

FIGURE 3.10 – Exemple de sortie du caractéristiques taille

Trous : Dans cette section, nous implémentons un algorithme de détection des trous dans l'image des grains de blé. Le fragment de code effectue les étapes suivantes :

```

#holes detection
# Split the image into individual color channels
b, g, r = cv2.split(image)

diff_rg = np.abs(g - r)
diff_rg2=np.abs(r - g)
add = r+g
add2=r+b
diff_gb = np.abs(g - b)
diff_gb2=np.abs(b - g)

# Create a mask based on the condition (diff_rg < 10)
mask = (((diff_rg < 10) | (diff_rg2 < 10)) & ((b > 60) & ((add > 20) | (add2 > 20))) )

# Set new color values (red) for pixels that satisfy the condition
r[mask] = 255
g[mask] = 0
b[mask] = 0

# Merge the modified color channels back into a single image
modified_image = cv2.merge((b, g, r))
# Initialize a counter for red pixels
red_pixel_count = 0

# Iterate over each pixel in the image
for row in range(image.shape[0]):
    for col in range(image.shape[1]):
        # Get the pixel values for each channel (B, G, R)
        b, g, r = modified_image[row, col]
        # Check if the red channel value is above the threshold
        if (b, g, r) == (0,0,255):
            red_pixel_count += 1
if (red_pixel_count>24):
    result= True
else:
    result=False

```

FIGURE 3.11 – Code détection des trous

Le code fourni implémente un algorithme de détection des trous. Il commence par diviser l'image d'entrée (grain de blé individuel) dans ses canaux de couleur individuels (bleu, vert et rouge). Les différences absolues entre les canaux vert et rouge, ainsi qu'entre les canaux rouge et vert, sont calculées et stockées dans les variables `diff_rg` et `diff_rg2`, respectivement.

De plus, les sommes des canaux rouge et vert et les sommes des canaux rouge et bleu sont calculées et stockées dans les variables `add` et `add2`, respectivement. Les différences absolues entre les canaux vert et bleu et entre les canaux bleu et vert sont calculées et stockées dans les variables `diff_gb` et `diff_gb2`, respectivement.

Ensuite, un masque est créé en fonction de conditions spécifiques. Les conditions impliquent de vérifier si les différences absolues entre les canaux vert et rouge sont inférieures à 10, si les valeurs de canal bleu sont supérieures à 60, et si les sommes des canaux rouge et vert ou les sommes des canaux rouge et bleu sont supérieures à 20.

Le masque est ensuite utilisé pour modifier les valeurs de couleur de l'image. Les pixels qui remplissent les conditions sont mis en rouge ($R = 255, G = 0, B = 0$) en assignant les valeurs correspondantes aux canaux de couleur (`b, g, r`).

Après avoir fusionné les canaux de couleur modifiés en une seule image, un compteur est initialisé pour garder une trace du nombre de pixels rouges.

En itérant sur chaque pixel de l'image, le code récupère les valeurs de pixel pour chaque canal et vérifie si le pixel est rouge en comparant les valeurs de couleur à $(0, 0, 255)$.

Enfin, le code compare le nombre de pixels rouges à un seuil spécifié (24). Si le nombre dépasse le seuil, le résultat de la variable est défini sur Vrai, indiquant la présence de trous. Sinon, le résultat est défini sur Faux.

L’algorithme de détection des trous est conçu pour identifier et analyser la présence de trous dans l’image, en tenant compte des spécificités propres à notre jeu de données. Ces conditions spécifiques sont basées sur les caractéristiques et les propriétés de notre dataset de grains de blé.

Pour finaliser cette étape, nous accomplissons l’instanciation des caractéristiques que nous avons extraites dans l’ontologie.

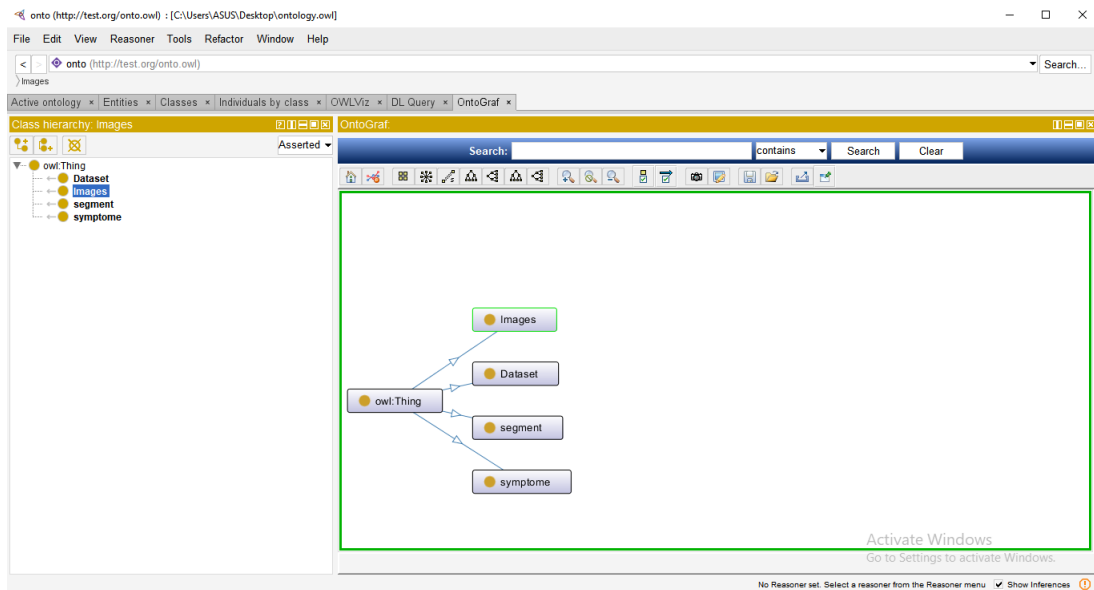


FIGURE 3.12 – Visualisation d’ontologie avant l’instanciation avec Protégé

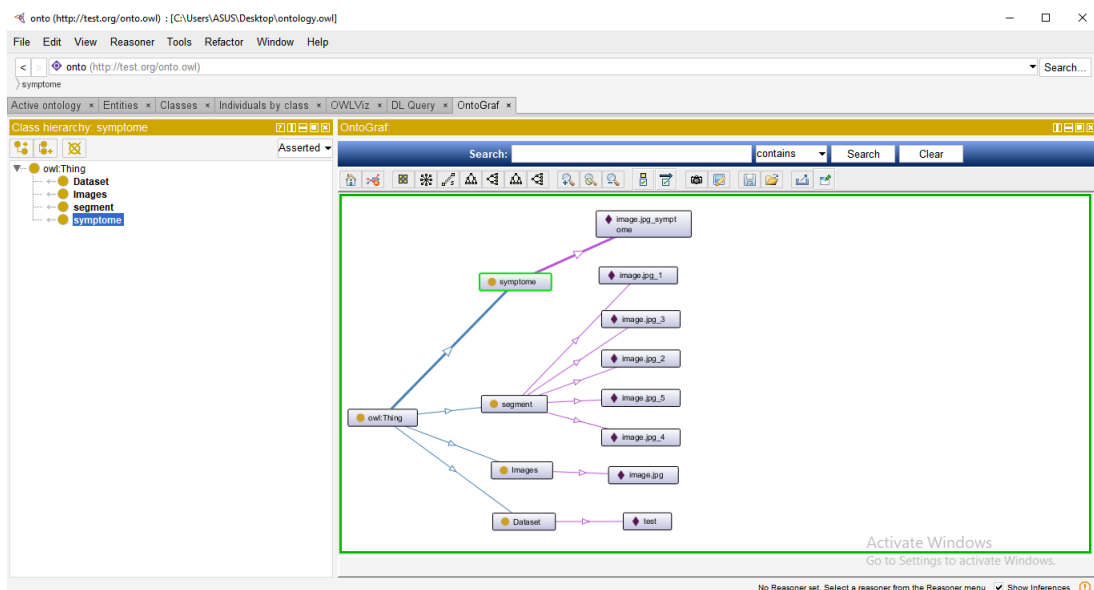


FIGURE 3.13 – Visualisation d’ontologie après l’instanciation avec Protégé

3.5 Un scénario d'exécution

Dans notre scénario d'exécution, nous mettons en évidence les processus précédents qui débute par une interaction entre l'utilisateur et le système. L'utilisateur a la possibilité de visualiser le contenu du dataset sélectionné, qui comprend 11 fichiers images. La figure ci-dessous illustre le contenu du dataset "Unhealthy_wheat_images".



FIGURE 3.14 – Visualisation du contenu d'un dataset

L'utilisateur a la possibilité de télécharger le dataset via notre système. La figure suivante présente la fenêtre qui permet de charger le dataset. L'utilisateur saisit le chemin d'accès du dataset et appuie ensuite sur le bouton "load" pour lancer le processus de téléchargement.

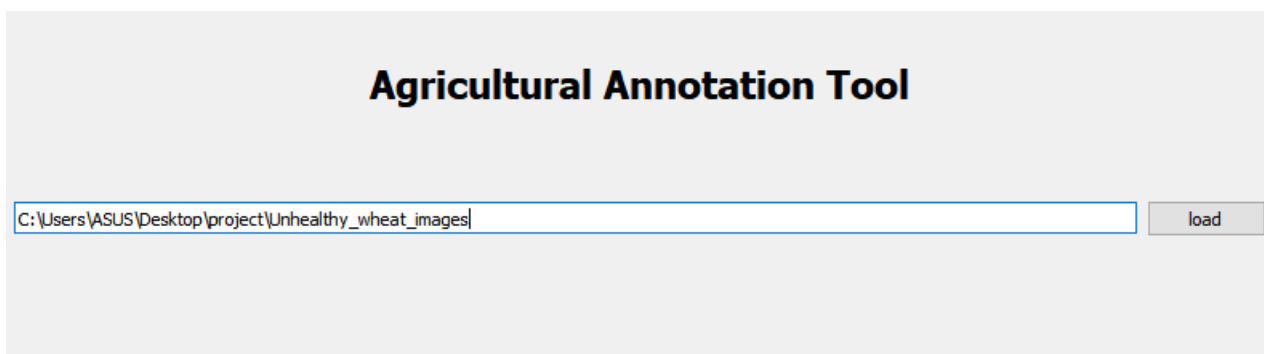


FIGURE 3.15 – Page d'accueil du système d'annotation

La figure ci-dessous présente le message d'erreur qui s'affiche lorsque le chemin du dataset est inexistant :

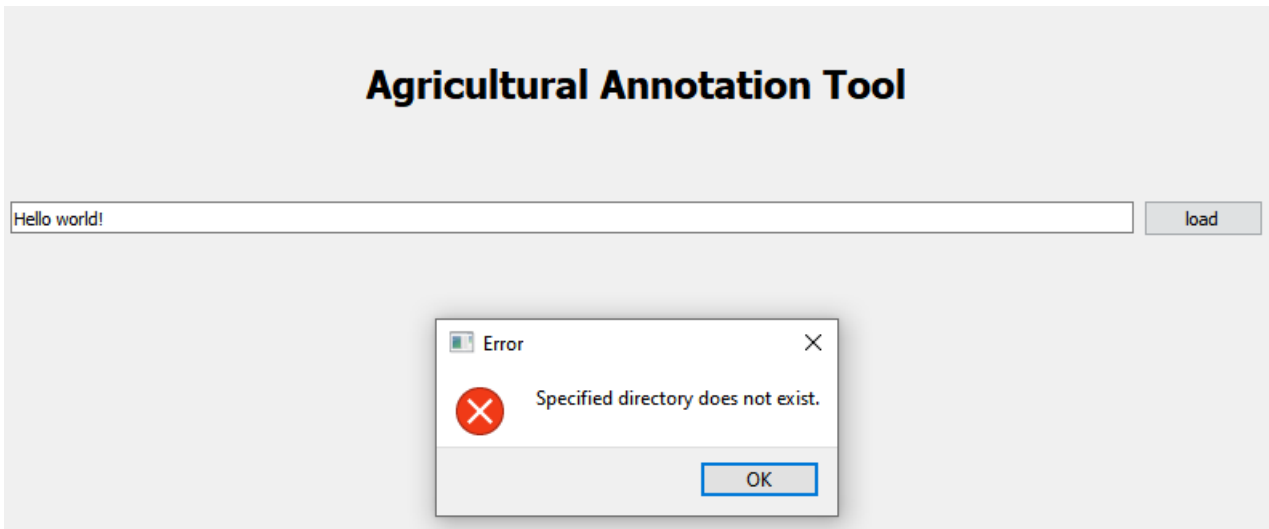


FIGURE 3.16 – Message d'erreur en cas de chemin invalide

La figure ci-dessous illustre le message d'erreur qui apparaît lorsque le chemin du dataset ne contient pas de fichiers d'images :

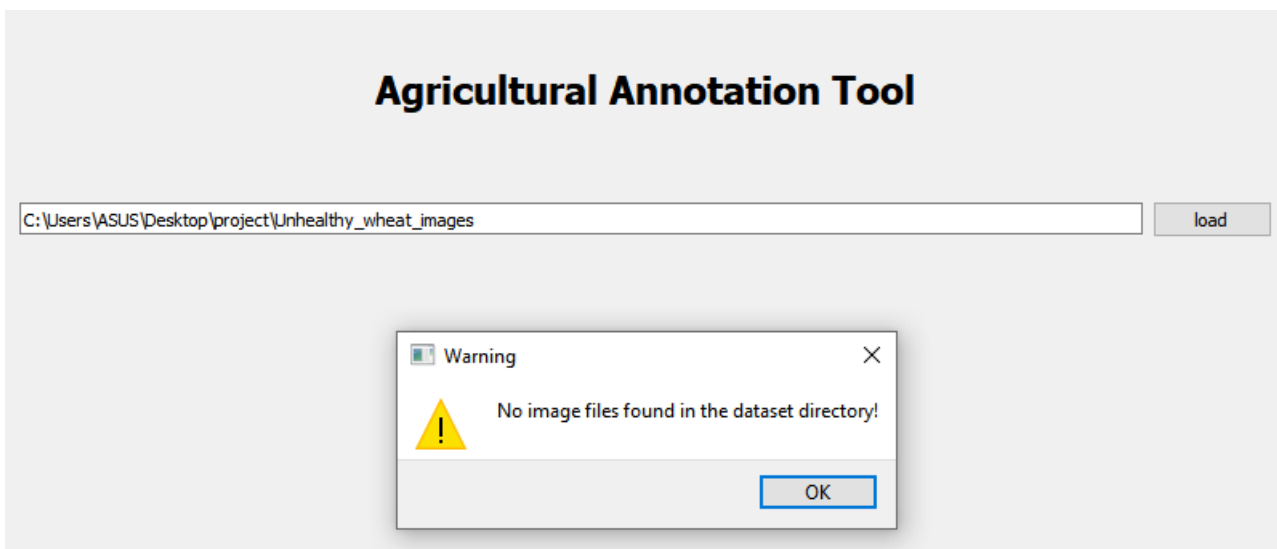


FIGURE 3.17 – Message d'erreur pour un chemin de dataset ne contenant pas de fichiers d'images

La figure ci-dessous montre le message d'erreur qui s'affiche lorsque l'instanciation du dataset existe déjà dans l'ontologie. Cela signifie qu'une instance du dataset avec le même nom est déjà présente dans l'ontologie :

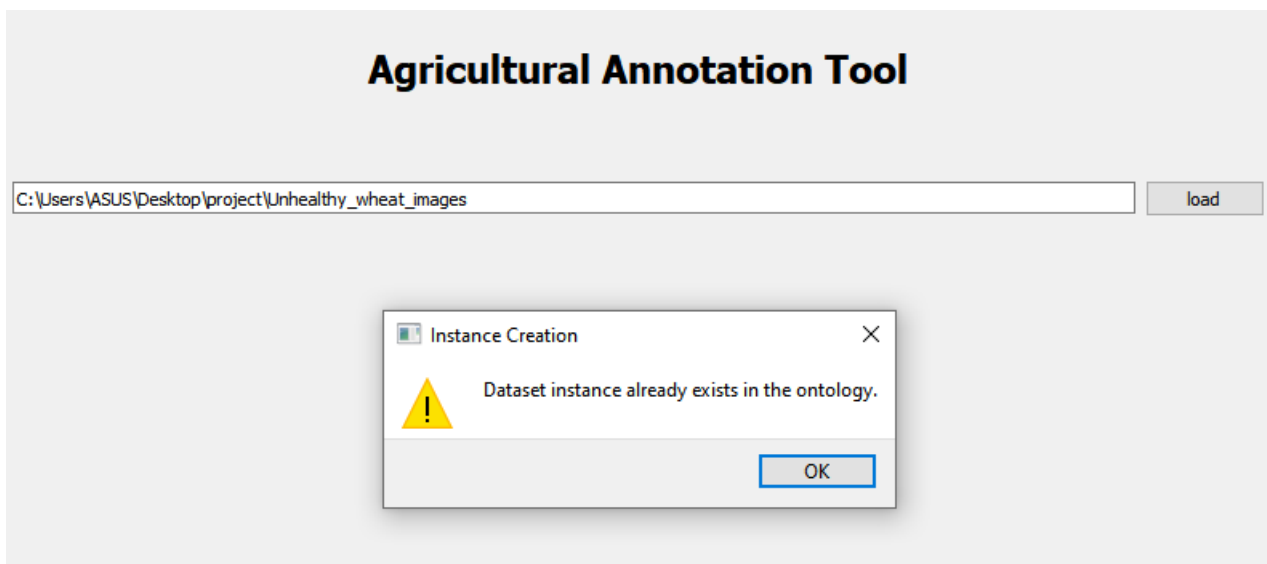


FIGURE 3.18 – Erreur d’instanciation : Instance existante dataset

Si aucune erreur n’est survenue lors des étapes précédentes, l’instance sera créée dans l’ontologie et l’interface suivante sera affichée :

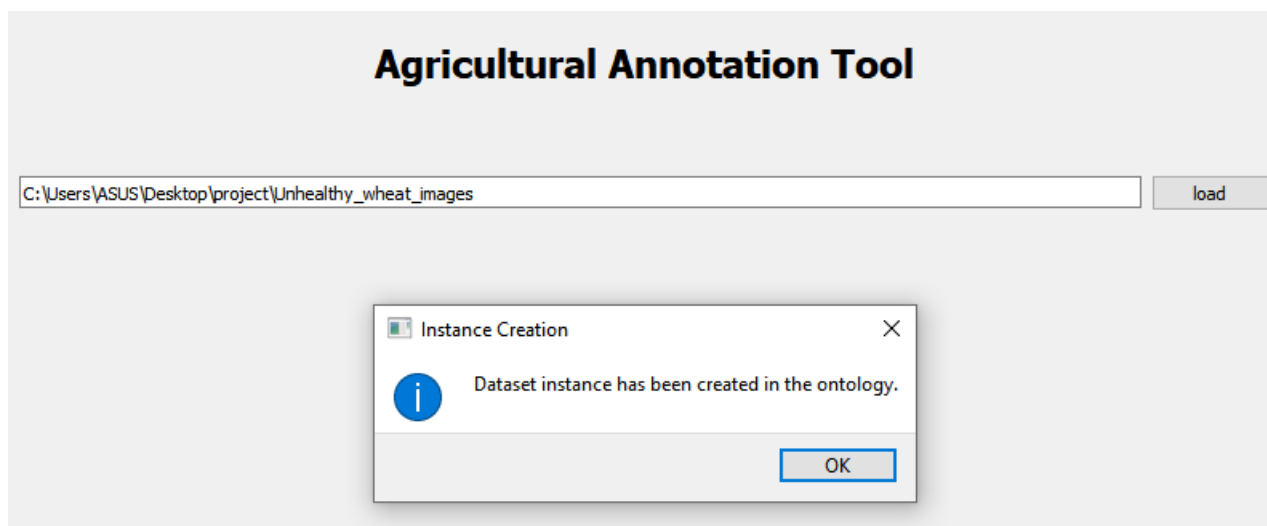


FIGURE 3.19 – La création des instances en cours

La figure suivante présente le contenu du jeu de données, ainsi que les symptômes correspondants extraits et analysés à partir de chaque grain de blé. De plus, l’image modifiée est affichée avec une mise en évidence en rouge des trous dans les grains de blé. La figure se compose de deux sections principales : la section "Manuelle Annotation " et une section dédiée à l’affichage de l’image et de ses informations :

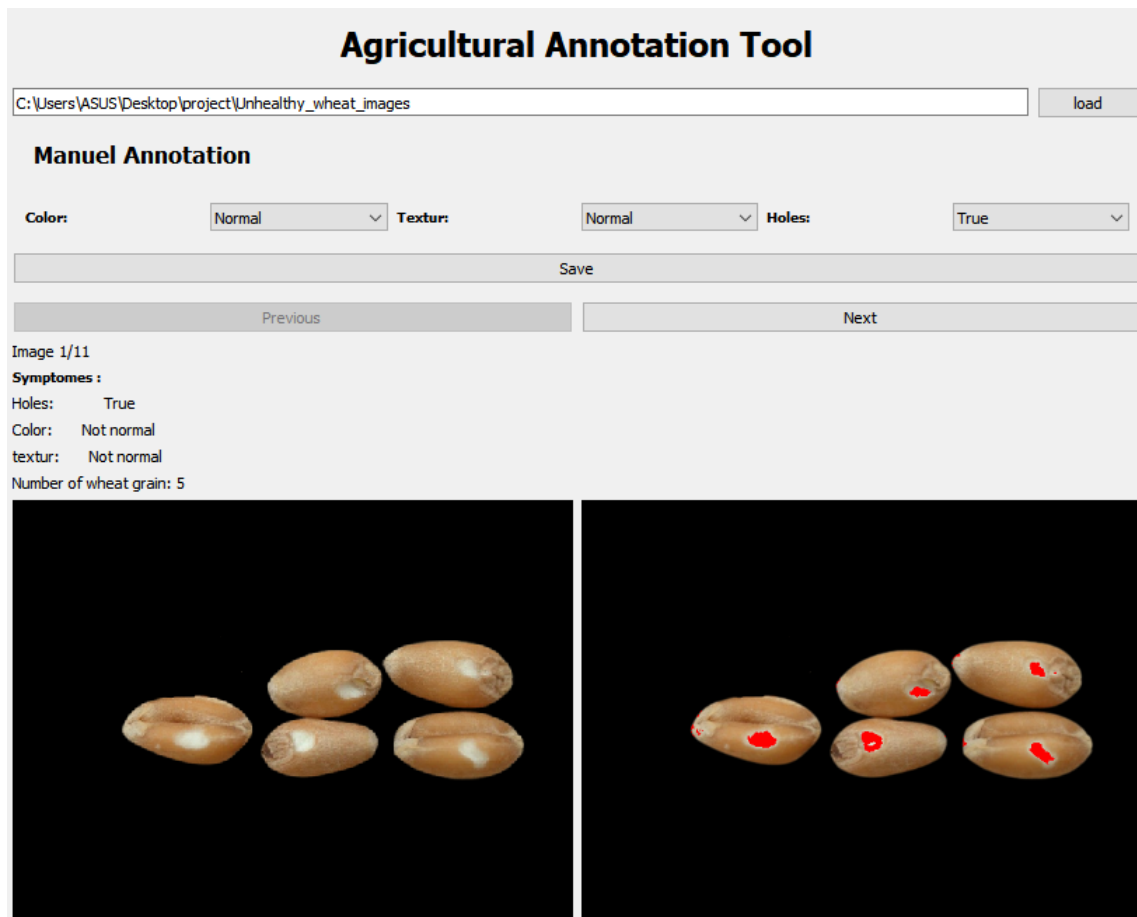


FIGURE 3.20 – Visualisation du dataset avec symptômes et identification des trous.

La figure suivante indique de façon plus visible la partie "Mannuel Annotation", Dans cette partie de l'interface, les utilisateurs ont la possibilité d'annoter l'image en spécifiant les valeurs de couleur, de texture et de trous. Une fois que l'utilisateur a sélectionné ces valeurs, elles sont enregistrées dans l'ontologie en cliquant sur le bouton "Enregistrer". Cette fonctionnalité permet de créer des instances correspondantes dans l'ontologie pour chaque image annotée :

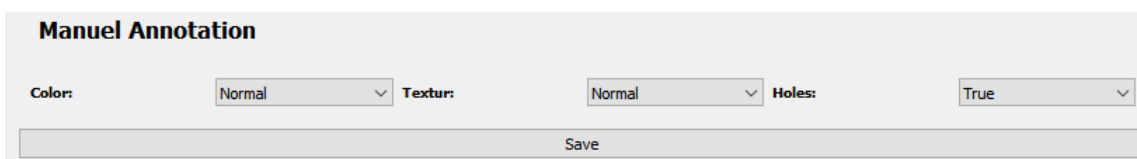


FIGURE 3.21 – Détails d'annotation manuelle

Dans l'interface suivante, nous avons une section dédiée à l'affichage de l'image et de ses caractéristiques. Tout d'abord, nous avons des boutons "next" et "pervious" pour naviguer entre les images du dataset. Ensuite, nous avons une étiquette indiquant le numéro de l'image en cours. Les symptômes, qui sont extraits automatiquement à partir de l'image, sont également affichés. Nous pouvons également voir le nombre de grains de blé présents dans l'image. À gauche de l'interface, l'image originale est affichée, tandis qu'à droite, nous avons l'image modifiée. Cette disposition permet de comparer facilement les deux versions de l'image et de visualiser la mise en évidence

des trous dans les grains de blé :

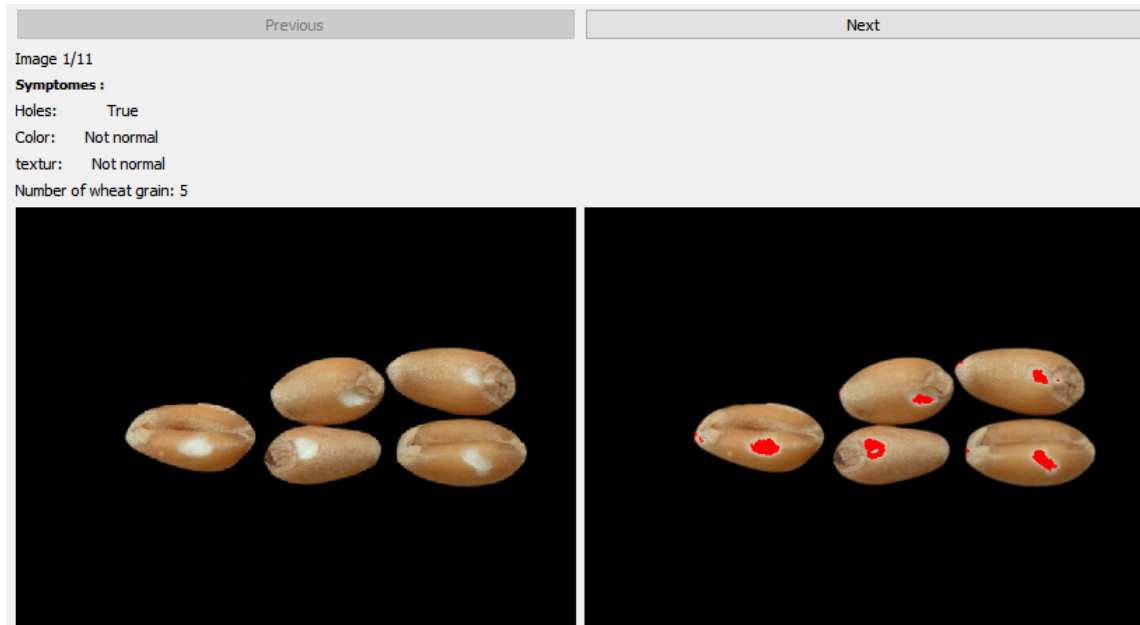


FIGURE 3.22 – Détails des caractéristiques de l'image

3.6 Conclusion

Ce chapitre a couvert en détail l'implémentation de l'ontologie proposée, ainsi que les étapes clés de l'implémentation de notre outil d'annotation pour le big data agricole.

Nous avons commencé par discuter de l'importance de l'ontologie dans la structuration et la représentation des connaissances dans le domaine de la surveillance des maladies du blé. En utilisant Owlready, nous avons pu créer une ontologie spécifique qui permet une modélisation précise des concepts et des relations entre les entités.

Ensuite, nous avons abordé les différentes étapes de l'implémentation de l'outil d'annotation. Le prétraitement d'image a été exploré en détail, mettant en évidence les techniques telles que le filtrage qui est contribué à une préparation optimale des images pour la segmentation ultérieure.

La segmentation, étape cruciale, a été décrite en détail, montrant comment diviser l'image en régions d'intérêt pour faciliter l'identification et l'annotation des grains de blé à analyser. Différents algorithmes ont été utilisés pour cette tâche, offrant une flexibilité et une précision élevées.

Enfin, nous avons abordé l'extraction de caractéristiques, qui a permis une analyse approfondie des données obtenues. L'utilisation d'algorithmes appropriés a joué un rôle clé dans l'identification des caractéristiques pertinentes pour la détection des symptômes maladies du blé, ouvrant ainsi la voie à une meilleure compréhension et à de meilleures décisions dans le domaine agricole.

Ce mémoire a approfondi l'exploration du potentiel du big data agricole en se concentrant sur l'importance des entrepôts de stockage dans le domaine de l'agriculture. Nous avons proposé le développement d'un outil d'annotation sémantique de big data pour détecter les symptômes des maladies des blés dans les entrepôts de grain. Cet outil combine des techniques du Web sémantique et du traitement d'image afin d'extraire des connaissances spécifiques liées aux maladies des blés, permettant ainsi d'identifier les symptômes caractéristiques et de fournir des informations précieuses.

Cependant, il est important de souligner que le domaine du big data agricole présente encore des défis majeurs. La gestion et le traitement des données massives requièrent des efforts continus, de même que le développement de nouvelles approches d'analyse afin de tirer pleinement parti de ces données. De plus, l'intégration de différentes sources de données demeure un défi complexe qui exige une gestion rigoureuse.

Nous avons identifié plusieurs perspectives d'amélioration pour notre outil, notamment l'extension de ses fonctionnalités et son adaptation à d'autres grains et maladies. En poursuivant l'exploration de ces pistes de recherche, nous pourrions apporter une contribution encore plus significative à la gestion efficace des maladies agricoles et à l'amélioration globale de la chaîne agricole.

- 1 ABACHA, Asma Ben et Pierre ZWEIGENBAUM (2010). « Annotation et Interrogation Sémantiques de Textes Médicaux ». In.
- 2 AMARDEILH, Florence (2007). « Web Sémantique et Informatique Linguistique : propositions méthodologiques et réalisation d'une plateforme logicielle ». Thèse de doct. Université de Nanterre-Paris X.
- 3 AMARDEILH, Florence et Danica DAMLJANOVIC (2009). « Du texte à la connaissance : annotation sémantique et peuplement d'ontologie appliqués à des artefacts logiciels ». In : p. 29.
- 4 ARVOR, Damien et al. (2013). « Advances in Geographic Object-Based Image Analysis with ontologies : A review of main contributions and limitations from a remote sensing perspective ». In : *ISPRS Journal of Photogrammetry and Remote Sensing* 82, p. 125-137.
- 5 AUSSENAC-GILLES, Nathalie et al. (2013). « Enrichissement d'ontologies grâce à l'annotation sémantique de pages web. » In : p. 229-234.
- 6 BALLU, Victor (2018). « Une approche simplifiée de l'annotation sémantique d'un corpus spécialisé à l'aide de ressources externes ». In : HAL Id : mem02096754.
- 7 BANGA, Km Sheetal et al. (2018). « Techniques for insect detection in stored food grains : An overview ». In : *Food Control* 94, p. 167-176.
- 8 BAUER, Michael et Michael GARLAND (2019). « Legate NumPy : Accelerated and distributed array computing ». In : p. 1-23.
- 9 BELLON-MAUREL, Véronique et al. (2018). « Le Big Data en agriculture ». In : *Big Data : économie et régulation*, p. 77.
- 10 BETTAHAR, Fehd (2016). « Conception et prototypage d'un système complet pour la surveillance du grain dans les silos de stockage ». Thèse de doct. Université Toulouse le Mirail-Toulouse II.
- 11 BIFFI, Leonardo Josoé et al. (2020). « ATSS deep learning-based approach to detect apple fruits ». In : *Remote Sensing* 13.1, p. 54.

- 12 BLOK, Pieter M et al. (2022). « Active learning with MaskAL reduces annotation effort for training Mask R-CNN on a broccoli dataset with visually similar classes ». In : *Computers and Electronics in Agriculture* 197, p. 106917.
- 13 BOUKADOUM, Omar et al. (oct. 2015). « Annotation Automatique d'Images Rétiennes par le Contenu Visuel et la Structure Arborescente ». In : *Proceedings of the Conference Name*.
- 14 CHEN, Lin et al. (2012). « Tag-based image retrieval improved by augmented features and group-based refinement ». In : *IEEE Transactions on Multimedia* 14.4, p. 1057-1067.
- 15 CHEN, Shi, Meng WANG et Xuan CHEN (2020). « Image annotation via reconstitution graph learning model ». In : *Wireless Communications and Mobile Computing 2020*, p. 1-9.
- 16 CHIHANA, Sipiwe, Jackson PHIRI et Douglas KUNDA (2018). « An IoT based warehouse intrusion detection (E-Perimeter) and grain tracking model for food reserve agency ». In : *International Journal of Advanced Computer Science and Applications* 9.9.
- 17 CHUA, Tat-Seng et al. (2009). « Nus-wide : a real-world web image database from national university of singapore ». In : p. 1-9.
- 18 CUI, Hongwei et al. (2022). « Modeling and Application of Temporal Correlation of Grain Temperature during Grain Storage ». In : *Agriculture* 12.11, p. 1883.
- 19 DEEPTHI, RS et S SANKARAI AH (2011). « Implementation of mobile platform using Qt and OpenCV for image processing applications ». In : p. 284-289.
- 20 DEVI, R (2017). « Geometric transformations and thresholding of images using Opencv-Python ». In : *GRD Journal for Engineering* 2.11.
- 21 DUYGULU, Pinar et al. (2002). « Object recognition as machine translation : Learning a lexicon for a fixed image vocabulary ». In : p. 97-112.
- 22 ENGHOFF, Henrik (2009). « What is taxonomy?—An overview with myriapodological examples ». In : *Soil organisms* 81.3, p. 441-451.
- 23 ESHKOL-TARAVELLA, Iris (2015). « La définition des annotations linguistiques selon les corpus : de l'écrit journalistique à l'oral ». Thèse de doct. Université d'Orléans.
- 24 EVERINGHAM, M et al. (2007). « The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Results ». In : *VOC/voc2007/workshop/index.html*. URL : <http://www.pascal-network.org/challenges>.
- 25 GARSHOL, Lars Marius (2004). « Metadata? Thesauri? Taxonomies? Topic maps! Making sense of it all ». In : *Journal of information science* 30.4, p. 378-391.
- 26 GUARINO, Nicola, Daniel OBERLE et Steffen STAAB (2009). « What is an ontology? » In : *Handbook on ontologies*, p. 1-17.
- 27 GUILLAUMIN, Matthieu et al. (2009). « Tagprop : Discriminative metric learning in nearest neighbor models for image auto-annotation ». In : p. 309-316.

- 28 GUISSÉ, A et al. (2009). « Annotation sémantique pour l'indexation de règles métiers ». In : *L'Homme, M.-C. and Szulman, S., editors, Conférence Internationale sur la Terminologie et l'Intelligence Artificielle (TIA 2009)*, page (electronic medium). Université Paul Sabatier-Toulouse.
- 29 HARALICK, Robert M (1979). « Statistical and structural approaches to texture ». In : *Proceedings of the IEEE* 67.5, p. 786-804.
- 30 HUISKES, Mark J et Michael S LEW (2008). « The mir flickr retrieval evaluation ». In : p. 39-43.
- 31 HYVÖNEN, Eero et al. (2003). « Ontology-Based Image Retrieval. » In : *WWW (posters)*.
- 32 J, Lydia et al. (2022). « Automated food grain monitoring system for warehouse using IOT ». In : *Measurement : Sensors* 24, p. 100472. ISSN : 2665-9174. DOI : <https://doi.org/10.1016/j.measen.2022.100472>. URL : <https://www.sciencedirect.com/science/article/pii/S2665917422001064>.
- 33 JAMROEN, Chaowanan et al. (2020). « An intelligent irrigation scheduling system using low-cost wireless sensor network toward sustainable and precision agriculture ». In : *IEEE Access* 8, p. 172756-172769. ISSN : 21693536. DOI : 10.1109/ACCESS.2020.3025590.
- 34 JHAMB, Shreya et al. (2020). « Understanding Complexity in Language Learning Through Data Visualization Using Python ». In : p. 268-274.
- 35 JIAN, Fuji (2019). « Influences of stored product insect movements on integrated pest management decisions ». In : *Insects* 10.4, p. 100.
- 36 JING, Xiao-Yuan et al. (2016). « Multi-label dictionary learning for image annotation ». In : *IEEE Transactions on Image Processing* 25.6, p. 2712-2725.
- 37 KARIDURAGANAVAR, Rohini M et SG SHIVAPRASAD YADAV (2020). « Design and Implementation of IoT Based Rodent Monitoring and Avoidance System in Agricultural Storage Bins ». In : p. 233-241.
- 38 KODALI, Ravi Kishore, Jeswin JOHN et Lakshmi BOPANA (2020). « IoT monitoring system for grain storage ». In : p. 1-6.
- 39 KRESTENITIS, Marios et al. (2022). « CoFly-WeedDB : A UAV image dataset for weed detection and species identification ». In : *Data in Brief* 45, p. 108575.
- 40 LAC, Louis et al. (2022). « An annotated image dataset of vegetable crops at an early stage of growth for proximal sensing applications ». In : *Data in Brief* 42, p. 108035.
- 41 LI, Lei et al. (2021). « Deep learning-based automatic monitoring method for grain quantity change in warehouse using semantic segmentation ». In : *IEEE Transactions on Instrumentation and Measurement* 70, p. 1-10.
- 42 LI, Xirong et Cees GM SNOEK (2013). « Classifying tag relevance with relevant positive and negative examples ». In : p. 485-488.
- 43 LI, Xirong, Cees GM SNOEK et Marcel WORRING (2009). « Learning social tag relevance by neighbor voting ». In : *IEEE Transactions on Multimedia* 11.7, p. 1310-1322.
- 44 LIU, Dong et al. (2009). « Tag ranking ». In : p. 351-360.
- 45 LONGLEY, Paul A. et al. (2015). *Geographic Information Science and Systems*. Wiley.

- 46 LYNN, ND, AI SOURAV et AJ SANTOSO (2021). « Implementation of Real-Time Edge Detection Using Canny and Sobel Algorithms ». In : 1096.1, p. 012079.
- 47 MA, Yue, Laurent AUDIBERT et Adeline NAZARENKO (2009). « Ontologies étendues pour l'annotation sémantique ». In : p. 205-216.
- 48 MA, Yue, Francois LEVY et Adeline NAZARENKO (2013). « Annotation sémantique pour des domaines spécialisés et des ontologies riches ». In : *Actes de la 20ème conférence du Traitement Automatique du Langage Naturel (TALN 2013)*.
- 49 MAKADIA, Ameesh, Vladimir PAVLOVIC et Sanjiv KUMAR (2010). « Baselines for image annotation ». In : *International Journal of Computer Vision* 90, p. 88-105.
- 50 MAMAT, Normaisharah et al. (2023). « Enhancing Image Annotation Technique of Fruit Classification Using a Deep Learning Approach ». In : *Sustainability* 15.2, p. 901.
- 51 MENZE, Bjoern H et al. (2014). « The multimodal brain tumor image segmentation benchmark (BRATS) ». In : *IEEE transactions on medical imaging* 34.10, p. 1993-2024.
- 52 MILLE, Dominique (2005). « Modèles et outils logiciels pour l'annotation sémantique de documents pédagogiques ». Thèse de doct. Université Joseph-Fourier-Grenoble I.
- 53 MOHANRAJ, K. et al. (2019). « Smart warehouse monitoring using IoT ». In : *International Journal of Engineering and Advanced Technology (2019)* 8 (6). DOI : 10.35940/ijeat.F9355.088619.
- 54 MYLONAS, Nikos et al. (2022). « Eden library : A long-term database for storing agricultural multi-sensor datasets from uav and proximal platforms ». In : *Smart Agricultural Technology* 2, p. 100028.
- 55 PANG, Bo et Lillian LEE (2008). « A survey of opinion mining and sentiment analysis ». In : *Foundations and Trends in Information Retrieval* 2.1-2, p. 1-135.
- 56 PARK, Yujin et Jean-Michel GULDMANN (2020). « Measuring continuous landscape patterns with Gray-Level Co-Occurrence Matrix (GLCM) indices : An alternative to patch metrics? » In : *Ecological Indicators* 109, p. 105802.
- 57 PELI, Eli (1990). « Contrast in complex images ». In : *JOSA A* 7.10, p. 2032-2040.
- 58 PHASINAM, Khongdet, Thanwamas KASSANUK et Mohammad SHABAZ (2022). « Applicability of internet of things in smart farming ». In : *Journal of Food Quality* 2022, p. 1-7.
- 59 PIMENTEL, João Felipe et al. (2021). « Understanding and improving the quality and reproducibility of Jupyter notebooks ». In : *Empirical Software Engineering* 26.
- 60 PUSTEJOVSKY, James et Annie STUBBS (2010). « Natural Language Annotation for Machine Learning ». In : sous la dir. de Nitin INDURKHYA et Fred J. DAMERAU, p. 554-572.
- 61 PYTHON, Why (2021). « Python ». In : *Python Releases Wind* 24.
- 62 RAI, Bipin Kumar, Syed Aamir Hussain ZAIDI et Dhishwari SINGH (2022). « DevOps Sensing Neuro Cluster ». In : p. 69-77.
- 63 RAMADIER, Lionel et al. (2014). « Annotations et inférences de relations dans un réseau lexico-sémantique : application à la radiologie ». In : *TALN : Traitement Automatique des Langues Naturelles*.

- 64 RASMUSSEN, Christoffer Bøgelund et Thomas B MOESLUND (2019). « Maize silage kernel fragment estimation using deep learning-based object recognition in non-separated kernel/stover rgb images ». In : *Sensors* 19.16, p. 3506.
- 65 RENGIFO, Gabriel (1976). « High temperature and high humidity grain storage ». Thèse de doct. Kansas State University.
- 66 ROLON-MÉRETTE, Damien et al. (2016). « Introduction to Anaconda and Python : Installation and setup ». In : *Quant. Methods Psychol* 16.5, S3-S11.
- 67 SAPOUKHINA, Natalia, Tristan BOUREAU et David ROUSSEAU (2022). « Plant disease symptom segmentation in chlorophyll fluorescence imaging with a synthetic dataset ». In : *Frontiers in Plant Science* 13, p. 969205.
- 68 SASIKUMAR, S, PK PANDIKALA et P RAJAPANDI (2013). « An assemble pattern relevance feedback technique for an automatic image annotation ». In : *International Journal of Managment, IT and Engineering* 3.7, p. 316-322.
- 69 SINDWANI, Aseem et al. (2020). « Prediction and Monitoring of stored food grains health using IoT Enable Nodes ». In : p. 516-522.
- 70 SUDARS, Kaspars et al. (2020). « Dataset of annotated food crops and weed images for robotic computer vision control ». In : *Data in brief* 31, p. 105833.
- 71 SURAL, Shamik, Gang QIAN et Sakti PRAMANIK (2002). « Segmentation and histogram generation using the HSV color space for image retrieval ». In : 2, p. II-II.
- 72 SZWEDZIAK, Katarzyna et al. (2020). « Neural modeling of the distribution of protein, water and gluten in wheat grains during storage ». In : *Sustainability* 12.12, p. 5050.
- 73 TARIQ, Amara et Hassan FOROOSH (2017). « Image Annotation using Multi-Layer Sparse Coding ». In : *arXiv preprint arXiv :1705.02460*.
- 74 TOUSCH, Anne-Marie, Stéphane HERBIN et Jean-Yves AUDIBERT (2012). « Semantic hierarchies for image annotation : A survey ». In : *Pattern Recognition* 45.1, p. 333-345.
- 75 TRUONG, Binh, Aixin SUN et Sourav BHOWMICK (2012). « Content is still king : the effect of neighbor voting schemes on tag relevance for social image retrieval ». In : *Proceedings of the ACM International Conference on Multimedia Retrieval (ICMR)*.
- 76 TSIRONIS, V, S BOUROU et C STENTOUMIS (2020). « TOMATOD : EVALUATION OF OBJECT DETECTION ALGORITHMS ON A NEW REAL-WORLD TOMATO DATASET. » In : *International Archives of the Photogrammetry, Remote Sensing & Spatial Information Sciences* 43.
- 77 USBECK, Ricardo et al. (2018). « Benchmarking the Extraction and Disambiguation of Named Entities on the Semantic Web ». In : *Semantic Web* 9.3, p. 389-426. DOI : 10 . 3233/SW-170277.
- 78 VAN DE SANDE, Koen, Theo GEVERS et Cees SNOEK (2009). « Evaluating color descriptors for object and scene recognition ». In : *IEEE transactions on pattern analysis and machine intelligence* 32.9, p. 1582-1596.

- 79 VENKATRAMAN, S., S. K. BHATIA et R. SINGH (2018). « A review on data annotation techniques ». In : p. 134-139.
- 80 VERBEEK, Jakob et al. (2010). « Image annotation with tagprop on the mirflickr set ». In : p. 537-546.
- 81 WANG, Qijian et al. (2022). « Design and Implementation of WSN-based Warehouse Temperature Sensor Monitoring System ». In : 2246.1, p. 012006.
- 82 WANG, X. et al. (2019). « Object detection in video with spatiotemporal sampling networks ». In : *Neurocomputing* 338, p. 155-167. DOI : 10.1016/j.neucom.2019.01.077.
- 83 WHITE, NDG (2001). « Protection des céréales, des oléagineux et des légumineuses à grain entreposées à la ferme contre les insectes, les acariens et les moisissures. Agriculture et Agroalimentaires Canada ». In : *Agriculture et Agroalimentaires Canada*. <http://www.agr.gc.ca/science/winnipeg>.
- 84 WILLMAN, Joshua M (2020). *Beginning PyQt*. Springer.
- 85 YANG, Yang et al. (2014). « Image tagging with social assistance ». In : p. 81-88.
- 86 ZHANG, Denghui et al. (2019). « Sgpxy : Protecting integrity of python applications with intel sgx ». In : p. 418-425.
- 87 ZHENG, Yang-Yang et al. (2019). « CropDeep : The crop vision dataset for deep-learning-based classification and detection in precision agriculture ». In : *Sensors* 19.5, p. 1058.
- 88 ZHOU, Chenhong et al. (2020). « One-pass multi-task networks with cross-task guided attention for brain tumor segmentation ». In : *IEEE Transactions on Image Processing* 29, p. 4516-4529.
- 89 ZHU, Xiaofei, Wolfgang NEJDL et Mihai GEORGESCU (2014). « An adaptive teleportation random walk model for learning social tag relevance ». In : p. 223-232.
- 90 ZNAIDIA, Amel, Hervé LE BORGNE et Céline HUDELLOT (2013). « Tag completion based on belief theory and neighbor voting ». In : p. 49-56.
- 91 ZULPE, Nitish et Vrushsen PAWAR (2012). « GLCM textural features for brain tumor classification ». In : *International Journal of Computer Science Issues (IJCSI)* 9.3, p. 354.

Participation au 1er atelier sur l'agriculture intelligente le 15/12/2022 organisé au niveau du centre de recherche d'agropastoralisme de l'université de Ziane Achour, Djelfa. Avec une communication affichée intitulé "Toward an Agricultural Big Data Annotation Tool "

