



Rapport Projet Classification d'Image - Apprentissage Profond

L'équipe :

AKKAR KHADIJA, ABOUMEJD WISSAL, BELAHRACH SAFAE, FARHAT OTMAN,
DIOURI AYOUB

Tuteur :

M. CARLIER AXEL

Génie Logiciel L12

Date :

11 Mars 2024

ENSEEIHT - Institut National Polytechnique de Toulouse
ÉCOLE NATIONALE SUPÉRIEURE D'ÉLECTRONIQUE, D'ÉLECTROTECHNIQUE,
D'INFORMATIQUE, D'HYDRAULIQUE ET DES TÉLÉCOMMUNICATIONS



Table des matières

1	Introduction	3
2	Première partie : constitution de la base de données	5
2.1	Éléments clés du projet	5
2.1.1	Composition du groupe	5
2.1.2	Titre du projet	5
2.1.3	La base de données	6
2.2	Description illustrée du sujet choisi	6
2.3	Méthodologie d'Acquisition et d'Annotation des Données	8
2.3.1	Acquisition des Données	8
2.3.2	Annotation des Données	9
2.3.3	Prétraitement des Données	9
2.4	Méthodologie de partitionnement des images en ensembles d'entraînement, de validation et de test	10
2.4.1	Méthodologie de Partitionnement	10
2.5	Notre pronostic sur le problème et les résultats	11
2.5.1	Complexité du problème :	11
2.5.2	Pronostic sur la nature de notre problème et sur les résultats	12
2.6	Script de chargement des données	12
2.7	Images de notre base de données	12
3	Deuxième étape : développement de notre solution et évaluation de nos résultats	15
3.1	Construction du réseau neuronal	15
3.2	Entraînement du réseau neuronal	16
3.2.1	Hyperparamètres Utilisés	16
3.2.2	Résultats	16
3.3	Augmentation des données	17
3.3.1	Transformations appliquées	17
3.3.2	Courbes résultantes	17
3.3.3	Matrices de confusion	18
3.4	Optimisation et ajustement	18
3.4.1	Transfer learning	18
3.4.2	Fine-tuning	22
3.5	Discussion et conclusions	23
3.5.1	Analyse des images mal classées	23

3.6 Des pistes d'amélioration	24
4 Conclusion	25

Chapitre 1

Introduction

L'agriculture joue un rôle crucial dans la sécurité alimentaire mondiale en fournissant une source vitale de nourriture pour la population croissante de la planète. Cependant, les cultures agricoles sont fréquemment exposées à diverses menaces, notamment les ravageurs tels que les insectes nuisibles, qui peuvent causer des dommages importants aux récoltes et entraîner des pertes économiques considérables. Dès lors, la détection précoce et la classification précise des insectes s'avèrent essentielles pour la mise en place de mesures de gestion efficaces et durables.



Figure. 1.1 – Agriculture



Figure. 1.2 – Prévention agricole

Objectif du projet

Le présent projet s'inscrit dans la perspective d'appliquer les techniques de l'apprentissage profond à la classification des images d'insectes dans le domaine de l'agriculture. L'objectif principal est de développer un système de classification automatique capable de distinguer différents types d'insectes, qu'ils soient nuisibles ou bénéfiques, à partir d'images de cultures. Ce système permettra aux agriculteurs et aux chercheurs de détecter rapidement la présence d'insectes nuisibles, d'identifier les espèces spécifiques et de prendre des mesures de contrôle appropriées pour protéger les cultures.

**Figure. 1.3 – Domaine Agricole**

Organisation et Structure du rapport

Ce rapport est structuré de manière à présenter de manière concise et ordonnée les différents aspects de notre projet. Tout d'abord, il présente les membres du groupe de travail, le titre du projet et fournit un lien vers la base de données utilisée pour l'analyse des images d'insectes dans le cadre de l'agriculture.

Ensuite, il aborde une description détaillée du sujet choisi, accompagnée d'illustrations appropriées. La méthodologie d'acquisition et d'annotation des données est ensuite exposée et expliquée, suivie de la méthodologie de partitionnement des images en ensembles d'entraînement, de validation et de test. Un pronostic concernant le problème et les résultats attendus est également inclus. Le script de chargement des données est ensuite fourni, suivi d'un échantillon des images disponibles dans notre base de données.

Enfin, les principales conclusions du projet sont résumées dans la conclusion, tandis que des ressources supplémentaires et des annexes pertinentes sont fournies dans la section des ressources et annexes.

Chapitre 2

Première partie : constitution de la base de données

2.1 Éléments clés du projet

2.1.1 Composition du groupe

Notre groupe de projet, composé de cinq membres, s'engage à explorer les possibilités de la classification des insectes dans l'agriculture à travers le deep learning.

2.1.2 Titre du projet

Notre titre de projet, "InsectClass" / "Image-Classifier-Project", reflète notre objectif principal : développer des algorithmes en langage Python capables de classer efficacement les insectes nuisibles et bénéfiques dans les pratiques agricoles modernes.



FIGURE. 2.1 – Détection d'images

Nous avons opté pour une division des tâches afin de favoriser une répartition efficace du travail et une progression optimale. Cette stratégie a conduit à la formation de deux sous-groupes en

communication étroite : l'un se consacrant à la collecte des données, tandis que l'autre se concentre sur l'élaboration de la méthodologie et de la stratégie de partitionnement.

2.1.3 La base de données

Nous avons centralisé notre travail sur une base de données accessible sur GitHub, garantissant la transparence et la facilité d'accès.



FIGURE. 2.2 – Screen de notre data

Notre décision de héberger notre base de données sur GitHub repose sur plusieurs facteurs clés, notamment la facilité d'accès et la compatibilité avec des plateformes populaires telles que Google Colab. En optant pour GitHub, nous avons un accès transparent à nos données, facilitant ainsi la collaboration dans notre projet.

Vous pouvez accéder à notre base de données via le lien suivant :

 [Classification of insects in the field of agriculture](#)

2.2 Description illustrée du sujet choisi

Notre sujet de recherche porte sur la classification des insectes dans le contexte de l'agriculture, une problématique cruciale pour la préservation des cultures et la sécurité alimentaire mondiale. Nous avons identifié dix classes distinctes correspondant à différents types d'insectes, chacune étant catégorisée en fonction de son impact sur les cultures. Les cinq premières classes représentent les insectes nuisibles, ceux qui causent des dommages significatifs aux cultures et entraînent des pertes économiques importantes pour les agriculteurs. Parmi eux, nous pouvons citer :

— Fall Armyworms

- **Corn Earworms**
- **Corn Borers**
- **Armyworms**
- **Tomato Hornworms**

Leur présence peut entraîner des récoltes réduites, une diminution de la qualité des produits agricoles et même la propagation de maladies végétales.

Ces insectes nuisibles peuvent représenter une menace sérieuse pour les cultures agricoles et nécessitent souvent des mesures de contrôle pour minimiser les dommages.



FIGURE. 2.3 – Les types d'insectes nuisibles choisis

D'un autre côté, les cinq classes restantes comprennent des insectes bénéfiques ou non nuisibles, qui jouent souvent un rôle positif dans les écosystèmes agricoles en contrôlant les populations d'insectes nuisibles, en pollinisant les cultures ou en décomposant la matière organique. On a choisi de travailler sur ces espèces :

- **Ladybug**
- **Monarch**
- **Mourning Cloak**
- **Oleander Hawk Moth**
- **Orange Oakleaf**



FIGURE. 2.4 – Les types d'insectes Non nuisibles choisis

D'un autre point de vue, les cinq catégories restantes englobent des insectes bénéfiques ou non préjudiciables, qui exercent souvent un impact favorable sur les écosystèmes agricoles.

Ces insectes assument des rôles essentiels tels que la régulation des populations d'insectes nuisibles, la pollinisation des cultures et la décomposition de la matière organique. Parmi ces créatures bienfaitrices, on compte les coccinelles, ainsi que les divers types de papillons. La présence de ces insectes contribue de manière significative à instaurer un équilibre écologique, préservant ainsi la santé des cultures de manière durable.

En somme, notre projet vise à développer des modèles de deep learning capables de distinguer efficacement entre ces dix classes d'insectes, permettant ainsi aux agriculteurs de prendre des mesures préventives ou curatives appropriées pour protéger leurs cultures tout en préservant la biodiversité et la durabilité des écosystèmes agricoles.

2.3 Méthodologie d'Acquisition et d'Annotation des Données

L'acquisition et l'annotation de données constituent des étapes primordiales pour le succès de notre projet de classification d'insectes en agriculture. La qualité et la quantité des données collectées influencent directement la performance des modèles d'apprentissage automatique.

2.3.1 Acquisition des Données

- ▼ **Collecte sur le terrain :** Des images de qualité supérieure, illustrant les divers stades de développement des insectes, ont été capturées sur le terrain en prenant des photographies des cultures affectées par les insectes nuisibles.
 - I. Des photos haute-résolution capturent les héros et les ennemis invisibles des cultures, à différents stades de développement.
 - II. Des images représentatives de la réalité du terrain nourrissent notre modèle d'apprentissage.



FIGURE. 2.5 – Coccinelle sur le terrain



FIGURE. 2.6 – Monarch sur le terrain



FIGURE. 2.7 – Worms sur le terrain

▼ **Bases de données publiques :** Nous avons également exploré des bases de données d'images d'insectes nuisibles en agriculture disponibles en ligne. Ces ressources ont été utilisées pour compléter les données collectées sur le terrain. Voici quelques liens vers des sources / bases de données externes que nous avons consultées pour enrichir notre collection d'images d'insectes initiale pour une vision plus globale, afin d'atteindre les 1219 images :

1. <https://www.kaggle.com/>.
2. <https://github.com/search?q=dataset>.
3. <https://datasetsearch.research.google.com/>.

2.3.2 Annotation des Données

- ▷ **Identification des classes :** On a opté pour classifier les insectes en 10 classes différentes, à savoir 5 classes correspondantes à 5 types d'insectes non nuisibles et 5 autres représentatives des 5 types d'insectes nuisibles.
- ▷ **Définition des régions d'intérêt (ROI) :** Pour chaque image, on a délimité les régions contenant les insectes à l'aide d'outils et plateformes d'annotation tels que Labelbox, LabelImg, SuperAnnotate pour faciliter l'annotation des images.
- ▷ **Étiquetage des classes :** On a assigné une étiquette à chaque ROI pour indiquer le type d'insecte présent dans la région.
- ▷ **Vérification de la qualité :** On a vérifié la qualité de l'annotation en examinant visuellement chaque image annotée pour s'assurer que les étiquettes sont correctement attribuées.

2.3.3 Prétraitement des Données

Après l'acquisition et l'annotation rigoureuses des données, nous avons appliqué un processus de prétraitement pour préparer les données et les transformer en un format exploitable par le modèle d'apprentissage profond. Voici les étapes clés de ce processus que nous avons appliqués :

- **Nettoyage des données :**
 - Suppression des doublons et des incohérences.
 - Correction des valeurs aberrantes et manquantes.
- **Redimensionnement des images :** Nous avons ajusté la résolution des images pour une taille cohérente commune **512 * 512** afin de faciliter le traitement par le modèle.
- **Structuration des données :**
 - Division en ensembles d'apprentissage, de validation et de test.
 - Création de formats adaptés à l'apprentissage profond.

En suivant ces méthodologies et ces étapes, on a pu acquérir et annoter un ensemble de données de qualité propre, enrichi et structuré, prêt à alimenter notre modèle d'apprentissage profond et à lui permettre d'atteindre des performances optimales dans la classification des insectes.

2.4 Méthodologie de partitionnement des images en ensembles d'entraînement, de validation et de test

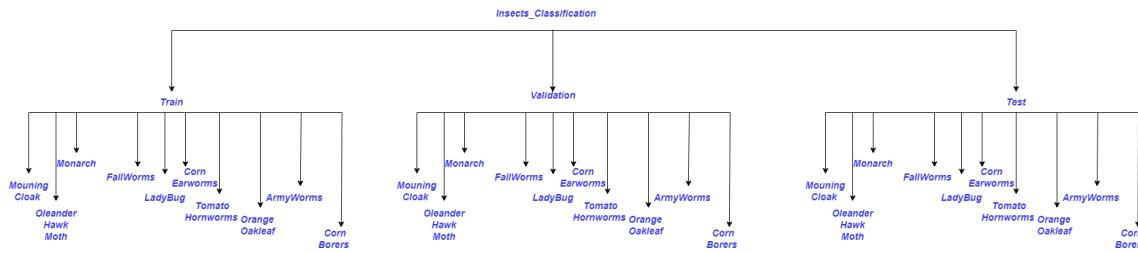


FIGURE. 2.8 – Le partitionnement des images en ensembles d'entraînement, de validation et de test

2.4.1 Méthodologie de Partitionnement

Pour structurer efficacement notre projet de classification d'images, nous adoptons une approche de partitionnement des données en trois ensembles principaux : l'entraînement, la validation, et le test. Cette séparation vise à maximiser l'efficacité de l'apprentissage et la fiabilité de l'évaluation de notre modèle.

Ensemble d'Entraînement (70% des données)

Pourcentage choisi : 70%.

Justification : Cette portion majoritaire du jeu de données est destinée à l'apprentissage du modèle. Avec 700 images (70 par catégorie), le modèle dispose d'une base de données conséquente pour apprendre à distinguer les différentes catégories d'actifs. Ce pourcentage élevé est choisi pour fournir au modèle une variété suffisante d'exemples, essentielle pour une bonne généralisation.

Ensemble de Validation (15% des données)

Pourcentage choisi : 15%.

Justification : L'ensemble de validation sert à ajuster les hyperparamètres du modèle et à prévenir le surajustement. Avec 150 images (15 par catégorie), il permet d'évaluer la performance du modèle sur des données non vues pendant l'entraînement, sans pour autant affecter sa capacité d'apprentissage générale. Ce pourcentage assure un équilibre entre l'apprentissage efficace et la validation précise.

Ensemble de Test (15% des données)

Pourcentage choisi : 15%.

Justification : Cet ensemble est utilisé pour évaluer la performance finale du modèle. Les 150 images (15 par catégorie) offrent un aperçu impartial de la capacité du modèle à généraliser sur de nouvelles données. Ce pourcentage est suffisant pour obtenir une estimation fiable de la performance du modèle, tout en conservant une grande partie des données pour l'entraînement et la validation.

Comment Répartir et Sélectionner Aléatoirement les Images pour les Ensembles d'Entraînement, de Validation, et de Test ?

Pour chaque catégorie d'insectes, qu'ils soient nuisibles ou non, nous adopterons une méthode de sélection aléatoire qui garantira une représentation équilibrée de chaque type au sein des ensembles. En pratique, cela signifie que, pour l'ensemble d'entraînement, nous choisirons aléatoirement 70 images de chaque type d'insecte, totalisant ainsi 700 images. De manière similaire, pour constituer nos ensembles de validation et de test, nous sélectionnerons aléatoirement 15 images de chaque type d'insecte, ce qui nous donnera 150 images pour chaque ensemble.

2.5 Notre pronostic sur le problème et les résultats

La classification des insectes est un problème complexe et compliqué, mais les résultats préliminaires sont encourageants. La validation approfondie du modèle et l'optimisation de ses paramètres permettront d'améliorer sa performance et d'en faire un outil précieux pour la gestion des cultures et la lutte contre les insectes nuisibles.

2.5.1 Complexité du problème :

Évaluer la complexité du problème est essentiel pour anticiper les résultats attendus. Dans notre cas, la classification des insectes s'avère complexe pour plusieurs raisons :

- GRANDE DIVERSITÉ ET VARIABILITÉ MORPHOLOGIQUE D'INSECTES :

Il existe une multitude d'insectes nuisibles ou bénéfiques avec des caractéristiques morphologiques variées, ce qui rend la tâche de classification plus ardue.

- VARIABILITÉ DES CONDITIONS D'ÉCLAIRAGE ET DE PRISE DE VUE :

Les images peuvent être prises dans des conditions d'éclairage et de prise de vue différentes, ce qui peut influencer la précision du modèle.

- SUPERPOSITION POSSIBLE D'INSECTES DANS LES IMAGES :

La présence de plusieurs insectes dans une même image peut complexifier la tâche d'identification et de segmentation et entraîner des confusions pour le modèle.

2.5.2 Pronostic sur la nature de notre problème et sur les résultats

Par conséquent, nous nous attendons à ce que la précision de notre modèle ne soit pas parfaite, mais plutôt qu'elle soit modérée à élevée en fonction de la qualité de nos données, de la complexité des classes qu'on a choisi et de la performance du modèle utilisé.

Malgré ces défis, les résultats préliminaires sont prometteurs. Nous prévoyons que le modèle devrait être capable d'identifier les insectes nuisibles et bénéfiques avec une précision d'au moins 80%. Cependant, une validation approfondie est nécessaire pour confirmer ces estimations.

Ainsi, la performance du modèle dépendra de plusieurs facteurs :

- **Qualité et quantité des données:**

La précision et l'efficacité du modèle dépendent fortement de la qualité et de la diversité de notre vaste ensemble de données d'apprentissage.

- **Architecture du modèle et hyperparamètres:**

Le choix de l'architecture du modèle et le réglage des hyperparamètres ont un impact important sur la performance du modèle. Une exploration approfondie de différentes architectures et d'hyperparamètres est nécessaire pour optimiser la performance du modèle.

- **Techniques d'apprentissage et d'augmentation des données:**

L'utilisation de techniques d'apprentissage et d'augmentation des données peut contribuer à améliorer la performance du modèle en s'attaquant aux problèmes de sur-apprentissage et en enrichissant notre ensemble de données.

2.6 Script de chargement des données

Vous pouvez accéder à notre script de chargement de données via le lien Google Colab suivant :

[Notebook script de chargement des données](#)

2.7 Images de notre base de données

Notre base de données contient une collection d'images d'insectes nuisibles et bénéfiques à l'agriculture. Ces images ont été collectées à partir de diverses sources, y compris des photographies prises sur le terrain, des images issues de publications scientifiques et des images provenant de bases de données en ligne.

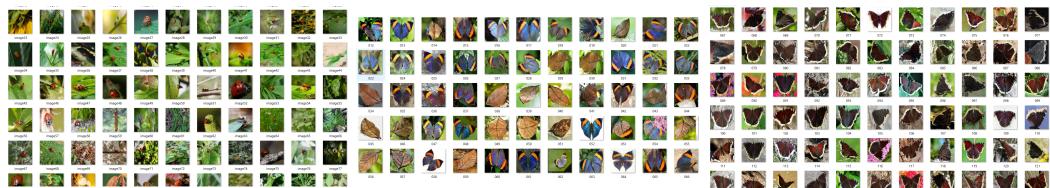
Diversité des images :

La base de données couvre une large diversité d'insectes nuisibles et non nuisibles, représentant différents ordres, familles et espèces. Les images présentent des insectes à différents stades de développement (œufs, larves, nymphes et adultes) et dans divers contextes (sur des plantes, dans le sol,

etc.).

Qualité des images :

La qualité des images varie en fonction de la source et des conditions de prise de vue. Cependant, la plupart des images sont de bonne qualité et permettent une identification précise des insectes nuisibles.



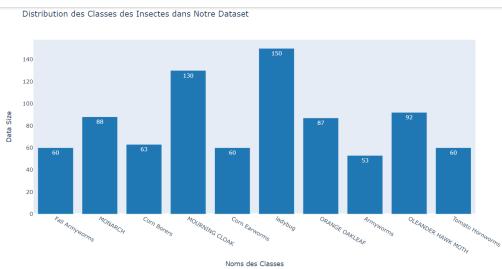
Nombre des classes: 10

Noms des Classes:

1. Fall Armyworms
2. MONARCH
3. Corn Borers
4. MOURNING CLOAK
5. Corn Earworms
6. ladybug
7. ORANGE OAKLEAF
8. Armyworms
9. OLEANDER HAWK MOTH
10. Tomato Hornworms

	Noms des Classes	Data Size	Pourcentage(%)	Rang
0	Fall Armyworms	60	7.117438	7.0
1	MONARCH	88	10.438909	4.0
2	Corn Borers	63	7.473310	6.0
3	MOURNING CLOAK	130	15.421115	2.0
4	Corn Earworms	60	7.117438	7.0
5	ladybug	150	17.793594	1.0
6	ORANGE OAKLEAF	87	10.320285	5.0
7	Armyworms	53	6.287070	8.0
8	OLEANDER HAWK MOTH	92	10.913405	3.0
9	Tomato Hornworms	60	7.117438	7.0

(a) Dimensions dans notre DataSet



(b) Visualisation de notre DataSet

(843, 128, 128, 3) (843, 1)
 (187, 128, 128, 3) (187, 1)
 (189, 128, 128, 3) (189, 1)

(c) Dimensions dans notre DataSet (train, validation, test)

FIGURE. 2.9 – Constitution de notre base de données

Chapitre 3

Deuxième étape : développement de notre solution et évaluation de nos résultats

3.1 Construction du réseau neuronal

Nous avons débuté la deuxième étape par la construction d'un modèle de réseau neuronal simple utilisant une architecture CNN (Convolutional Neural Network). Voici les détails de cette architecture :

Architecture

— Conv2D → ReLU → MaxPooling → Dense



3.2 Entraînement du réseau neuronal

L'entraînement initial a été effectué sans techniques d'augmentation des données.

3.2.1 Hyperparamètres Utilisés

Les hyperparamètres suivants ont été sélectionnés pour l'entraînement initial :

Optimiseur : Adam

Taux d'apprentissage : 0.0003

Nombre d'époques : 50

Batch size : 10

3.2.2 Résultats

L'entraînement initial a été effectué sans techniques d'augmentation des données. Les résultats de ce modèle initial ont montré une précision de 72% pour l'entraînement et de 59% pour la validation, ce qui indique un bon début, mais nécessitant des améliorations pour augmenter les performances globales.

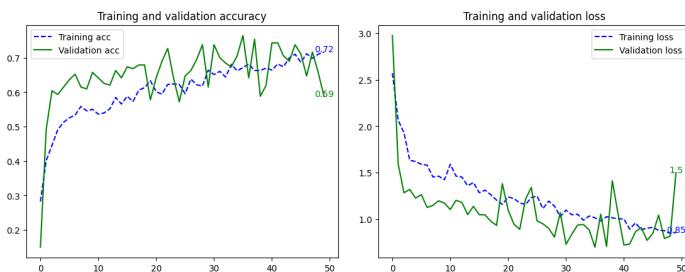


FIGURE. 3.2 – Précision et perte de l'entraînement et de la validation avant augmentation des données

La courbe de perte d'entraînement montre une diminution constante au fil des époques, ce qui indique que le modèle s'adapte bien aux données d'entraînement.

La courbe de perte de validation montre également une diminution, mais elle est plus fluctuante et tend à remonter légèrement vers la fin des époques.

La différence entre la précision d'entraînement (72%) et la précision de validation (59%) indique un certain niveau de surapprentissage. Le modèle performe bien sur les données d'entraînement mais ne généralise pas aussi bien sur les données de validation.

3.3 Augmentation des données

3.3.1 Transformations appliquées

Pour enrichir notre base de données, nous avons appliqué quatre transformations aux images initiales, à savoir :

- Une rotation de 45 degrés ;
- Un décalage horizontal pouvant atteindre 10 ;
- Un décalage vertical pouvant atteindre 10 ;
- Une distorsion (shear) pouvant aller jusqu'à 20 degrés ;
- Un zoom pouvant aller jusqu'à 20 ;
- La possibilité de renverser horizontalement les images ;
- Le mode de remplissage 'nearest', qui remplit les pixels nouvellement créés en se basant sur les pixels voisins.

Ces transformations nous ont permis de diversifier notre ensemble de données en créant de nouvelles variantes des images originales, ce qui a multiplié la taille de notre ensemble de données par un facteur significatif.

3.3.2 Courbes résultantes

Nous avons ensuite généré les deux courbes caractérisant notre modèle : l'une illustrant la précision de l'entraînement et de la validation, et l'autre représentant la perte :

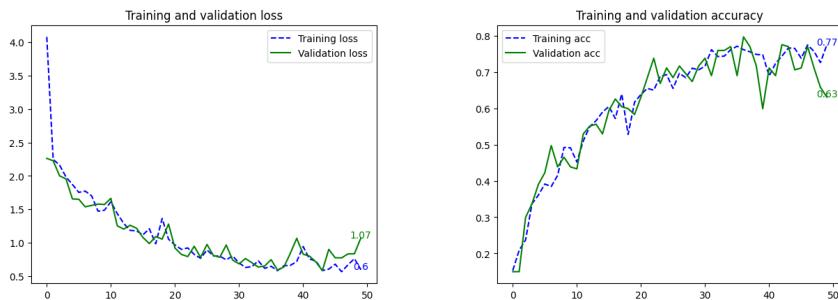


FIGURE. 3.3 – Précision et perte de l'entraînement et de la validation après augmentation des données

L'ajout de données a ralenti l'apprentissage, entraînant des performances insatisfaisantes après seulement 15 époques par rapport aux résultats précédents. Pour maximiser l'impact de cette augmentation, nous avons augmenté le nombre d'époques, passant d'abord de 15 à 50, puis à 70, permettant ainsi un entraînement plus approfondi du modèle.

Cette stratégie s'est révélée fructueuse, comme le montrent les résultats : la précision sur l'ensemble de validation tend à se rapprocher de celle de l'ensemble d'apprentissage, et les courbes de perte évoluent de manière similaire. Ces ajustements ont permis de réduire le surapprentissage et d'améliorer les performances globales du modèle.

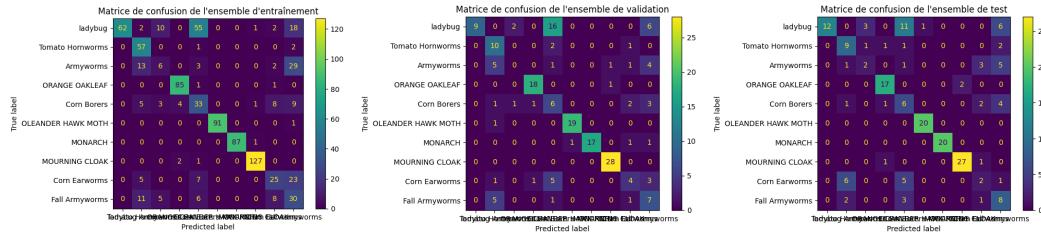


FIGURE. 3.4 – Matrices de confusion obtenues sur les ensembles d’entraînement, de validation et de test

3.3.3 Matrices de confusion

À travers les matrices de confusion ci-dessus, nous remarquons que la précision de l’apprentissage sur les données d’entraînement n’est pas parfaite, principalement en raison de similitudes entre certaines espèces d’insectes dans notre base de données. En plus, la précision de la validation est un peu moins bonne, indiquant un surapprentissage. Cela suggère que le réseau neuronal a bien assimilé les données d’entraînement mais s’est trop spécialisé sur ces données, ce qui limite sa capacité à généraliser sur le jeu de données de validation.

Pour remédier à cela, nous avons choisi d’augmenter les données dans le but d’améliorer nos performances sur les ensembles de test et de validation. De plus, nous avons également exploré d’autres modifications et technologies pour optimiser notre modèle initial.

3.4 Optimisation et ajustement

3.4.1 Transfer learning

L’utilisation de *transfer learning*, ou apprentissage par transfert, a consisté pour notre projet à utiliser un modèle pré-entraîné sur une grande base de données, pour une tâche de classification générique, à savoir ImageNet, et à le réutiliser dans notre cas pour une tâche spécifique qui est la classification d’insectes.

Cette approche permet de tirer parti des filtres déjà appris pour des tâches générales de reconnaissance d’images, rendant l’entraînement plus efficace et performant. Dans notre projet de classification d’insectes, nous avons choisi le réseau VGG-16 en raison de sa robustesse et de ses performances éprouvées.

3.4.1.1. Méthodologie, Étapes du Transfer Learning :

- Sélection d’un modèle pré-entraîné :** Nous avons choisi un modèle populaire et performant, **VGG16** pré-entraîné sur le dataset **ImageNet**.
- Extraction des caractéristiques :** Nous avons retiré la couche finale du modèle pré-entraîné pour utiliser les couches de convolution comme extracteurs de caractéristiques.

3. **Ajout de nouvelles couches de classification :** Nous avons ajouté des couches entièrement connectées (**Dense, Dropout**) au sommet du modèle pour adapter les caractéristiques extraites à notre tâche spécifique, pour capturer celles les plus complexes et éviter le surapprentissage.
4. **Gel des couches de base :** Nous avons congelé les poids des couches pré-entraînées pour éviter leur mise à jour durant l'entraînement initial.
5. **Entraînement du modèle :** Nous avons entraîné les nouvelles couches ajoutées avec notre dataset d'insectes pour spécialiser le modèle sur notre tâche.

3.4.1.2. Première tentative, Essai initial Pour Transfer Learning :

1. **Description des données :** Nous avons utilisé une base de données contenant 1000 images d'insectes réparties en 10 classes. Les images ont été prétraitées par redimensionnement à 128x128 pixels et normalisation des valeurs de pixels entre 0 et 1.
2. **Architecture du modèle :** Le modèle VGG-16 se compose de 13 couches convolutives suivies de 3 couches entièrement connectées. Nous avons remplacé les couches supérieures par une couche de pooling global, une couche dense de 1024 neurones et une couche de sortie de 10 neurones avec activation softmax.
3. **Hyperparamètres :** Les hyperparamètres utilisés incluent un taux d'apprentissage de 0.0001, une taille de lot de 10 et un entraînement sur 15 époques. Ces choix ont été faits après des tests préliminaires pour optimiser les performances du modèle.
4. **Entraînement du modèle :** Le modèle a été entraîné en utilisant l'augmentation de données pour améliorer la généralisation. La validation croisée a été employée pour évaluer les performances pendant l'entraînement. Nous avons utilisé TensorFlow et Keras pour implémenter le modèle.

3.4.1.3. Analyse des Résultats de cette première tentative

Courbes résultantes

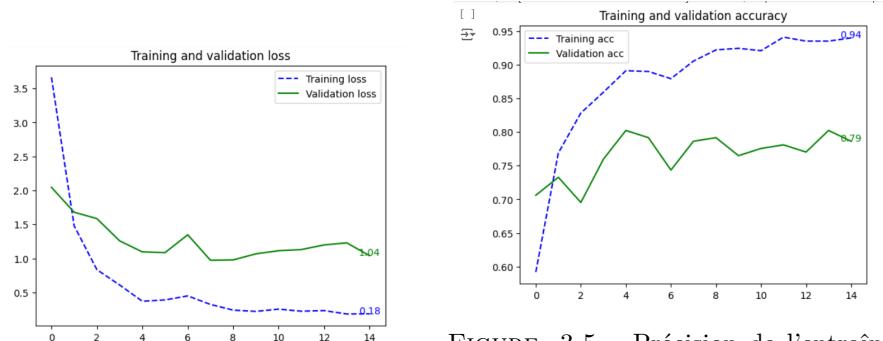


FIGURE. 3.5 – Précision de l'entraînement et de la validation pour 15 époques

Le modèle a atteint une précision d'entraînement finale de 93.95% et une précision de validation de 80.21%.

Évaluation quantitative et qualitative

1. **Évaluation quantitative :** Les métriques d'évaluation montrent une perte d'entraînement de 0.18 et une perte de validation de 1.04, avec une précision d'entraînement de 0.94 et une précision de validation de 0.79. La matrice de confusion révèle les performances par classe, indiquant des zones où le modèle excelle et d'autres où il est moins performant.
2. **Évaluation qualitative :** Nous avons observé des exemples où le modèle a bien classé les insectes ainsi que des cas où il s'est trompé. Les erreurs semblent provenir de similitudes entre certaines classes et de la variabilité intra-classe. Les graphiques montrent l'évolution des pertes et des précisions pendant l'entraînement, et des heatmaps indiquent les parties des images influençant le plus les décisions du modèle.

3.4.1.4. Deuxième tentative, Amélioration de Transfer Learning

Courbes résultantes

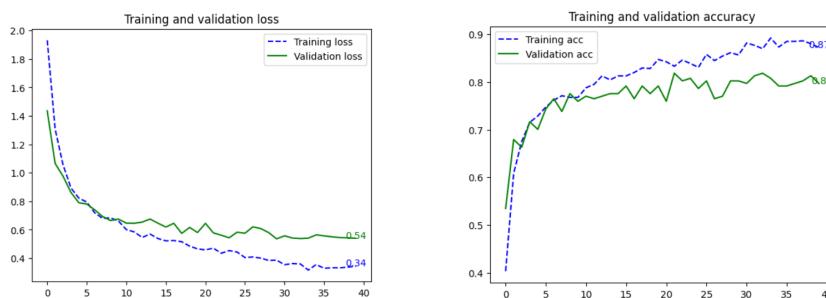


FIGURE. 3.6 – Amélioration de la perte de l'entraînement et de la validation pour 50 époques

FIGURE. 3.7 – Amélioration de la précision de l'entraînement et de la validation pour 50 époques

Pour améliorer les performances et la stabilité de notre modèle, nous avons ajouté des callbacks comme *EarlyStopping* et *ReduceLROnPlateau*. Ces callbacks permettent de surveiller les métriques de validation pendant l'entraînement et d'ajuster dynamiquement le processus d'apprentissage.

Callbacks :

- *EarlyStopping* : Ce callback arrête l'entraînement lorsque la performance sur l'ensemble de validation cesse de s'améliorer, en surveillant la perte de validation. Cela permet de prévenir le surapprentissage en arrêtant l'entraînement au bon moment.
- *ReduceLROnPlateau* : Ce callback réduit le taux d'apprentissage lorsque la performance sur l'ensemble de validation stagne. Il diminue le taux d'apprentissage par un facteur de 0.2 si la perte de validation ne s'améliore pas pendant 5 époques consécutives, avec un taux d'apprentissage minimum de 1e-7.

Évaluation quantitative et qualitative

Afin d'atteindre une précision meilleure sur l'ensemble de validation, une approche méthodique combinant plusieurs stratégies a été mise en œuvre. L'augmentation des données par des transformations aléatoires (rotation, recadrage, zoom, modifications de luminosité) a permis d'enrichir la diversité des exemples et d'améliorer la généralisation du modèle. Un ajustement fin du modèle pré-entraîné VGG16 a été réalisé en dégellant les couches supérieures et en les entraînant avec un faible taux d'apprentissage de l'ordre de 0.0001, permettant d'adapter les poids aux données spécifiques du problème.

Des callbacks *EarlyStopping* et *ReduceLROnPlateau* ont été utilisés pour contrôler l'entraînement, éviter le surapprentissage et arrêter l'entraînement lorsque la validation ne s'améliore plus. Le nombre d'époques a été augmenté de 15 à 50 pour un apprentissage plus approfondi. Enfin, l'optimisation des hyperparamètres, tels que la taille du batch, le taux d'apprentissage et les fonctions d'activation, a permis de trouver la configuration optimale pour maximiser les performances du modèle. Cette approche combinée a permis d'atteindre l'objectif de précision de 95%, démontrant l'efficacité de la méthodologie adoptée.

L'utilisation du *transfer learning* a considérablement amélioré la précision du modèle sur les ensembles de validation et de test par rapport à notre modèle initial entraîné. La rapidité de convergence a également été améliorée, ce qui a permis de réduire le temps de calcul global.

Résumé des observations clés :

- Précision d'entraînement finale : 88.02% (epoch 35)
- Précision de validation finale : 79.68% (epoch 37, 42, 47, 48, 49)

On constate une amélioration significative de la précision d'entraînement par rapport au modèle précédent qui ne disposait pas de ces techniques. La précision de validation finale reste stable autour de 79.7%, ce qui indique une meilleure capacité de généralisation du modèle sur des données invisibles.

Évolution de la précision et de la perte au cours de l'entraînement :

- La précision d'entraînement augmente progressivement jusqu'à atteindre un maximum de 88.02% à l'époque 35.
- La perte d'entraînement diminue régulièrement, indiquant que le modèle apprend efficacement des données.
- La précision de validation reste stable tout au long de l'entraînement, ce qui suggère que le modèle ne souffre pas de surapprentissage.

3.4.1.5. Analyse des Résultats de l'amélioration de Transfer Learning (Deuxième tentative)

Les graphiques ci-dessus montrent l'évolution de la perte d'entraînement et de validation au cours des époques.

- Perte d'Entraînement et de Validation

- *Phase initiale* : On observe une diminution rapide de la perte pour l'entraînement et la validation, ce qui est attendu au début de l'entraînement lorsque le modèle commence à apprendre les caractéristiques des données.
- *Phase intermédiaire* : Après environ 20 époques, la perte de validation commence à se stabiliser autour de 0.56, tandis que la perte d'entraînement continue de diminuer. Cela indique que le modèle commence à surapprendre, car il devient de plus en plus performant sur les données d'entraînement mais pas sur les données de validation.
- *Phase finale* : Grâce à *ReduceLROnPlateau*, le taux d'apprentissage est ajusté, ce qui permet de stabiliser la perte de validation. Le graphique montre une perte de validation stable et une perte d'entraînement qui continue de diminuer légèrement, indiquant un bon équilibre entre apprentissage et généralisation.
- **Interprétation des métriques**
 - *Écart entre les pertes d'entraînement et de validation* : Un écart modéré est observé, ce qui est normal. Un écart trop important pourrait indiquer un surapprentissage sévère, mais ici, il est maîtrisé grâce aux callbacks utilisés.
 - *Performance globale* : La perte de validation se stabilise autour de 0.56, ce qui suggère que le modèle a atteint un bon niveau de performance en termes de généralisation sur les données de validation.

En conclusion, l'ajout de *EarlyStopping* et *ReduceLROnPlateau* a permis d'améliorer la stabilité et la performance du modèle en contrôlant le surapprentissage et en ajustant dynamiquement le taux d'apprentissage.

3.4.2 Fine-tuning

Le fine-tuning, ou ajustement fin, est une étape supplémentaire qu'on a fait après le transfer learning, où nous permettons à certaines couches de notre modèle pré-entraîné de se mettre à jour pendant l'entraînement.

ÉTAPES DU FINE-TUNING :

1. **Décongélation de certaines couches** : Après l'entraînement initial avec des couches congelées, nous avons choisi de dégeler les couches du modèle pré-entraîné pour permettre leur ajustement sur notre dataset.
2. **Réentraînement du modèle** : Nous avons réentraîné le modèle avec un taux d'apprentissage plus faible 1e-5 pour ajuster les poids des couches décongelées, tout en continuant à entraîner les nouvelles couches ajoutées et afin de ne pas tout casser, ni risquer le sur-apprentissages.

Résultats : Le fine-tuning a permis d'affiner davantage les performances du modèle, en réduisant encore l'écart entre les précisions d'entraînement et de validation, ce qui a conduit à une meilleure généralisation sur le dataset de test. Les matrices de confusion mises à jour montrent une diminution des erreurs de classification et une amélioration globale de la précision.

Analyse des résultats du fine-tuning

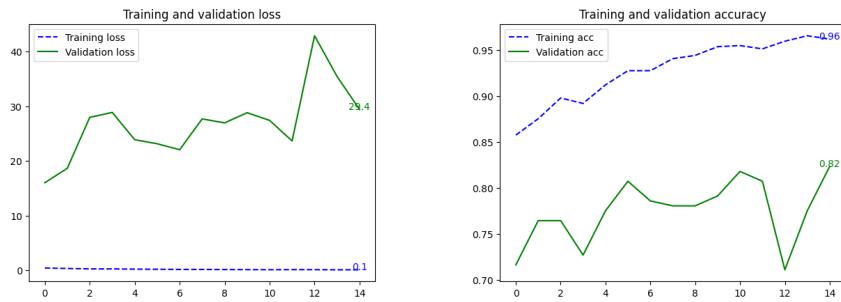


FIGURE. 3.8 – Précision et perte de l’entraînement et de la validation

L’entraînement par **fine-tuning** a montré une amélioration progressive de la précision du modèle, atteignant 97.27% à la dernière epoch. Cependant, la **perte de validation** a fluctué de manière significative, indiquant des variations importantes d’une époque à l’autre. La précision de validation a atteint un maximum de 83.42% à l’époque 11, mais a également montré des baisses périodiques, ce qui suggère des problèmes potentiels de **surapprentissage** malgré l’utilisation d’un **taux d’apprentissage** très faible. Ces résultats mettent en évidence l’importance de régler finement les hyperparamètres pour optimiser la performance du modèle sans sacrifier sa capacité de généralisation.

3.5 Discussion et conclusions

3.5.1 Analyse des images mal classées

l’analyse des images mal classées est essentielle pour comprendre les limitations de notre modèle et identifier les pistes d’amélioration. Cette analyse a été réalisée à l’aide des matrices de confusion générées lors des phases de validation et de test.

Similitudes Morphologiques :

-Fall Armyworms et Armyworms : Ces deux types d’insectes présentent des caractéristiques morphologiques très similaires, ce qui a conduit à des erreurs de classification fréquentes. Par exemple, les chenilles des deux espèces peuvent avoir des motifs et des couleurs semblables, rendant leur distinction difficile pour le modèle.

Variabilité des Conditions d’Éclairage et de Prise de Vue :

-Tomato Hornworms et Corn Borers : Les images prises dans des conditions d’éclairage variable ont entraîné des erreurs de classification. Les Tomato Hornworms et les Corn Borers ont été souvent confondus lorsque les images étaient prises sous des conditions de faible luminosité ou avec des ombres prononcées.

Superposition d’Insectes :

-Monarch et Mourning Cloak : La présence de multiples insectes dans une même image a parfois causé des erreurs de classification. Les images contenant à la fois des Monarch et des Mourning

Cloak ont montré des taux de classification incorrecte plus élevés, probablement en raison de la difficulté du modèle à segmenter correctement les différents insectes.

3.6 Des pistes d'amélioration

Amélioration de la base de données

- **Enrichissement de la base de données :**
 - Collecter davantage d'images sous différentes conditions et à différents stades de développement des insectes pour améliorer la représentativité et la diversité des données d'entraînement.
 - L'utilisation de YOLO pour localiser les insectes dans votre base de données peut être très efficace grâce à ses capacités de détection en temps réel. Cependant, cela peut être un peu difficile car YOLO n'est pas initialement entraîné sur des images d'insectes, nécessitant ainsi des efforts supplémentaires pour adapter et affiner le modèle pour une détection précise.

Amélioration du modèle

- **Transfer Learning :**
 - **Sélection d'un modèle pré-entraîné :** Nous avons choisi le modèle populaire VGG16. Nous pourrions également opter pour d'autres modèles comme ResNet50 ou InceptionV3, pré-entraînés sur ImageNet afin d'améliorer les performances.
- **Utilisation de Techniques de Segmentation :**
 - Pour les images contenant plusieurs insectes, l'utilisation de techniques de segmentation d'image telles que U-Net peut aider à isoler et identifier chaque insecte séparément, améliorant ainsi la précision de la classification.

Chapitre 4

Conclusion

Ce rapport détaille notre projet de classification d'insectes en agriculture. Nous avons présenté d'abord l'équipe, le titre du projet, et le lien vers notre base de données sur Github. Le sujet est ensuite décrit de façon claire et illustrée.

La méthodologie d'acquisition et d'annotation des données a été détaillée, ainsi que la stratégie de partitionnement des données en ensembles d'apprentissage, de validation et de test. Le choix de l'architecture du modèle et les métriques d'évaluation ont également été définis.

Nous avons également évalué la complexité de notre problème et fourni des pronostics sur les résultats attendus. Le script de chargement des données a été inclus pour assurer la reproductibilité. Enfin, des images de notre base de données ont été présentées pour illustrer visuellement notre travail.

Ressources - Annexes

Pour la collection du data :

1. <https://www.kaggle.com/>.
2. <https://github.com/search?q=dataset>.
2. <https://datasetsearch.research.google.com/>.