CLASSIFICATION DES MALADIES DES FEVES DE CACAO EN UTILISANT LES ALGORITHMES CONVNEXT ET SVM

AMADOU BAH, FREDERIC AKADJE

Introduction

Le cacao, culture emblématique de la Côte d'Ivoire, constitue un pilier central de l'économie nationale, contribuant à près de *15 %* du produit intérieur brut (PIB) et représentant environ *40 %* des exportations du pays. Premier producteur mondial avec plus de 2 millions de tonnes annuelles, la Côte d'Ivoire assure ainsi la subsistance de près de 6 millions de personnes, dont la majorité sont des petits producteurs ruraux. Cependant, cette filière stratégique est confrontée à des défis majeurs, notamment les maladies des fèves de cacao, qui entraînent des pertes de rendement pouvant atteindre *30 à 40 %* selon l'Organisation Internationale du Cacao (ICCO). Ces pertes compromettent non seulement la sécurité économique des agriculteurs, mais aussi la stabilité d'un secteur vital pour l'économie ivoirienne.

Parmi les pathologies les plus dévastatrices figurent la pourriture brune des cabosses (causée par Phytophthora spp.), la maladie du swollen shoot (due à un virus transmis par des cochenilles) et la moniliose, responsables de dégâts irréversibles sur les plantations. Ces maladies, souvent difficiles à identifier précocement à l'œil nu, se propagent rapidement sous des conditions climatiques favorables, exacerbées par les changements climatiques. Les méthodes traditionnelles de diagnostic, basées sur une expertise visuelle subjective et sporadique, peinent à offrir des solutions rapides et précises, laissant les producteurs démunis face à ces fléaux.

La littérature scientifique récente a exploré diverses approches technologiques pour améliorer la détection des maladies agricoles. Des travaux pionniers ont employé des algorithmes de traitement d'images (comme les réseaux de neurones convolutifs, CNN) pour classifier des lésions foliaires, tandis que d'autres ont combiné ces méthodes avec des modèles d'apprentissage machine (machine learning) tels que les SVM (Support Vector Machines), réputés pour leur efficacité dans des espaces de grande dimension. Toutefois, ces études se heurtent souvent à des limites pratiques : complexité computationnelle, besoin en larges jeux de données annotées, ou inadéquation aux réalités des terrains africains, où l'accès à des technologies avancées reste limité.

Dans ce contexte, notre étude propose une approche innovante combinant l'architecture ConvNeXt, une évolution récente des CNN optimisée pour l'extraction de caractéristiques visuelles hiérarchiques, et un classifieur SVM, afin de pallier les lacunes des méthodes existantes. ConvNeXt, inspiré des transformers tout en conservant l'efficacité des CNN, permet une capture robuste de motifs morphologiques subtils dans les images de fèves infectées, même sous des résolutions variables ou des conditions d'éclairage hétérogènes. Les caractéristiques extraites sont ensuite traitées par un SVM, dont la capacité à généraliser à partir de petits

ensembles de données convient particulièrement aux contextes où les ressources informatiques et les données annotées sont limitées.

Cette hybridation méthodologique vise à offrir un outil de diagnostic précoce, accessible via des applications mobiles, permettant aux agriculteurs ivoiriens de photographier leurs fèves et de recevoir une prédiction instantanée sur la présence et le type de maladie. En validant cette approche sur un corpus d'images collectées in situ, enrichi par des collaborations avec des agronomes locaux, cette recherche ambitionne de combler un gap critique entre les avancées en intelligence artificielle et les besoins concrets des producteurs. Une détection rapide et fiable pourrait réduire significativement les pertes post-récolte, sécuriser les revenus des ménages ruraux et renforcer la résilience d'un secteur clé pour la Côte d'Ivoire.

Matériels et méthodes

Cette section présente les outils et méthodologies employés dans le cadre des travaux réalisés.

Matériels

Les données utilisées sont 627 images de fèves de Cacao réparties en trois classes, représentant les états de base à détecter. Ainsi que 627 fichiers texte (.txt) représentant les régions d'intérêt de chaque image.

Figure 1: images initiales

Sur chaque image figure plusieurs états de fèves "Fito" (phytophthora), "Monilia" (monilia) et "Sana" (sains). Le phytophthora et le monilia représente deux maladies des fèves de cacao. La figure ci-dessus présente précisément les différents états étudiés.

Figure 2: Etats des différentes fèves étudiées

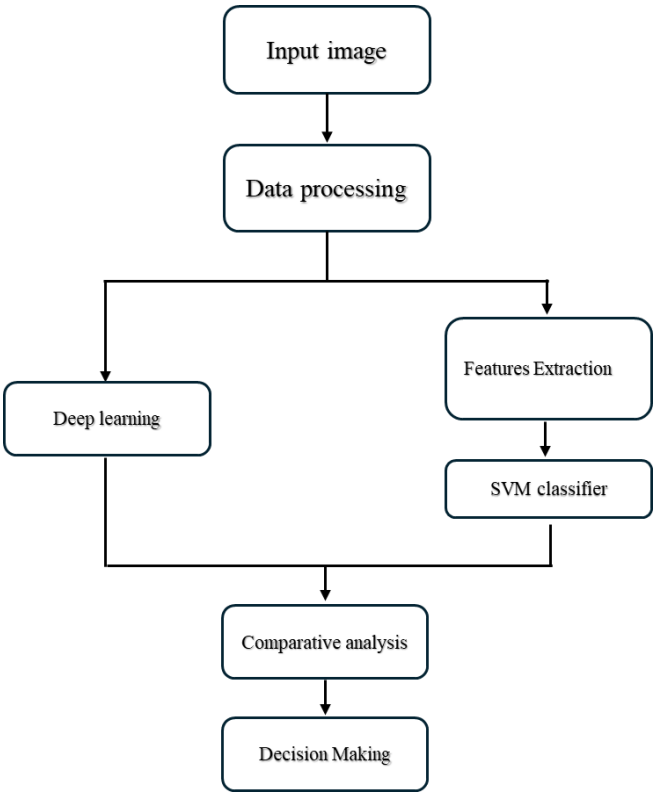
Sain | Phytophthora | Monilia

Phytophthora est un genre de champignons pathogènes qui affectent le cacao (Theobroma cacao). Les espèces de Phytophthora sont responsables de la maladie des cabosses noires du cacao, les symptômes commencent par une tache brune circulaire qui s'étend rapidement pour couvrir toute la cabosse. Quant à la Monilia, elle est causée par est un champignon basidiomycète. Les cabosses infectées développent une couche blanche duveteuse, qui devient ensuite beige ou marron clair. Il s'agit donc deux maladies distinguables à travers l'apparence des fèves.

Méthodes

Ces régions seront utilisées ultérieurement dans le pipeline de données. Chaque image est accompagnée d'un fichier texte (.txt) contenant des annotations sous forme de bounding boxes normalisées, précisant les régions d'intérêt (ROI) correspondant aux zones saines ou infectées. Ces annotations, réalisées par des experts, constituent la base du processus de prétraitement.

Figure 3: Pipeline de données



Pour extraire les régions d'intérêt (ROI) à partir des images annotées, un script Python automatisé a été développé. Ce script convertit les coordonnées normalisées des bounding boxes en coordonnées absolues, en fonction des dimensions de chaque image, puis extrait les zones correspondantes. Ces ROI ont ensuite été sauvegardées sous forme d'images individuelles dans un dossier dédié. Pour garantir l'unicité des noms de fichiers, chaque ROI a été renommée en utilisant un identifiant unique généré via *un hash MD5*[1]. Le format adopté pour les noms de fichiers est ***class{id_class}_{identifiant}.jpg***, où ***id_class*** correspond à l'identifiant numérique de la classe (*0 pour Fito, 1 pour Monilia, 2 pour Sana*), et ***identifiant*** est une chaîne de 8 caractères

Un prétraitement des données a été réalisé pour assurer une compatibilité optimale avec l'architecture ConvNeXt. Les images ont été redimensionnées à ***224 × 224 pixels*** et normalisées pour répondre aux exigences du modèle. Les labels ont été extraits des noms de fichiers et encodés pour permettre une classification multi-classes. Les données ont ensuite été organisées en lots de 32 images à l'entraînement pour une accélération du traitement. Ce processus garantit une préparation efficace des données pour le modèle hybride *ConvNeXt + SVM*[2].

[1] *Résultat de l'application de l'algorithme MD5 (Message Digest Algorithm 5) à des données. Cet algorithme produit une empreinte numérique (ou condensat) de 128 bits, souvent représentée par une chaîne hexadécimale de 32 caractères.*

[2] *Architecture basée sur une extraction des caractéristiques par le modèle ConvNeXt et une classification à l'aide de l'algorithme SVM*

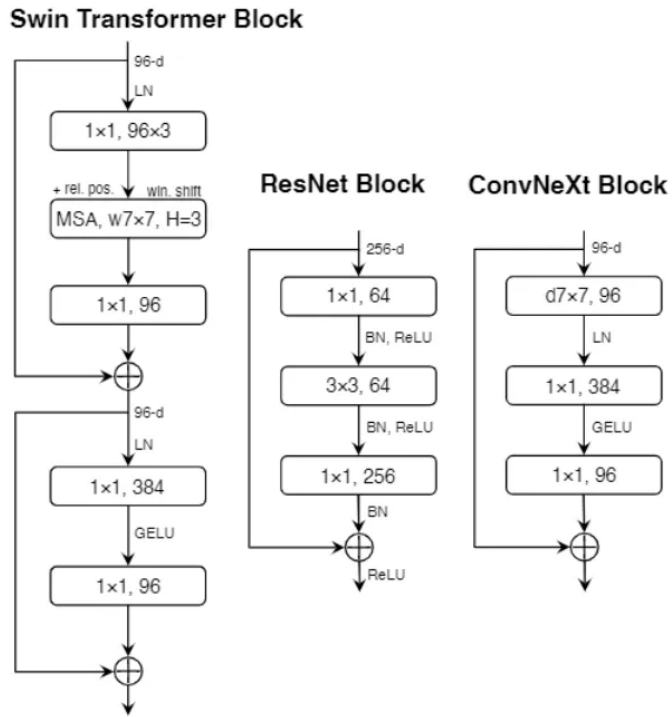
Modèle convNext

ConvNeXt est une architecture CNN modernisée qui intègre des améliorations inspirées par des approches issues des Vision Transformers

Une Révolution des Réseaux de Convolution pour la Vision par Ordinateur Moderne

ConvNeXt représente une avancée majeure dans le domaine des réseaux neuronaux convolutifs (ConvNets), combinant les principes éprouvés des architectures convolutionnelles avec des innovations inspirées des Vision Transformers. Développé initialement par Zhuang Liu et al. en 2022.

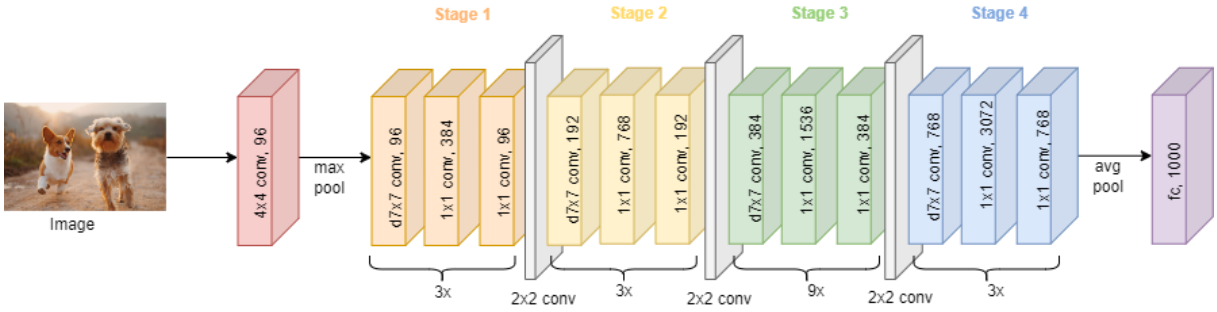
Figure 4: Achitecture Swin transformer, RestNet et ConvNeXt



Source: [A ConvNet for the 2020s]

ConvNeXt émerge d'une refonte systématique de ResNet-50, incorporant 11 modifications architecturales clés inspirées des Swin Transformers. Contrairement aux approches hybrides, cette modernisation préserve la pureté convolutionnelle tout en adoptant des principes de conception Transformer.

Figure 5 : Architecture ConvNeXt détaillée



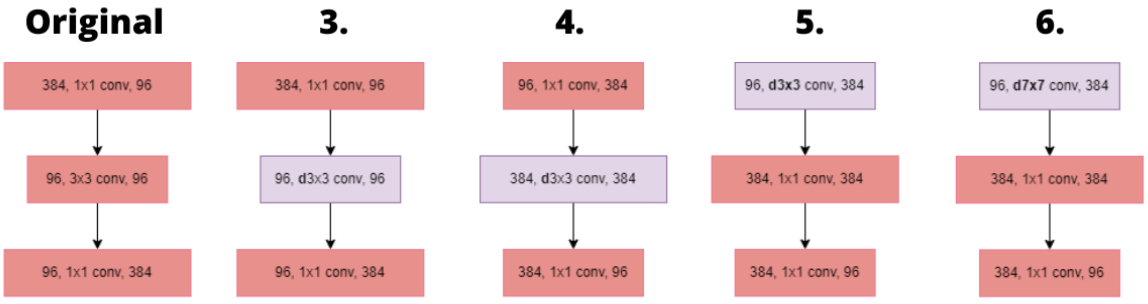
Source : https://tech.bertelsmann.com/en/blog/articles/convnext

Les ajustements des 11 caractéristiques distinctes à l'architecture ResNet augmentent la précision de 3,2 % au total et concerne :

1. Modifier le ratio de calcul des blocs ResNet utilisés par étape. ResNet-50 comporte quatre étapes, où les étapes un et quatre ont trois blocs, l'étape deux à quatre blocs et l'étape trois à six blocs. Swin-T se compose également de quatre étapes mais utilise des blocs de transformateurs Swin dans chaque étape. Les étapes un, deux et quatre de Swin-T ont deux blocs, tandis que l'étape trois en a six. Après avoir ajusté le ratio de calcul des blocs par étape de l'architecture ResNet pour le rendre plus similaire à celui du transformateur Swin, ConvNeXt a trois blocs dans les étapes un, deux et quatre, avec neuf blocs dans l'étape trois.
2. Modifier le stem pour "Patchify". Le stem de ResNet est une couche convolutionnelle 7x7. Mais le transformateur Swin utilise des patches non chevauchants avec une petite taille de noyau de quatre. Par conséquent, le stem de ConvNeXt est maintenant une couche convolutionnelle 4x4 avec un stride de quatre, afin que les patches ne se chevauchent pas.

Les prochains changements sont effectués à un niveau plus petit, les blocs ResNet :

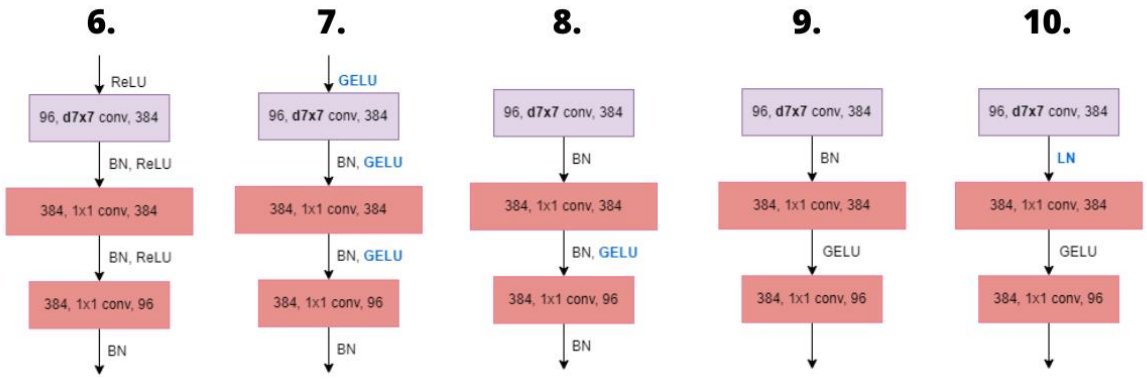
Figure 6: Ajustement 1, 2, 3, 4, 5 et 6



Source : https://tech.bertelsmann.com/en/blog/articles/convnext

3. Remplacer la couche convolutionnelle 3x3 par une convolution à profondeur. La convolution à profondeur est essentiellement une convolution groupée, où le nombre de groupes est égal au nombre de canaux. Cette opération est similaire à la somme pondérée en auto-attention.
4. Inverser le bottleneck. Changer les tailles des couches convolutionnelles dans un bloc de sorte que le bloc central soit le plus grand.
5. Déplacer la couche convolutionnelle à profondeur vers le haut pour qu'elle soit maintenant la première couche dans le bloc.
6. Augmenter la taille du noyau de la première couche dans le bloc à un noyau (à profondeur) de *7x7*.

Figure 7: Ajustement 6, 7, 8, 9 et 10



Source : https://tech.bertelsmann.com/en/blog/articles/convnext

7. Remplacer toutes les unités linéaires redressées (ReLU) dans les couches d'activation par des unités linéaires d'erreur gaussienne (GELU), car elles sont plus douces et utilisées par les transformateurs Swin.
8. Réduire le nombre de fonctions d'activation et ne conserver que celle après le bottleneck inversé.
9. Réduire le nombre de couches de normalisation et ne conserver que celle avant le bottleneck inversé.
10. Remplacer la normalisation par lots (BN) par la normalisation de couche (LN).
11. Ajouter des couches de sous-échantillonnage séparées entre les étapes, composées d'une normalisation de couche, d'une couche convolutive *2x2* avec un stride de 2 et d'une autre normalisation de couche. Ajouter également des normalisations de couche avant la première étape et après le pooling moyen après la quatrième étape.

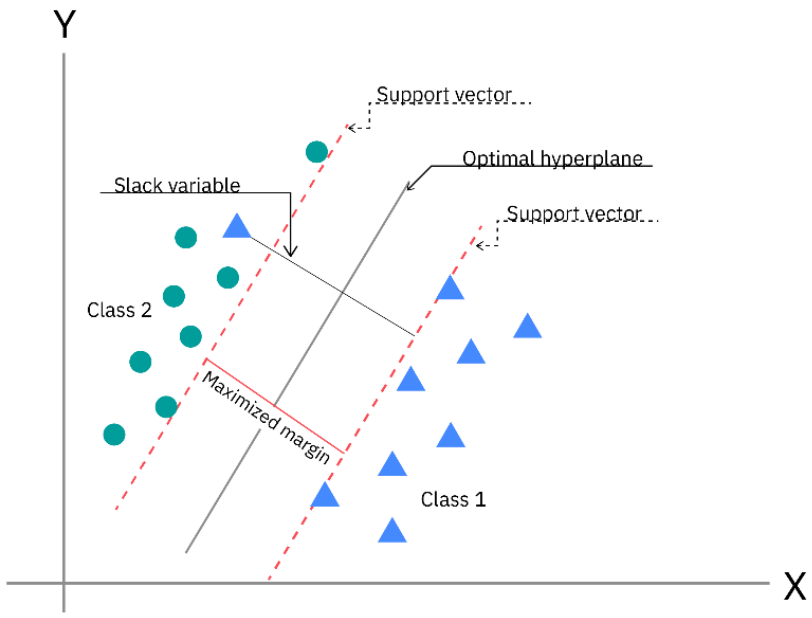
SVM (Shift Inverted Euclidiean-Linear kernel)

L'utilisation des noyaux hybrides dans les Machines à Vecteurs de Support (SVM) représente une avancée significative dans le domaine de l'apprentissage automatique. Cette étude approfondie se concentre sur le noyau "shift inverted Euclidean-Linear", une approche novatrice qui combine les propriétés des métriques de distance euclidienne inversée avec les caractéristiques de régularisation des noyaux linéaires. D'après les recherches récentes, ce noyau hybride offre des avantages substantiels en termes de stabilité et de performance dans les tâches de classification complexes. Notamment, l'intégration des techniques de décalage de valeurs propres et d'écrêtage assure la définie-positivité du noyau, condition essentielle pour l'efficacité des SVM. Cette approche a démontré des résultats particulièrement prometteurs dans la classification des dialectes et du langage familier, surpassant les méthodes traditionnelles tant en précision qu'en équilibre entre précision et rappel.

Machines à Vecteurs Supports

Les Machines à Vecteurs Supports constituent une classe d'algorithmes d'apprentissage automatique développée initialement par *Vladimir Vapnik*, reposant sur des principes solides de la théorie de l'apprentissage statistique. Le fonctionnement fondamental des SVM consiste à projeter les données dans un espace d'attributs de dimension élevée afin que les points puissent être classifiés, même lorsque les données ne sont pas linéairement séparables dans l'espace d'origine. Cette transformation permet d'identifier un séparateur optimal entre différentes catégories de données, généralement défini comme un hyperplan.

Figure 8:Illustration de svm appliquée en dimension 2



Source : https://www.ibm.com/think/topics/support-vector-machine

Dans le cas le plus simple d'une classification binaire linéairement séparable, l'algorithme cherche à déterminer l'hyperplan qui maximise la marge entre les deux classes. Cette marge correspond à la distance minimale entre l'hyperplan et les points d'apprentissage les plus

proches, appelés vecteurs supports. Mathématiquement, cela se traduit par un problème d'optimisation convexe visant à minimiser $\|w\|^2$ sous certaines contraintes, où w représente le vecteur normal à l'hyperplan séparateur. La formulation duale de ce problème d'optimisation révèle que la solution dépend uniquement des produits scalaires entre les vecteurs d'entrée, une propriété fondamentale qui permet l'introduction des fonctions noyau. Ces fonctions noyau, notées $K(xi, xj)$, permettent de calculer le produit scalaire des images des vecteurs d'entrée dans l'espace de caractéristiques sans avoir à calculer explicitement la transformation φ qui projette les données dans cet espace : $K(xi, xj) = \varphi(xi) \cdot \varphi(xj)$. L'utilisation des noyaux constitue ainsi une "astuce" computationnelle majeure ("kernel trick") qui permet aux SVM d'opérer efficacement dans des espaces de dimension potentiellement infinie sans jamais avoir à calculer explicitement les coordonnées des points dans cet espace transformé.

Le noyau "shift inverted Euclidean-Linear"

Le noyau "shift inverted Euclidean-Linear" représente une innovation récente dans la conception des fonctions noyau pour les SVM. Ce noyau hybride combine deux approches distinctes : une métrique de distance euclidienne inversée et une composante linéaire. Cette conception vise à exploiter simultanément les avantages des deux types de mesures de similarité pour améliorer les performances de classification.

La particularité de ce noyau réside dans l'intégration de techniques de décalage des valeurs propres ("Eigen value shifting") et d'écrêtage ("clipping") qui garantissent sa stabilité numérique et sa définie-positivité, une condition mathématique essentielle pour l'application valide d'un noyau dans le cadre des SVM. Sans cette propriété, la garantie de convergence vers une solution optimale globale serait compromise, rendant l'algorithme potentiellement instable ou inefficace.

L'approche hybride permet d'adresser simultanément deux aspects complémentaires de la similarité entre échantillons : la distance euclidienne inversée capture efficacement les relations de proximité non-linéaires entre points de données, tandis que la composante linéaire fournit une régularisation qui améliore la généralisation du modèle. Cette dualité rend le noyau particulièrement adapté aux problèmes où la structure des données présente à la fois des aspects linéaires et non-linéaires.

Dans sa formulation mathématique, le noyau hybride peut être conceptualisé comme une combinaison pondérée entre le noyau de distance euclidienne inversée (après les opérations de décalage et d'écrêtage) et un noyau linéaire standard. Cette combinaison crée un espace de caractéristiques enrichi qui permet une séparation plus efficace des classes complexes, tout en maintenant une bonne capacité de généralisation.

Paramètres de performance

Ici les deux approches ont été évaluées selon différentes métriques. Ces métriques prendront en compte les notations suivantes : VP (Vrais Positif), VN (Vrais Négatifs), FP (Faux Positifs), FN (Faux Négatifs).

Accuracy

Il mesure le nombre d'images correctement classées par rapport au nombre total d'images de l'ensemble de données.

$$Accuracy = \frac{VP}{VP + VN + FP + FN} X100$$

Precision

Il mesure le nombre total de cas correctement classés et il est donné par :

$$Precision = \frac{VP}{VP + FP} X\ 100$$

Recall

Il mesure le nombre total de cas positifs correctement classés et est donné par :

$$recall = \frac{VP}{VP + VN} X\ 100$$

F1-score

Elle mesure l'équilibre entre la précision du modèle et le recall

$$F1 - score = 2X \frac{precsion\ X\ recall}{precision + recall}$$

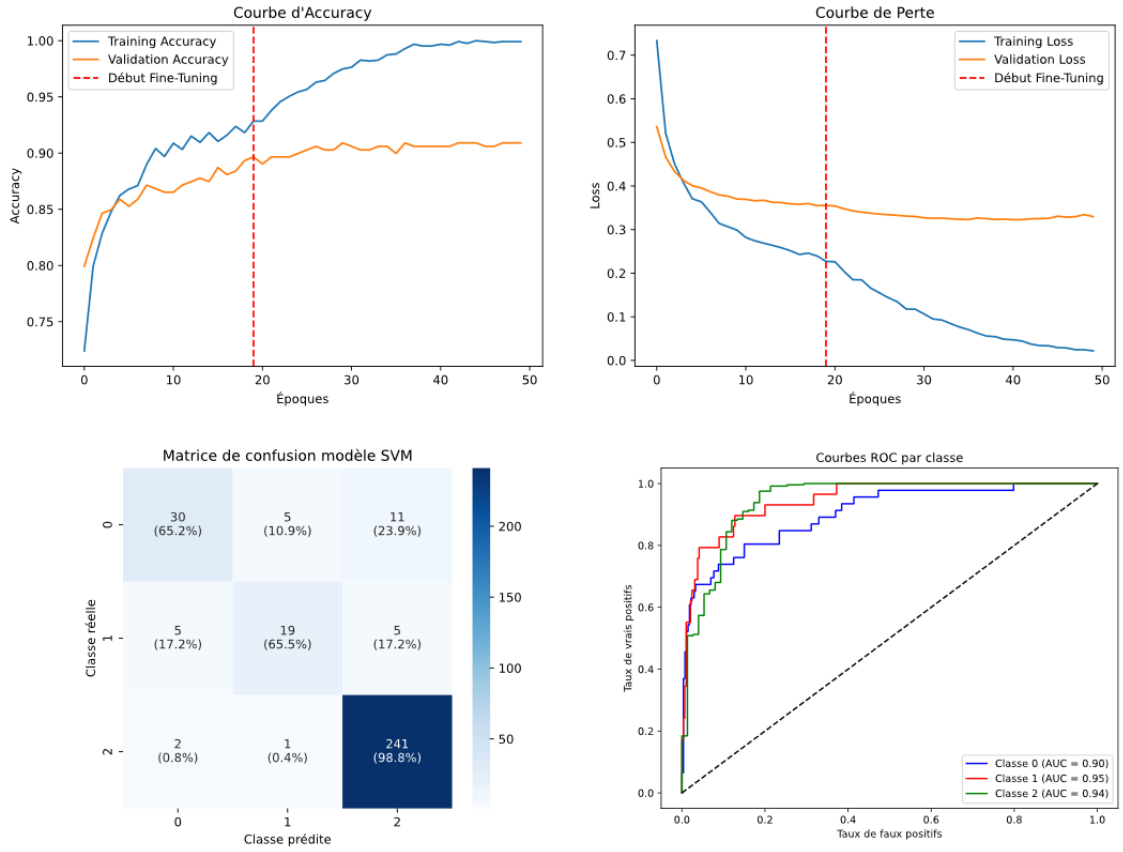
Résultats et discussions

Analyse des résultats

Les résultats suivants ont été obtenus en utilisant la version python 3.12.4 un ordinateur HP-Victus Intel® Core™ i5 12500H, 32 Gb Ram et 1Tera ssd.

a) Évaluation des Performances de ConvNeXt

Figure 9: Performances du modèle ConvNext

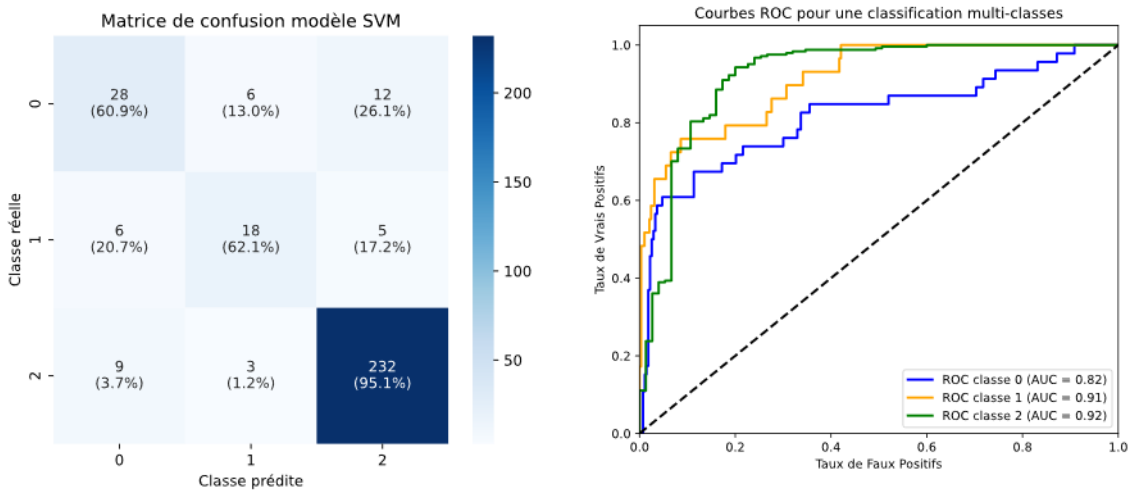


L'évaluation du modèle ConvNeXt montre une excellente performance sur les données d'entraînement, avec une accuracy, ***une précision et un rappel de 99,88 %***. Cependant, cette performance ne se maintient pas sur l'ensemble de validation, où nous observons une accuracy de ***90,91 %, une précision de 91,46 % et un rappel de 90,60 %.*** Cette diminution notable des performances en validation suggère un phénomène de surapprentissage (overfitting). Le modèle apprend trop spécifiquement les données d'entraînement, ce qui limite sa capacité à généraliser à de nouvelles images. La ***loss*** en validation de ***0,3297*** confirme cette tendance, indiquant que le modèle rencontre des difficultés à s'adapter aux variations de nouvelles données. L'AUC par classe est également élevé, avec des valeurs de ***0,90, 0,95 et 0,94*** pour les classes 0, 1 et 2 respectivement, confirmant la bonne séparation entre les classes.

b) Évaluation des Performances de ConvNeXt + SVM

Pour l'entraînement du modèle, ConvNeXt est utilisé comme extracteur de caractéristiques, tandis qu'un classifieur SVM assure la classification.

Figure 11: Performances du modèle ConvNext+SVM



L'évaluation du modèle sur l'échantillon de validation confère : une **accuracy de *86,94 %*, une précision de *86,85 %* et un rappel de *86,94 %***. La **loss de *0,3287*** est légèrement inférieure à celle du modèle ConvNeXt seul, mais la baisse de précision montre que **le SVM n'exploite pas aussi bien les caractéristiques extraites** que la dernière couche dense d'un CNN entraîné de bout en bout. L'AUC est également plus faible, avec des valeurs de 0,82, 0,91 et 0,92 pour les classes 0, 1 et 2 respectivement.

c) Analyse Comparative

Tableau 1: Tableau comparatif des perfomances des modèles étudiés

Modèle	**Accuracy**	**Précision**	**Recall**	**AUC class0**	**AUC class1**	**AUC class2**	**Temps d'exécution**
ConvNeXt	*90,9 %*	*91,5 %*	*90,6 %*	*90%*	*0,95*	*0,94*	*862 min*
ConvNeXt + SVM	*87,2 %*	*87,1 %*	*87,0 %*	*0,82*	*0,91*	*0,92*	*32 min*

L'analyse comparative entre **ConvNeXt** (version Small) et **ConvNeXt** (version Small) + **SVM** met en évidence un compromis entre performance et efficacité computationnelle. ConvNeXt seul affiche de meilleures performances avec une ***accuracy de 90,91 %***, une ***précision de 91,46 %*** et des **AUC plus élevés** pour toutes les classes, indiquant une meilleure capacité à discriminer les catégories. En revanche, l'ajout d'un SVM après ConvNeXt réduit ces performances (***accuracy de 87,2 %)***, bien que la **loss diminue légèrement *(0,3287 vs. 0,3297)***. L'avantage principal du SVM réside dans une **réduction drastique du temps d'entraînement *(32 min contre 862 min)***, ce qui peut être déterminant dans des environnements contraints en ressources.

Toutefois, la baisse de l'AUC et de l'accuracy suggère une perte d'efficacité du modèle hybride. Ainsi, le choix entre ces approches dépendra des priorités : **maximisation de la performance avec ConvNeXt seul** ou **optimisation du temps de calcul avec ConvNeXt + SVM, au prix d'une légère dégradation des résultats**.

Conclusion

En conclusion, l'analyse des performances montre que **ConvNeXt seul** offre des résultats supérieurs en termes de précision, d'accuracy et de pouvoir discriminant (AUC), confirmant sa capacité à extraire des caractéristiques pertinentes pour la classification. Cependant, ce gain de performance se fait au prix d'un **temps d'entraînement extrêmement long (862 min)**, ce qui peut être un frein dans un contexte où l'efficacité computationnelle est cruciale. À l'inverse, **l'association ConvNeXt + SVM** réduit drastiquement le temps d'entraînement (**32 min seulement**), tout en conservant des performances acceptables malgré une légère dégradation de l'accuracy et des scores AUC. Ainsi, le choix du modèle dépendra des contraintes spécifiques du projet : **ConvNeXt seul pour une performance optimale**, et **ConvNeXt + SVM pour un compromis entre rapidité et précision**.

Référence

[1] Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell2 Saining XieA ConvNet for the 2020s, https://arxiv.org/pdf/2201.03545

[2] Department of Computer Applications, Kalasalingam Academy of Research and Education, Krishnankovil, TamilNadu, India, A Hybrid Shifted Inverted Euclidean-Linear Kernel with Stacked Classifiers for Tamil Slang Classification https://ieeexplore.ieee.org/document/10859321/authors#authors

[3] Earl Clarence S. San Diego, Seph Gerald C. Rodrin, Edwin R. Arboleda, classification of prominent cacao pod diseases using multi-feature visual analysis and k-nearest neighbors' algorithm

[4] Rahma Kadria, b, Bassem Bouaziza , Mohamed Tmara, Faiez Gargouri, Innovative multi-modal approachestoAlzheimer'sdiseasedetection: TransformerhybridmodelandadaptiveMLP-Mixer

[5] MeghaArakeri, DhatvikMP, AVKavan, KammaSushreyaMurthy, Nagineni Lakshmi Nishitha and Napa Lakshmi, Intelligent pesticide recommendation system for cocoaplant using computervision anddeeplearningtechniques