

Master of Science HES-SO in Engineering

Orientation : Technologies de l'information et de la communication
(TIC)

Une IA pour surveiller les engagements de zéro déforestation

Fait par
Romain Capocasale

Sous la direction de
Prof. Andres Perez-Uribe,
HEIG-VD

Lausanne, HES-SO//Master, 04.06.2021

Accepté par la HES-SO//Master (Suisse, Lausanne) sur proposition de

Prof. Perez-Uribe Andres, conseiller du projet d'approfondissement
Andrei Popescu-Belis, expert principal

Lausanne, le 04.06.2021

Prof. Perez-Uribe Andres
Conseiller

Prof. Hennebert Jean
Responsable de la filière

Table des matières

| | |
|---|-----|
| Table des matières | iii |
| Remerciements | v |
| Abréviations | vi |
| Abstract | vii |
| 1. Introduction | 8 |
| 1.1. Contexte | 8 |
| 1.2. But de l'étude | 8 |
| 1.3. Scope et limitations de l'étude | 9 |
| 2. Analyse documentaire et proposition de recherche | 10 |
| 2.1. État de l'art | 10 |
| 2.2. Transfer learning | 10 |
| 3. État du projet | 10 |
| 3.1. Réseau de neurones from Scratch | 10 |
| 3.2. Modèle préentraîné DenseNet Eurosat | 11 |
| 4. Jeu de données | 13 |
| 4.1. Répartition des classes | 13 |
| 4.2. Séparation stratifiée aléatoire | 13 |
| 4.3. Séparation géographique | 14 |
| 4.3.1. Séparation sur 10 folds | 15 |
| 4.4. Culture contre non-culture | 16 |
| 5. Solutions développées | 17 |
| 5.1. Prétraitement des images | 17 |
| 5.2. Générateur et augmentation de données | 17 |
| 5.3. Normalisation | 18 |
| 5.4. RestNet-50 Tensorflow Hub | 19 |
| 5.5. DenseNet Transfer learning | 20 |
| 5.5.1. Variation du nombre de bandes | 21 |
| 5.6. Focal loss | 22 |
| 5.7. Dégel de toutes les couches du réseau | 22 |
| 5.8. Culture vs non-culture | 23 |
| 5.9. Modèle multi labels | 23 |
| 6. Environnement d'exécution | 25 |
| 7. Résultats | 26 |
| 7.1. Cross validation jeu de données stratifié | 27 |
| 7.1.1. Café contre reste | 27 |
| 7.1.2. Culture contre non-culture | 28 |
| 7.2. Cross validation géographique | 30 |

| | |
|-----------------------------------|----|
| 7.2.1. Café contre reste | 30 |
| 7.2.2. Culture contre non-culture | 32 |
| 7.3. Variation de bandes | 34 |
| 7.3.1. DenseNet 3 bandes | 34 |
| 7.3.2. DenseNet 13 bandes | 34 |
| 7.1. Modèle multi labels | 35 |
| 8. Discussion | 37 |
| 8.1. Variation de bandes | 37 |
| 8.2. Café contre reste | 37 |
| 8.2.1. Séparation géographique | 37 |
| 8.3. Culture contre non-culture | 39 |
| 8.3.1. Séparation géographique | 39 |
| 8.4. Modèle multi labels | 39 |
| 8.5. Comparaison des scores | 40 |
| 9. Récapitulatif des résultats | 41 |
| 10. Conclusion | 42 |
| 11. Table des figures | 43 |
| 12. Références | 44 |
| 13. Annexes | 45 |

Remerciements

Au terme de ce travail, je tiens à exprimer ma gratitude au professeur *Perez-Uribe Andres* pour les conseils et le soutien qu'il m'a prodigués tout au long du projet.

Je tiens à remercier également *Fahy Axel Brian* et *Satizabal Mejia Hector Fabio* pour l'aide et les conseils fournis.

Abréviations

| | |
|------|---|
| CNN | Convolutional Neural Network |
| CIAT | International Center for Tropical Agriculture |

Abstract

Au Vietnam, la forêt est petit à petit remplacée par différentes cultures, notamment des champs de café. De nombreuses entreprises vietnamiennes ont signé des accords de zéro-déforestation qui les engage à acheter des produits provenant d'une zone non déforestée. Cependant ces entreprises ne savent pas si les produits qu'elles achètent proviennent d'une région déforestée récemment ou non. La HEIG-VD en collaboration avec le centre de recherche sur l'Agriculture Tropicale (CIAT) et le King's College London (KCL) développent différents outils avec comme objectif de détecter la déforestation et surveiller les changements dans l'utilisation des sols.

Le but de ce projet est d'utiliser des modèles préentraînés pour effectuer du transfer learning avec l'objectif de détecter la présence ou non de champs de café sur une image. Le modèle préentraîné utilisé pour ce projet est un modèle DenseNet201 avec 13 bandes entraîné sur le jeu de données Eurosat [3]. En parallèle une expérience est réalisée pour détecter la présence ou non de culture sur une image satellite.

Les images satellites sont récupérées à partir de la localisation des champs de café fournis par le CIAT. Dans un premier temps, les images ont été séparées en jeu d'entraînement et de test d'une manière aléatoire et par la suite séparées en prenant compte leur position géographique.

Différentes expériences ont été réalisées dans le but d'augmenter la performance des modèles comme l'utilisation de différentes tailles d'images, modification de la méthode de normalisation des images, variation du nombre de bandes, dégel de toutes les couches d'un modèle de transfer learning, etc...

Cette étude montre que les modèles détectant la présence de champs de café obtiennent de meilleurs résultats avec une séparation aléatoire que géographiques. Il est également constaté que les modèles de transfer learning performe mieux que les modèles entraînés de manière « from scratch ». Le transfer learning apporte donc un gain de performance dans ce cas-ci. Le projet a également permis de montrer que les modèles entraînés à détecter la présence de culture ou non-culture donnent de bien meilleurs résultats que les modèles café contre reste, peu importe le type de séparation du jeu de données.

Mot-clés: Machine learning, Deep learning, Transfer learning, Feature extraction, Fine tuning, Traitement d'images, CNN, Déforestation, Image satellite

1. Introduction

1.1. Contexte

Au Vietnam, chaque année la forêt est remplacée par différentes cultures, dont des champs de café. De nombreuses entreprises vietnamiennes ont signé des accords de zéro-déforestation qui les engage à acheter des produits provenant d'une zone non déforestée. Cependant ces entreprises ne savent pas si les produits qu'elles achètent proviennent d'une région déforestée récemment ou non.

En collaboration avec le centre de recherche sur l'Agriculture Tropicale (CIAT) au Vietnam et le King's College London (KCL), la HEIG-VD développe des outils exploitant des algorithmes de Machine Learning pour traiter des informations fournies par des capteurs de télédétection avec comme but détecter la déforestation et surveiller les changements dans l'utilisation des sols.

Le but de ce projet est développé différents réseaux de neurone capable d'indiquer la présence ou non de champs de café sur une image satellite. Les images utilisées par le réseau de neurones proviennent la mission Sentinel-2 de l'ESA (European Space Agency) et le CIAT fournis des annotations de la localisation de différents types de terrains au Vietnam et notamment de champs de café.

De nombreux projets utilisent déjà des réseaux de neurones pour faire de la classification sur des images satellites. La grande majorité de ces projets utilisent pour cette tâche des architectures de type CNN. Au vu du grand nombre de réseau à disposition, il peut être contre-productif de réentraîner son propre réseau de neurones (surtout si le nombre d'exemples labellisé est faible). La solution proposée dans ce projet est donc d'utiliser des modèles préentraînés et d'effectuer du transfer learning sur notre propre jeu de donnée.

1.2. But de l'étude

Ce projet d'approfondissement dans le cadre du Master HES-SO est basé sur un travail de bachelor portant sur la même thématique [1]. La méthodologie du précédent travail consistait à entraîner les réseaux de neurones « From Scratch » avec des architectures CNN conventionnelles. Cette solution demande du temps et un jeu de données de grande taille.

Le but de ce projet est d'utiliser des techniques de transfert learning pour effectuer la même tâche de classification que le travail de bachelor avec un modèle préentraîné. Le modèle de transfert learning principalement utilisé pour ce projet est le résultat d'un travail de master [2] d'un ancien étudiant. Ce modèle a été entraîné sur le jeu de données Eurosat.

Les principales étapes du projet sont les suivants :

1. Mettre en place le pipeline de transformation des images satellites proposé dans les précédents travaux à disposition.
2. Entraînement de réseaux de neurones avec des modèles préentraînés sur Eurosat trouvés sur Internet.
3. Entraînement de réseaux de neurones avec le modèle préentraînés proposé dans le travail de Master.
4. Entraînement de réseaux de neurones pour détecter les terrains de type culture contre les terrains de type non-culture.
5. Comparaison des résultats des différentes expériences réalisées.

1.3. Scope et limitations de l'étude

Ce travail se concentre l'élaboration de réseaux de neurones pour détecter la présence ou non d'un champ de café sur une image satellite. Le centre de cette étude consiste à tester différente variante de réseau de neurones dans l'objectif de les comparer les unes avec les autres. Les différentes variantes se concentrent sur les étapes de prétraitement appliquée aux images, le type d'architecture utilisée, etc.

Il est considéré comme en dehors de ce travail, la détection de champs de café non labélisé sur la carte du Vietnam et la détection de la déforestation d'une année à l'autre.

2. Analyse documentaire et proposition de recherche

2.1. État de l'art

Il existe plusieurs réseaux de neurones entraînés sur des images satellites dans le but de classifier différents types de terrains. C'est notamment le cas du projet Eurosat [3] qui propose un benchmark des scores obtenu sur des images satellites de paysage européen labélisé à la main. La comparaison a été effectuée avec différentes architectures CNN.

Pour ce benchmark les 13 bandes des images satellites ont été utilisées. La taille des images est de 64x64.

Voici un aperçu des accuracy obtenues avec le jeu de données Eurosat avec train-test split de 80/20:

| Method | UCM | AID | SAT-6 | BCS | EuroSAT |
|-----------|--------------|--------------|--------------|--------------|--------------|
| ResNet-50 | 96.42 | 94.38 | 99.56 | 93.57 | 98.57 |
| GoogLeNet | 97.32 | 93.99 | 98.29 | 92.70 | 98.18 |

Figure 1 : P. Helber & al. results on their two best-performing fine-tuned CNNs[1, p. 6]

On constate que le réseau de neurones qui obtient le meilleur résultat est **ResNet-50** avec un résultat de **98.57%**.

2.2. Transfer learning

Le transfer learning consiste à utiliser un réseau de neurones préentraîné sur une grande quantité d'images. Le modèle préentraîné a appris à reconnaître les caractéristiques de bas niveau des images et uniquement les couches supérieures qui correspondent aux caractéristiques de haut niveau sont entraînées sur le jeu de données cibles. Cette approche permet de :

- Diminuer le nombre de données labélisé nécessaire.
- Utiliser les caractéristiques de bas niveau connu.

3. État du projet

Ce chapitre permet de donner une vue d'ensemble des deux projets qui définisse la base de ce travail.

3.1. Réseau de neurones from Scratch

Le projet sur lequel se base ce travail [1] propose différentes architectures de réseaux de neurones entraînés de manière "from scratch".

Différentes méthodes d'entraînements ont été utilisées comme :

- Variation du nombre de bandes utilisé
- Combinaison de certaines bandes entre elles
- Prise en compte de la saison d'acquisition des images
- Utilisation de différentes sources d'images satellites (Sentinel 2A, 1C, 1)

Voici un tableau récapitulatif des méthodologies utilisé dans ce travail.

| | |
|--------------------------|-------------------------------------|
| Images utilisées | Sentinel2A, Sentinel1C et Sentinel1 |
| Taille des images | 15 x 15 |
| Nombre de bandes | Entre 1 et 4 bandes |
| Normalisation | Normalisation Min-Max |

Tableau 1 : Paramètres modèles J.Rod [1]

Voici un tableau des trois meilleurs modèles obtenus avec les images Sentinel 2 type 1C:

| Bandes utilisées | Saison | F-Score |
|---------------------------------|---------------|----------------|
| Bleu, Vert, Rouge et Infrarouge | Hiver | 76.21% |
| Bleu, Rouge et Infrarouge | Automne | 76.10% |
| Bleu, Vert, Rouge et Infrarouge | Automne | 75.99% |

Tableau 2 : Scores modèles Sentinel 1C J.Rod [1]

Voici un tableau des trois meilleurs modèles obtenus avec les images Sentinel 2 type 2A:

| Bandes utilisées | Saison | F-Score |
|---------------------------------|---------------|----------------|
| Bleu, Vert, Rouge et Infrarouge | Pas précisé | 86.66% |
| Vert, Rouge et Infrarouge | Pas précisé | 86.18% |
| Bleu, Vert et Rouge | Pas précisé | 85.97% |

Tableau 3 : Scores modèles Sentinel 2A J.Rod [1]

Il est important de séparer les deux types d'images (1C et 2A), car on remarque que les modèles entraînés sur les images Sentinel 2A ont de meilleures performances (environ 10%) que les modèles entraînés sur les images Sentinel 1C. Dans notre cas, comme les images de type 1C sont utilisées il est intéressant de comparer les scores avec les deux types d'images.

3.2. Modèle préentraîné DenseNet Eurosat

Le principal modèle qui est utilisé pour le transfer learning provient d'un travail de master d'un ancien étudiant [2]. Ce travail consistait entre autres à prédire à partir d'une image satellite le type de terrain (Industriel, Forêt, Rivière, ...). Ce modèle a été entraîné à partir du jeu de données Eurosat [3].

Voici une représentation de l'architecture utilisée. Le modèle se base sur une architecture DensetNet modifiée pour prendre en entrée les 13 bandes des images Eurosat.

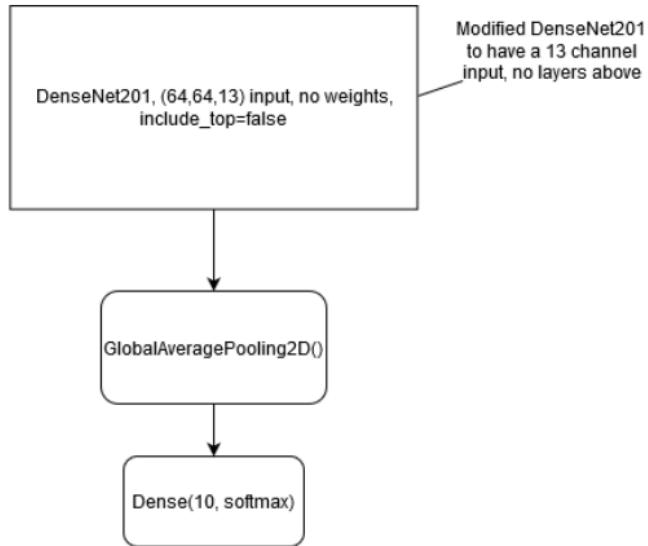


Figure 2: Architecture réseau de neurones E.Ransom

Le modèle a obtenu les résultats suivants :

| Accuracy | F-Score |
|----------|---------|
| 97.58 % | 97.0 % |

Tableau 4 : Score modèle E.Ransome [2]

Voici un tableau récapitulatif de la méthodologie utilisée dans ce travail.

| | |
|-------------------|-----------|
| Images utilisées | Eurosat |
| Taille des images | 64 x 64 |
| Nombre de bandes | 13 bandes |
| Normalisation | Z-norm |

Tableau 5 : Paramètres modèles E.Ransome [2]

Au vu des résultats obtenus, on constate qu'il est très intéressant d'utiliser ce modèle pour effectuer du transfer learning. Les caractéristiques de bas niveau apprises sur les images pourront donc être utilisées pour notre problème.

4. Jeu de données

Les images utilisées sont les images satellites Sentinel2 de type **1C** avec **13 bandes** de taille **10980x10980**. La même procédure de récupération des images que J.Rod [1] (Page 12, Chapitre 2.4) a été utilisé, c'est-à-dire 5 images ont été téléchargés. Ces 5 images permettent de couvrir l'ensemble les labels fournit par le CIAT.

4.1. Répartition des classes

Sur la capture ci-dessous, on constate qu'il y a moins d'images de café que la classe autre. Il faudra donc tenir compte de cette information lors de l'entraînement des modèles. Pour l'expérience culture contre non-culture, on constate que les classes sont plus ou moins équilibrées et qu'il n'y a pas de mesure particulière à prendre pour l'entraînement.

Les statistiques ci-dessous ont été réalisées sur les images de taille 64x64 (cf chapitre 5.1. Prétraitement des images). Les mêmes tendances sont conservées pour les images 32x32.

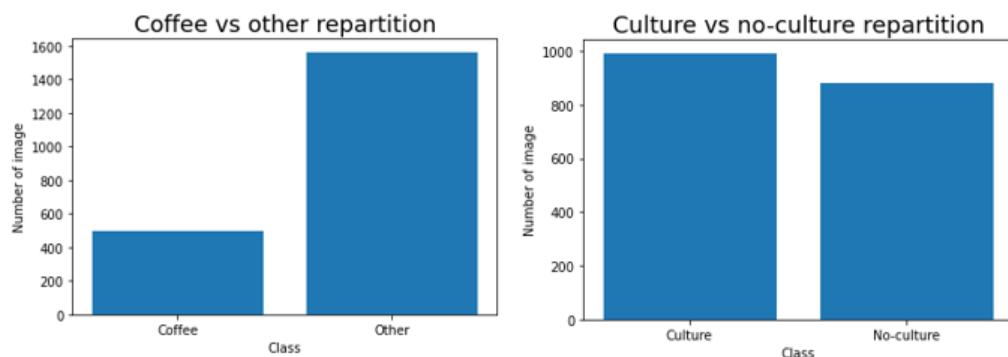


Figure 3 : Répartition des classes

4.2. Séparation stratifiée aléatoire

Un lot d'image Sentinel2 a été sélectionné sur la période de l'année du 1^{er} février au 1^{er} avril.

Les images ont été sélectionnées sur cette période à l'aide de l'étude réalisée par S. Walther (cf Annexe 1), qui met en avant 2 informations :

1. Les mois de janvier à avril sont les mois de l'année au Vietnam où il y a le moins de couvertures nuageuses sur les images satellites. L'utilisation des images de cette saison va permettre de limiter la présence d'image inutilisable (car perturbé par des nuages) dans le jeu de données.
2. De janvier à avril, les arbres à Café sont fortement arrosés, dans le but de fleurir en février. Durant cette période, les arbres à café sont donc plus facilement visibles depuis le ciel à l'aide des fleurs blanches qui les composent.

Cinq images satellites sur cette période ont alors été sectionnées. Comme le jeu de données est non balancé (moins de café que les autres classes), une répartition stratifiée entre le jeu d'entraînement et des tests sont réalisés pour avoir une fréquence des classes réaliste. La même idée est appliquée lors de la validation croisée.

Note : Une autre expérience a été réalisée initialement, mais n'a pas été retenue : il s'agit de la *sélection d'une image par saison sur une année* : cette option n'a pas été retenue, car il est plus difficile pour le modèle de généraliser sur des images de différentes saisons.

4.3. Séparation géographique

Une idée supplémentaire apparue en cours du projet est d'effectuer une séparation entre le jeu d'entraînement, de validation et de test de manière géographique. En effet, au vu des bons scores obtenus par certains modèles, il se peut que deux images proches géographiquement soient très similaires. Des images semblables peuvent alors se trouver dans le jeu d'entraînement et de test ce qui peut biaiser le résultat.

Une expérience réalisée est alors d'effectuer une séparation entre les jeux de données en prenant compte de la position géographique des images. Le package Python spacy¹ permet d'effectuer cette opération. Comme le montre la capture ci-dessus une séparation a été effectuée pour que le jeu d'entraînement et de test soit géographiquement à une position différente. Ceci permet d'assurer que le modèle n'aura pas déjà rencontré une image semblable lors de l'entraînement. La même opération est effectuée pour séparer le jeu d'entraînement en N fold lors de la validation croisée. La capture ci-dessous montre la séparation des différentes images sur la carte du Vietnam :

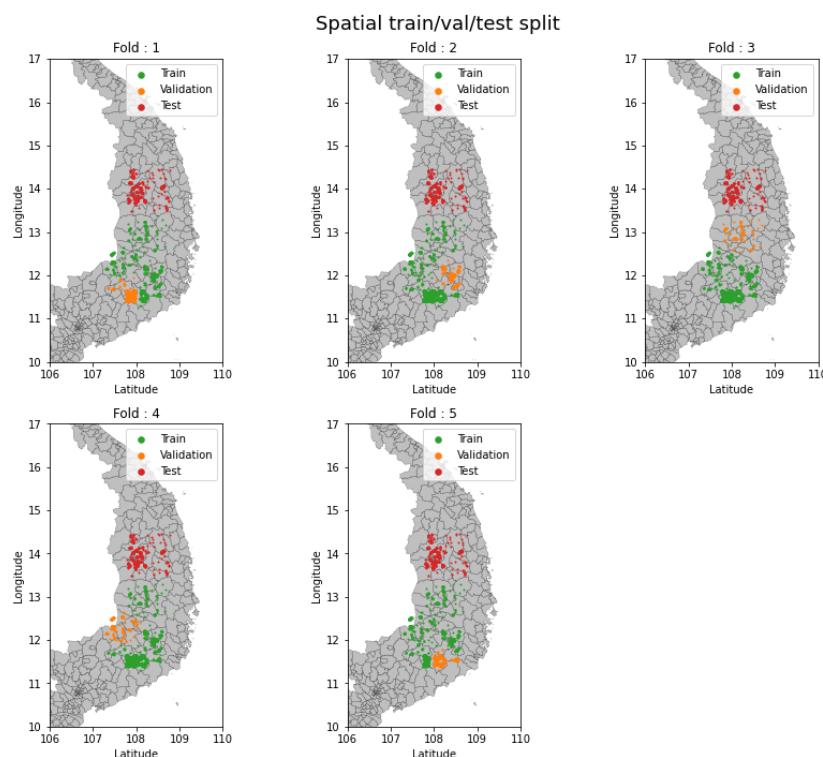


Figure 4 : Séparation géographique du jeu de données

Cependant cette méthode dispose de deux inconvénients principaux :

1. Comme le nombre de champs de café labélisé est inférieur à la classe autre et que la séparation ne prend pas en compte la classe, les folds se trouvent déséquilibrés. Certaines fold ne contiennent que très peu de champs de café et l'équilibre entre les classes est donc faussé.
2. On peut voir sur la capture ci-dessus que la séparation entre les différents jeux de données est très stricte. Le jeu de test par exemple se trouve géographiquement très éloigné du jeu d'entraînement. Il se peut que le climat et le type de sol soient différents d'une zone à l'autre. Un modèle entraîné sur une zone aura alors de la peine à bien généraliser sur un autre.

¹ <https://pypi.org/project/spacy/>

4.3.1. Séparation sur 10 folds

Une méthode alternative a été développée pour contrer au mieux ces deux problèmes. Cet algorithme fonctionne comme ceci : pour effectuer par exemple une validation en cinq folds, l'algorithme expliqué précédemment est exécuté séparément sur les points de café pour dix fold et sur les points autre que café pour dix fold. Deux folds de café et deux folds autres sont sélectionnés comme jeu de validation et les folds restantes sont utilisées pour l'entraînement. Pour notre exemple, cette opération est effectuée 5 fois pour obtenir 5 folds.

Cette version de l'algorithme permet de sélectionner des données de deux zones différentes et en respectant la répartition des classes. La capture suivante présente un schéma explicatif de comment les folds sont sélectionnées (dans le cas d'une répartition avec 5 folds) :

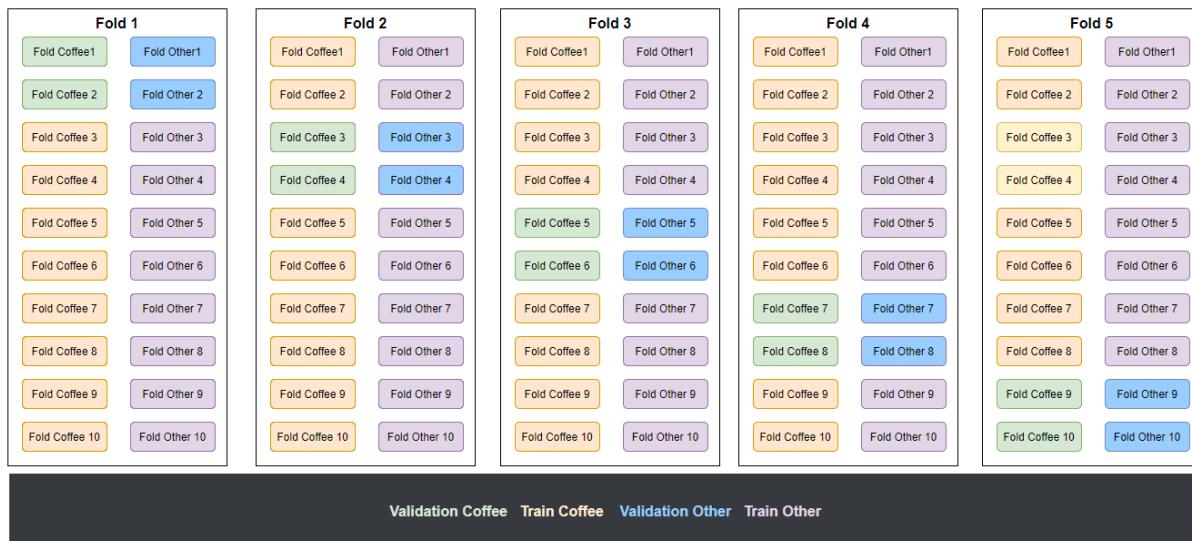


Figure 5 : Séparation jeu de données par classe sur 10 folds – visualisation

La capture suivante présente un exemple de séparation pour une fold sur la carte du Vietnam :

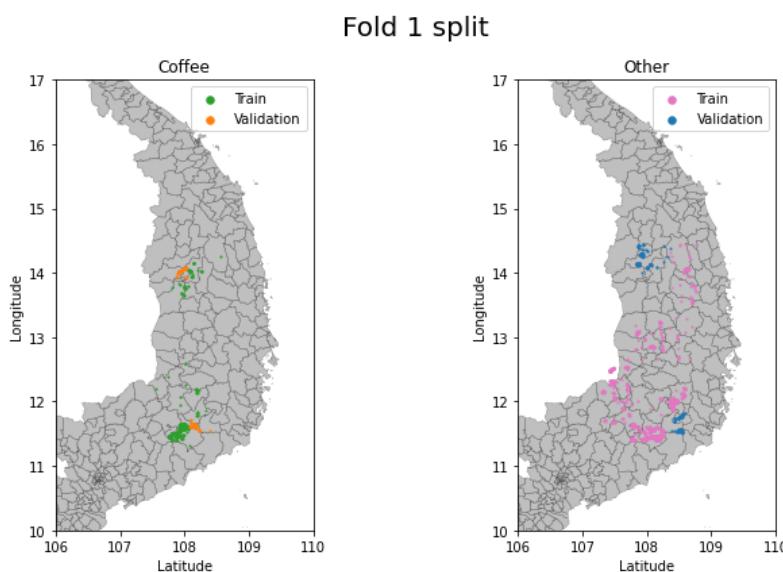


Figure 6 : Séparation jeu de donnée, méthode des 10 folds, séparation classe par classe

Les données en orange sont réunies avec les données en bleu et forment le jeu de validation pour la fold 1, les données en vert et en rose sont réunies pour former le jeu d'entraînement de la fold 1. L'opération est répétée pour les 4 folds suivantes.

Pour séparer le jeu d'entraînement du jeu de test, l'algorithme présenté ci-dessus est lancé sur toutes les données pour 5 folds. Une des folds est enregistrée comme jeu de test sur le disque tandis que les 4 restantes sont enregistrées comme jeu d'entraînement. L'algorithme est ensuite relancé sur le jeu d'entraînement pour effectuer la séparation entre entraînement et validation.

Voici un exemple de résultats finaux pour le jeu d'entraînement, validation et de test obtenu avec cette méthode :

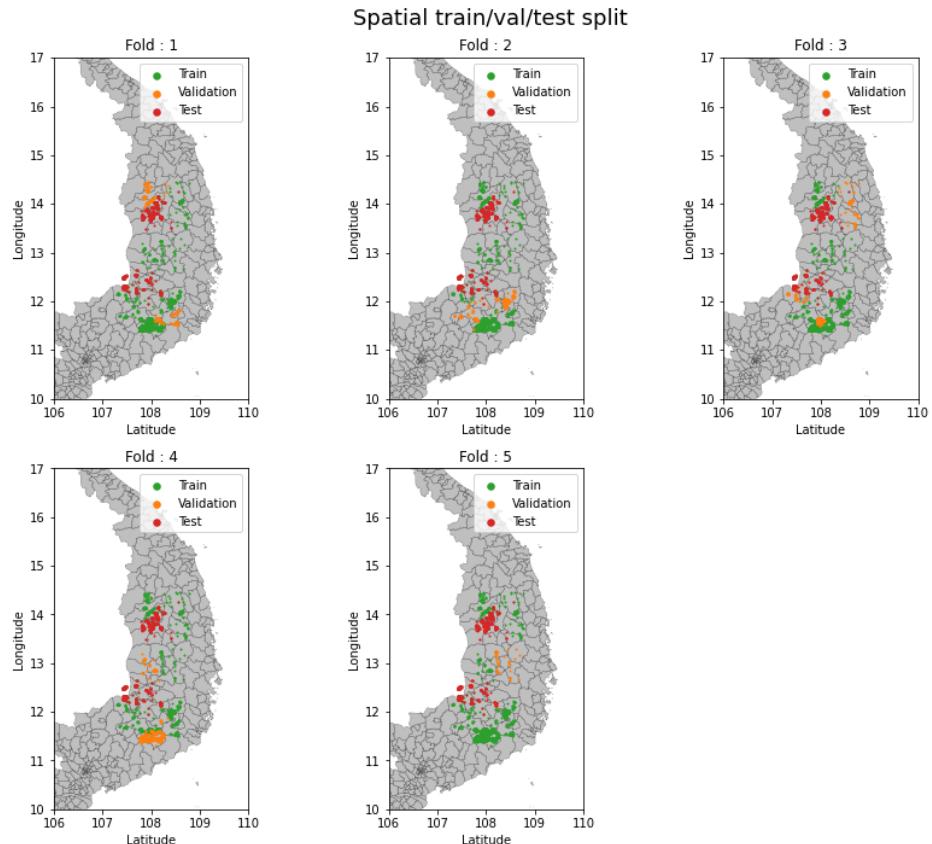


Figure 7: Séparation jeu de donnée, méthode des 10 folds, résultat final

Il est à noter que même si cette méthode effectue une séparation géographique par classe, le nombre de champs de café sélectionné d'une fold à l'autre reste variable étant donné que les champs de café n'ont pas été répartis uniformément sur la carte du Vietnam.

4.4. Culture contre non-culture

Les deux mêmes techniques ont été utilisées pour la création du jeu de données pour l'expérience culture contre non-culture.

5. Solutions développées

5.1. Prétraitement des images

Dans le but de rester le plus proche possible des images Eurosat et du modèle utilisé pour le transfer learning les points suivants ont été respectés.

- Utilisation des images Sentinel 1C
- Taille des images en 64x64
- Normalisation z-norm

Tout d'abord un script a été mis en place pour télécharger les images Sentinel 2 à l'aide du package Python SentinelSat².

Étant donné que les bandes des images Sentinel2 n'ont pas toute la même résolution³, il faut donc effectuer une interpolation bicubique pour que toutes les bandes aient une taille de pixel de 10 mètres comme indiqué dans le travail de E.Ransom [2] (Page 7, Chapitre4.3). Cette opération est effectuée avec la librairie opencv⁴.

Une fois toutes les bandes ont la même résolution, les images originales de taille 10980x10980 sont découpées en image de taille de 64x64. Il est à noter que comme 10980 n'est pas divisible par 64, une bande de 36 pixels sur la droite et en bas de l'image n'est pas utilisée. Les images découpées sont ensuite localisées et labélisées et fonction des fichiers Shapefile fournit par le CIAT. Pour les expériences avec des images de tailles 32x32, la même méthodologie est utilisée.

La dernière étape consiste à enregistrer les images sur le disque en fonction du label qui lui a été attribué. Ces différentes étapes de prétraitement ont été reprises du travail de J. Rod [1].

Le schéma ci-dessus présente un récapitulatif des différentes étapes :



Figure 8 : Récapitulatif, étapes de preprocessing

Ces différentes opérations se trouvent dans le notebook : *ImageProcessing.ipynb*.

5.2. Générateur et augmentation de données

Étant donné la taille importante des images, un générateur a été mis en place. Un générateur permet de charger les images et de les fournir au réseau de neurones à chaque epochs à la volée (online). Il est à noter que le

² <https://pypi.org/project/sentinelsat/>

³ https://developers.google.com/earth-engine/datasets/catalog/COPERNICUS_S2

⁴ <https://pypi.org/project/opencv-python/>

DataGenerator proposé par Keras⁵ ne peut pas être utilisé pour ce projet, car il ne gère que les images entre 1 et 4 bandes. Un générateur personnalisé prenant en compte les 13 bandes a été mis en place.

Pour augmenter la taille du jeu de données, des techniques d'augmentation de données sont mises en place. Les transformations sur les images sont appliquées dans le générateur et effectuées à la volée avant d'être transmises au réseau de neurones. Les transformations suivantes sont appliquées aux images :

- Rotation d'un angle de 90°, 180° ou 270°
- Inversion verticale ou horizontale des bits de l'image.

Il est à noter comme il s'agit d'image satellite certaine opération d'augmentation de données effectuée traditionnellement sur les images n'ont que peu de sens dans notre cas. Ces opérations sont par exemple :

- La translation
- Le cisaillement (shear en anglais)
- Le zoom
- La rotation avec des angles différents de 90°, 180° ou 270°

5.3. Normalisation

La normalisation des images est appliquée également à la volée à l'aide du générateur. La méthode de normalisation appliquée sur chaque pixel de l'image est Z-norm. Voici la formule utilisée :

$$x_{i,j,k}' = \frac{x_{i,j,k} - \mu_k}{\sigma_k}$$

Avec :

- $x_{i,j,k}'$ la valeur normalisée du pixel (i, j) de la bande k
- $x_{i,j,k}$ le pixel original de coordonnées (i, j) de la bande k
- μ_k la moyenne de toutes les images Eurosat pour la bande k
- σ_k l'écart-type de toutes les images Eurosat pour la bande k

Cependant quand on applique la normalisation avec les paramètres calculés sur le jeu de données Eurosat on constate que les pixels ne sont pas bien centrés autour de 0. Ceci provient du fait que la moyenne et l'écart-type ont été calculés sur les images des paysages européens (jeu de données Eurosat). Pour contrer cela, des expériences ont également été réalisées en utilisant la moyenne et l'écart-type des images Sentinel2 des paysages du Vietnam utilisé pour le projet.

⁵ <https://keras.io/api/preprocessing/image/#imagedatagenerator-class>

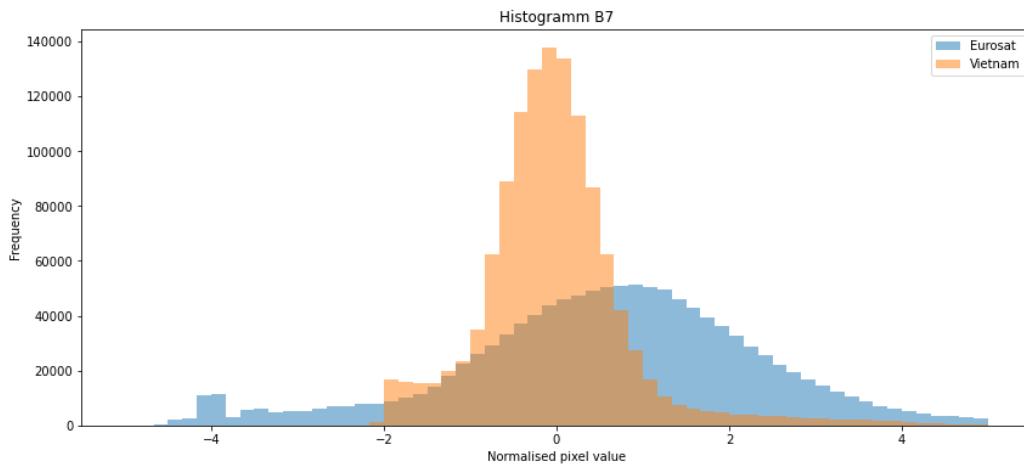


Figure 9 : Histogramme des pixels normalisés avec les paramètres Eurosat et Vietnam pour la bande B7

On constate qu'avec les paramètres du Vietnam les pixels de l'image sont plus centrés autour de 0.

5.4. RestNet-50 Tensorflow Hub

Comme première essaie d'architecture, un modèle ResNet-50 entraîné sur des images Eurosat⁶ a été récupéré sur Tensorflow hub. Ce modèle est alors utilisé pour faire du transfer learning. Il a noté que ce modèle utilise par défaut des images d'entrée de la forme 224x224 avec **3 bandes** seulement, alors qu'idéalement on souhaite exploiter toutes les bandes des images satellites avec une résolution de 64x64 ou 32x32. Ce modèle fonctionne comme une couche de Feature Extraction.

Ce modèle sera toutefois utilisé, étant donné qu'il est possible de préciser la taille des images d'entrée désirée. De plus uniquement les 3 bandes RGB seront utilisées.

Récapitulatif des paramètres initiaux du modèle Resnet50 Tensorflow Hub utilisé :

| | |
|--------------------------|----------|
| Images utilisées | Eurosat |
| Taille des images | 224x224 |
| Nombre de bandes | 3 bandes |

Tableau 6 : Paramètres modèles Tensorflow Hub

L'architecture utilisée pour cette expérience est la suivante :

⁶ https://tfhub.dev/google/remote_sensing/eurosat-resnet50/1

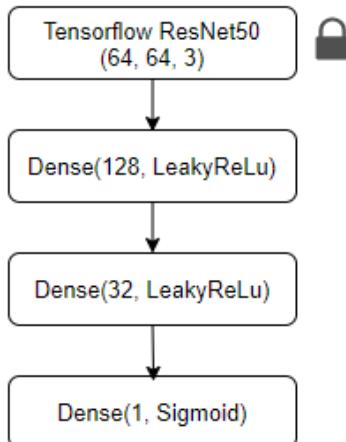


Figure 10 : Architecture modèle Tensorflow Hub

Note :

- Entre chaque couche exceptée, la dernière, un dropout de 0.5 et une régularisation L2 sont appliqués.
- La première couche est utilisée comme une couche de feature extraction et les poids ne sont donc pas modifiés durant la phase d'apprentissage.

Ces différents modèles se trouvent dans le notebook : *TrainingTensorflowHub.ipynb*.

5.5. DenseNet Transfer learning

Dans cette expérience, le modèle de transfer learning du chapitre 3.2 avec les **13 bandes** est utilisé. Des images de taille 64x63 ou 32x32 ont été utilisées pour entraîner ce modèle.

L'architecture utilisée pour cette expérience est la suivante :

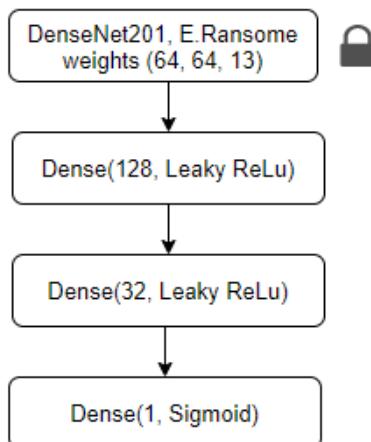


Figure 11 : Architecture modèle transfer learning

Note :

- Entre chaque couche exceptée, la dernière, un dropout de 0.5 et une régularisation L2 sont appliqués.
- La première partie du réseau correspond au modèle 13 bandes d'E.Ransome [2], toutes les couches de cette partie sont gelées durant la phase d'entraînement.
- Les poids du modèle DenseNet201 sont gelés pendant la phase d'entraînement (représenté par le cadenas sur le schéma).

Il est à noter que pour cette expérience les deux dernières couches du modèle de E.Ransome [2] visible sur la figure 2 (Couche de GlobalAveragePooling et couche Dense) ont été supprimées et remplacées par les trois couches présentes sur la figure 11. Les poids de ces trois dernières couches ne sont pas gelés et sont donc modifiés pendant l'apprentissage du modèle.

Ces différents modèles se trouvent dans le notebook : *TrainingTransferMulti.ipynb*.

5.5.1. Variation du nombre de bandes

D'après cet article Medium [6] il apparaît qu'une amélioration des résultats se produit en utilisant différentes combinaisons de bande.

Dans le but de réaliser cette expérience, l'architecture présentée précédemment est utilisée. Une couche de Convolution est ajoutée en entrée de ce réseau dans le but de modifier le nombre de bandes utilisé en entrée. Cette couche est définie comme non gelé pour que le réseau puisse effectuer de manière non supervisée une sélection des bandes. Le nombre de bandes en entrée est plus petit que le nombre de bandes d'entre du réseau DenseNet préentraîné.

Le schéma ci-dessous présent un exemple d'architecture avec 6 bandes en entrée :

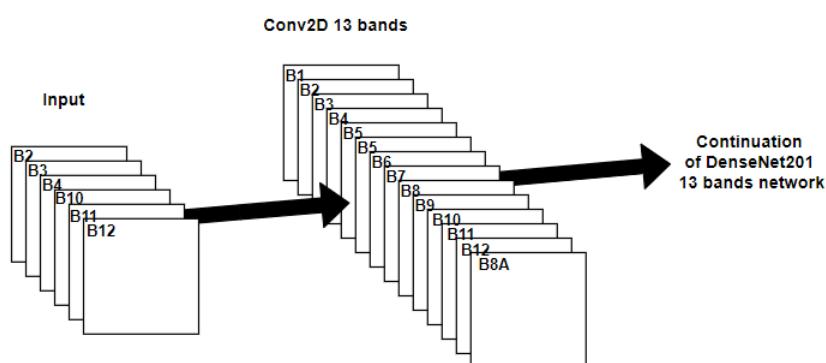


Figure 12 : Exemple architecture couche “trainable” 13 bandes

Des expériences alternatives ont été réalisées en utilisant le modèle de transfer learning préentraîné d'E.Ransome [2], mais cette fois-ci entraînées sur des images RGB (3 bandes). De cette manière, le nombre de bandes en entrée est plus grand que le nombre de bandes d'entre du réseau DenseNet préentraîné.

Le schéma ci-dessous présent un exemple d'architecture avec 6 bandes en entrée :

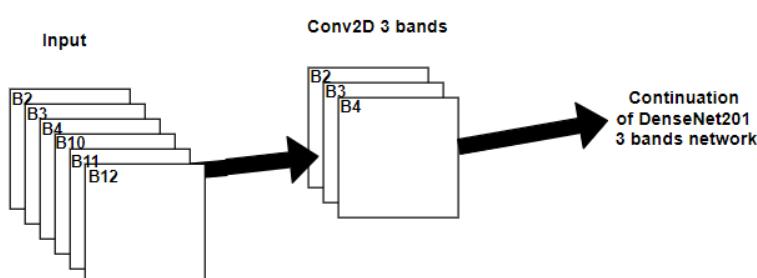


Figure 13 : Exemple architecture couche “trainable” RGB 3 bandes

- Énumération des solutions développées :

- DenseNet 13 bande :
 - Bande RGB (B2, B3, B4) + SWIR (B10, B11 et B12)
 - Bande haute résolution (B2, B3, B4, B5, B6, B7, B8, B11, B12, B8A)
- DenseNet 3 bande :
 - Bande RGB (B2, B3, B4) + SWIR (B10, B11 et B12)
 - Bande haute résolution (B2, B3, B4, B5, B6, B7, B8, B11, B12, B8A)
 - 13 Bandes

Ces différents modèles se trouvent dans le notebook : *TrainingBandVariation.ipynb*.

5.6. Focal loss

Par défaut, les expériences ont été réalisées avec la fonction de coût **Binary Cross Entropy**. Comme le jeu de données est non balancé entre les classes, la fonction de coût **Focal Loss** [8] a été utilisée. Dans le cadre d'une classification avec des images, la fonction de coût Focal Loss réduit l'importance des échantillons bien classés et se focalise sur les échantillons difficiles à classer. La valeur de la perte est beaucoup plus élevée pour un exemple mal classé par rapport à la perte d'un exemple bien classé par le modèle.

Une expérience est donc réalisée pour comparer les résultats du modèle avec la fonction de coût Focal Loss contre Binary Cross Entropy.

5.7. Dégel de toutes les couches du réseau

Une étape optionnelle du transfer learning consiste à réentraîner le modèle précédemment entraîné, mais en dégelant toutes les couches et en utilisant un learning rate très petit. Pour cette expérience un learning rate de **1e-7** est utilisé.

Cette étape a été réalisée à partir du modèle de transfer learning précédemment entraîné, c'est-à-dire avec la base du réseau gelé et les 3 couches supérieures dégelées. Dans ce cas-ci, toutes les couches dégelées. Cette expérience doit permettre de modifier très légèrement les poids du réseau de base pour les adapter à nos données et améliorer légèrement les performances. L'architecture utilisée pour cette expérience se présente comme ceci :

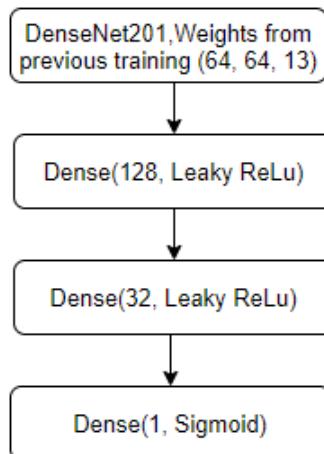


Figure 14 : Architecture modèle transfer learning – toutes couches dégelées

Note :

- Entre chaque couche exceptée, la dernière, un dropout de 0.5 et une régularisation L2 sont appliqués.

- Le réseau est initialisé avec les poids obtenus lors des précédentes expériences, notamment celle du chapitre 5.5 *DenseNet Transfer learning*.

5.8. Culture vs non-culture

Une expérience réalisée en parallèle avec les différentes méthodologies présentées ci-dessus est d'effectuer une classification pour différentier les images satellites contenant des cultures contre celle n'en contenant pas.

Les classes considérées comme **culture** sont :

- Cacao
- Café
- Caoutchouc
- Poivre
- Thé
- Riz
- Banane

Les classes considérées comme **non-culture** sont :

- Végétation native
- Urbain
- Eau
- Forêt feuillue
- Arbre de pin

Les mêmes architectures de modèle que précédemment ont été utilisées.

5.9. Modèle multi labels

Comme présenté au chapitre 7. Résultats, on constate que les modèles culture contre non-culture obtiennent de meilleurs résultats que les modèles café contre reste. Une idée est alors d'entraîner un modèle à prédire en même temps si :

- Une image est un champ de café ou non
- Une image est une culture ou non

On peut imaginer que comme le modèle arrive bien à classifié les cultures contre non-culture, le modèle arrive à « partager » certains poids pour également classifier le café contre le reste.

Pour effectuer cette expérience, le modèle de transfer learning DenseNet201 à été utilisé. Contrairement aux autres expériences, 2 neurones sont présents sur la couche de sortie. La fonction d'activation **Sigmoid** est utilisée, car les labels de sortie ne sont pas mutuellement exclusifs.

La capture ci-dessous présente l'architecture utilisée :

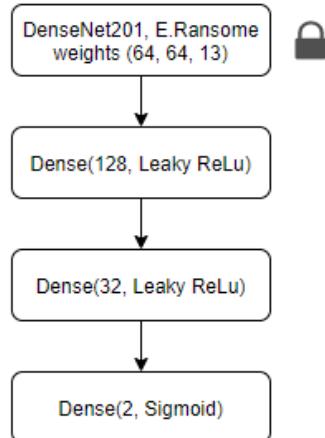


Figure 15 : Architecture modèle multi labels

6. Environnement d'exécution

Les différents modèles ont été exécutés sur le serveur trex de la HEIGVD. Le serveur possède les caractéristiques suivantes :

- **Carte graphique** : NVIDIA GeForce GTX 1080
- **Processeur** : Intel(R) Core(TM) i7-6700K CPU @ 4.00GHz, x86_64, 8 cœurs
- **Mémoire RAM** : 64 Gb

7. Résultats

Les résultats détaillés des modèles se trouvent dans le notebook : *ModelComparison.ipynb*. Les graphiques de l'accuracy et de la loss par epoch pour chaque fold se trouvent dans l'Annexe 2.

Pour comparer les modèles pendant la validation croisée, les deux métriques suivantes ont été utilisées :

- Accuracy
- Loss

Les boxplot ci-dessous ont été obtenus en entraînant les modèles avec la validation croisée sur **5 folds**. Les boxplots sont réalisés à partir des scores obtenus sur chacune de ces folds.

Pour effectuer les différentes expériences, les hyperparamètres suivants ont été utilisés (excepté indication contraire) :

- **Batch size** : 64
- **Epoch** : 100
- **Learning rate**
 - Café contre reste : 1e-5
 - Culture contre non-culture : 1e-4
- **Fonction de coût** : Binary Cross Entropy
- **Optimisateur** : Adam

Il est à noter qu'un premier tri des modèles est d'abord effectué à partir des métriques calculées pendant la validation croisée. Sur cette base, les modèles les plus prometteurs sont sélectionnés et évalués sur le jeu de test.

Pour l'évaluation sur le jeu de test, la métrique principale utilisée est le F1-Score qui a l'avantage notamment de prendre en compte le non-balancement des classes et favorise les classificateurs avec une précision et un rappel similaire. Pour une classe C la formule du F1-Score est la suivante :

$$F1 = \frac{TP_C}{TP_C + \frac{1}{2}(FP_C + FN_C)}$$

Où :

- TP_C : nombre de vrais positifs
- FP_C : nombre de faux positifs
- FN_C : nombre de faux négatifs

Pour l'évaluation sur le jeu de test, le F1-Score est calculé pour les deux classes. Cependant pour comparer les modèles entre eux uniquement le score pour la classe café est retenu et pour l'expérience culture non-culture uniquement la classe culture.

Une variante du F1 Score nommé Macro F1-Score est également utilisée. Cette métrique calcule le F1-Score pour chaque classe puis effectue la moyenne arithmétique. Dans le cas d'une classification binaire, la formule peut s'écrire comme ci :

$$MACRO - F1 = \frac{F1_{class0} + F1_{class1}}{2}$$

Cette métrique est utilisée pour donner un poids égal aux deux classes peu importe le balancement des classes (le café aura donc la même importance que la classe reste).

7.1. Cross validation jeu de données stratifié

7.1.1. Café contre reste

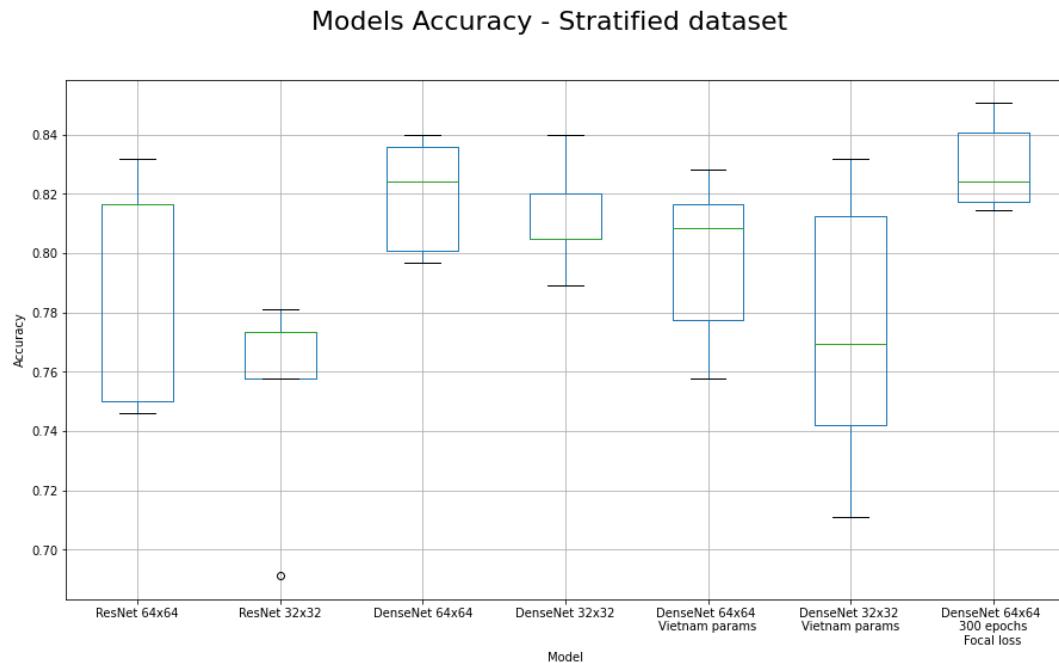


Figure 16 : Café contre autre - Accuracy modèles validation croisée – stratifié

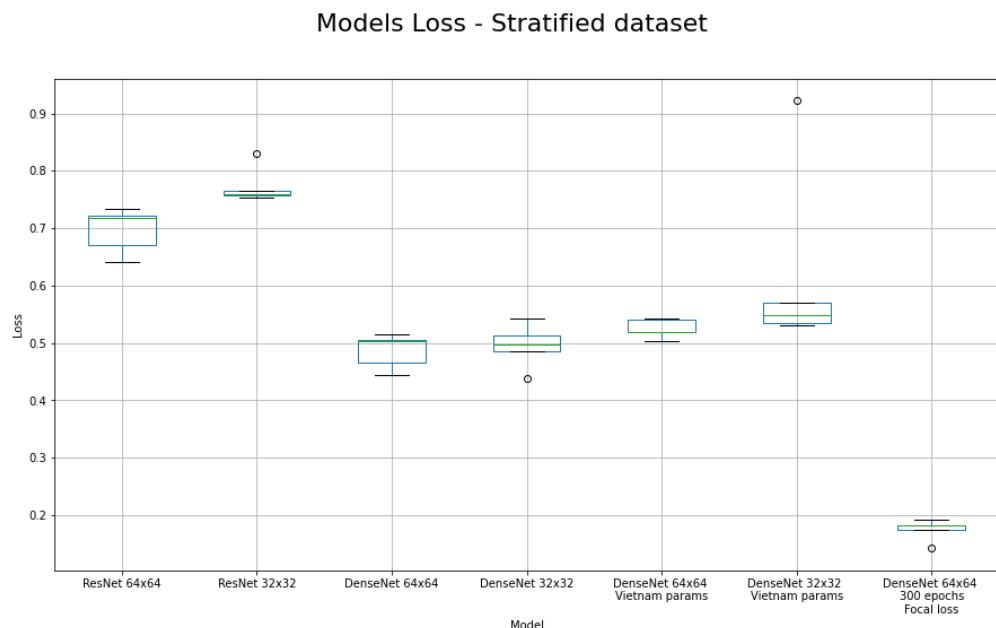


Figure 17 : Café contre autre - Loss modèles validation croisée – stratifié

Explication des modèles sur les graphiques de gauche à droite :

- Modèle Feature Extraction Tensorflow hub ResNet50, images 64x64
- Modèle Feature Extraction Tensorflow hub ResNet50, images 32x32
- Modèle Transfer Learning DenseNet201, images 64x64
- Modèle Transfer Learning DenseNet201, images 32x32
- Modèle Transfer Learning DenseNet201, images 64x64, normalisation z-norm avec paramètres calculés sur le jeu de données (5.3 Normalisation)

- Modèle Transfer Learning DenseNet201, images 32x32, normalisation z-norm avec paramètres calculés sur le jeu de données (5.3 Normalisation)
- Modèle Transfer Learning DenseNet201, images 64x64, 300 epochs, fonction de coût Focal Loss

Après les résultats ci-dessus, on constate que le modèle DenseNet201 entraîné sur les images de taille 64x64 donne les meilleurs scores. Un autre modèle a été entraîné avec la même architecture et les mêmes hyperparamètres ont été utilisés excepté que le modèle est entraîné sur :

- 300 epochs
- La fonction de coût Focal Loss.

On constate que ce dernier modèle obtient les meilleurs scores pour la phase de validation croisée.

Résultats sur le jeu de test

Les deux modèles suivants ont été évalués sur le jeu de test (de gauche à droite):

- Modèle Transfer Learning DenseNet201, images 64x64
- Modèle Transfer Learning DenseNet201, images 64x64, 300 epochs, fonction de coût Focal Loss

On constate que le modèle qui n'utilise pas la focal loss obtient de meilleures performances, malgré le fait que le modèle avec focal loss ait une meilleure performance lors de la validation croisée.

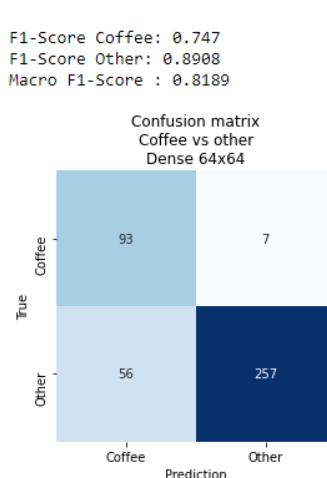


Figure 18 : Score – Café – DenseNet 64x64

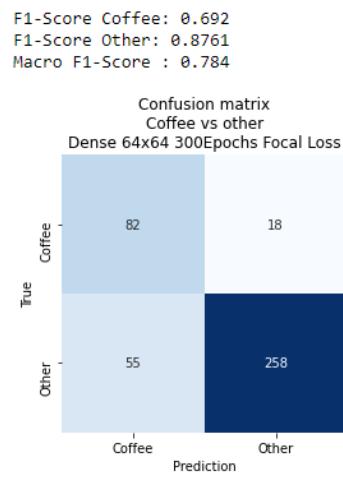


Figure 19 : Score – Café – DenseNet 64x64
Focal loss

Pour cette expérience, le meilleur modèle obtient un F1-Score de : **74.7%** et un Macro F1 de : **81.89%**.

7.1.2. Culture contre non-culture

Models Accuracy - Stratified dataset

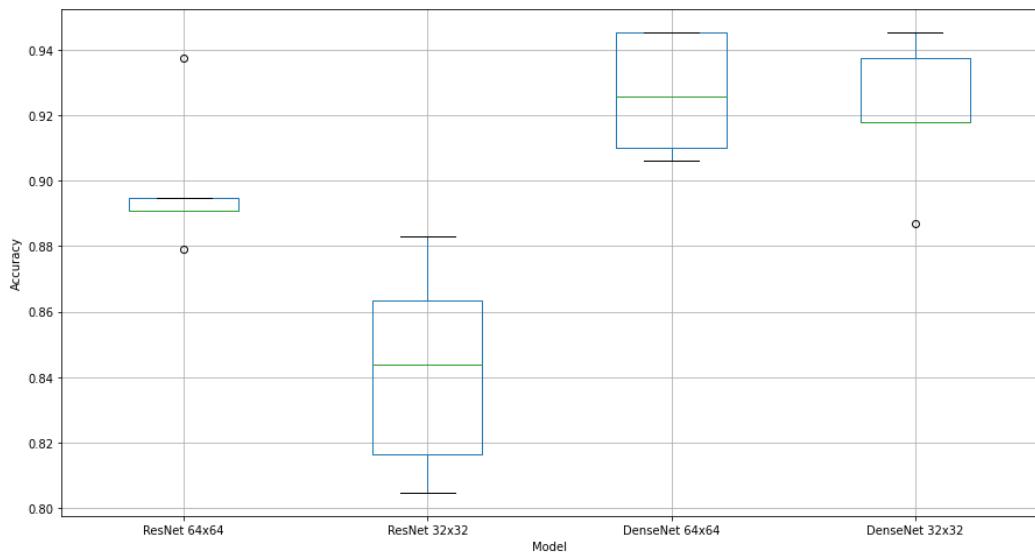


Figure 20 : Culture contre non-culture – Accuracy modèles validation croisée - stratifié

Models Loss - Stratified dataset

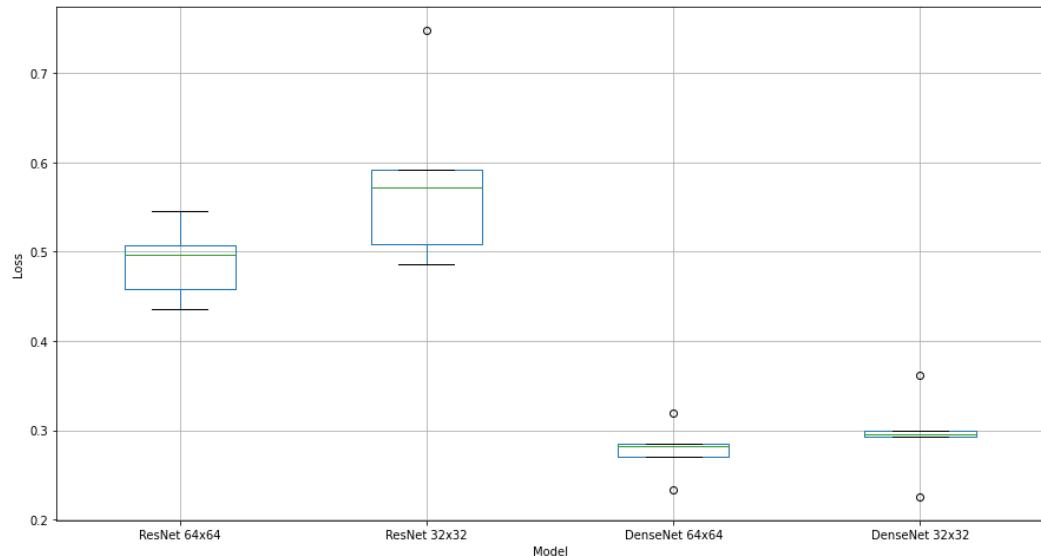


Figure 21 : Culture contre non-culture - Loss modèles validation croisée - stratifié

Explication des modèles sur les graphiques de gauche à droite :

- Modèle Feature Extraction Tensorflow hub ResNet50, images 64x64
- Modèle Feature Extraction Tensorflow hub ResNet50, images 32x32
- Modèle Transfer Learning DenseNet201, images 64x64
- Modèle Transfer Learning DenseNet201, images 32x32

Les boxplots sont obtenus avec la même méthodologie que les modèles café contre le reste. Les mêmes hyperparamètres ont été utilisés. Comme pour le café contre le reste, le modèle DenseNet201 obtient les meilleurs résultats.

Résultats sur le jeu de test

Résultat pour le modèle DenseNet201 64x64 :

F1-Score Culture: 0.9349
 F1-Score No Culture: 0.9194
 Macro F1-Score : 0.9272

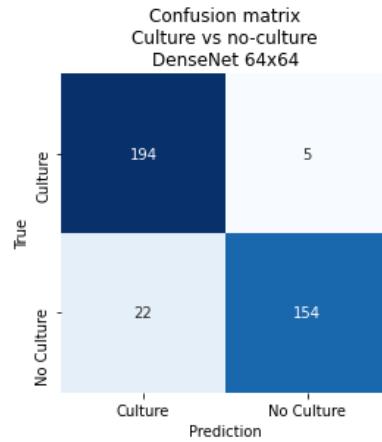


Figure 22 : Score – Culture – DenseNet 64x64

Pour cette expérience, le meilleur modèle obtient un F1-Score de : **93.49%** et un Macro F1 de : **92.72%**.

7.2. Cross validation géographique

7.2.1. Café contre reste

Models Accuracy - Spatial dataset

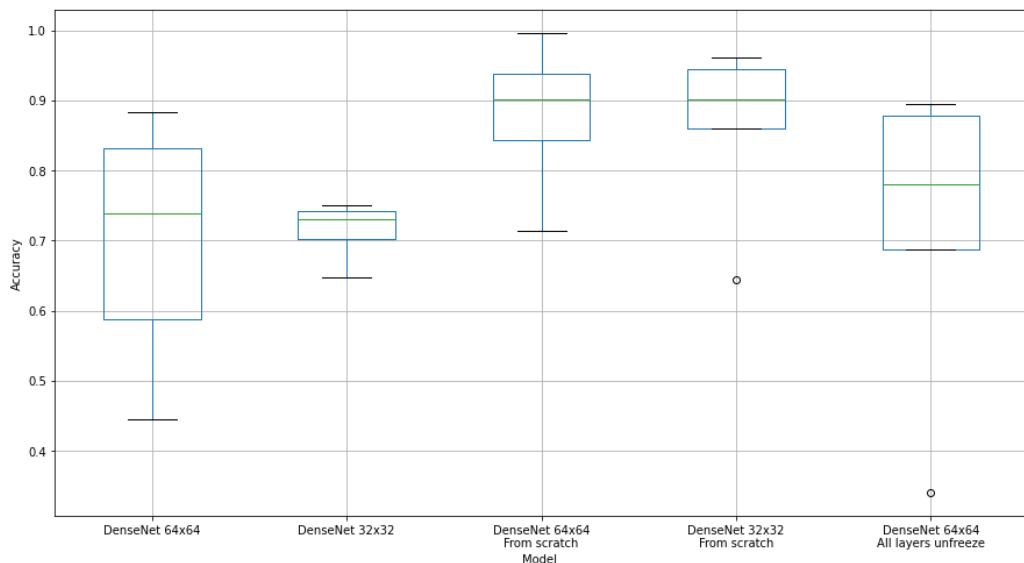


Figure 23 : Café contre autre - Accuracy modèles validation croisée – spatial

Models Loss - Spatial dataset

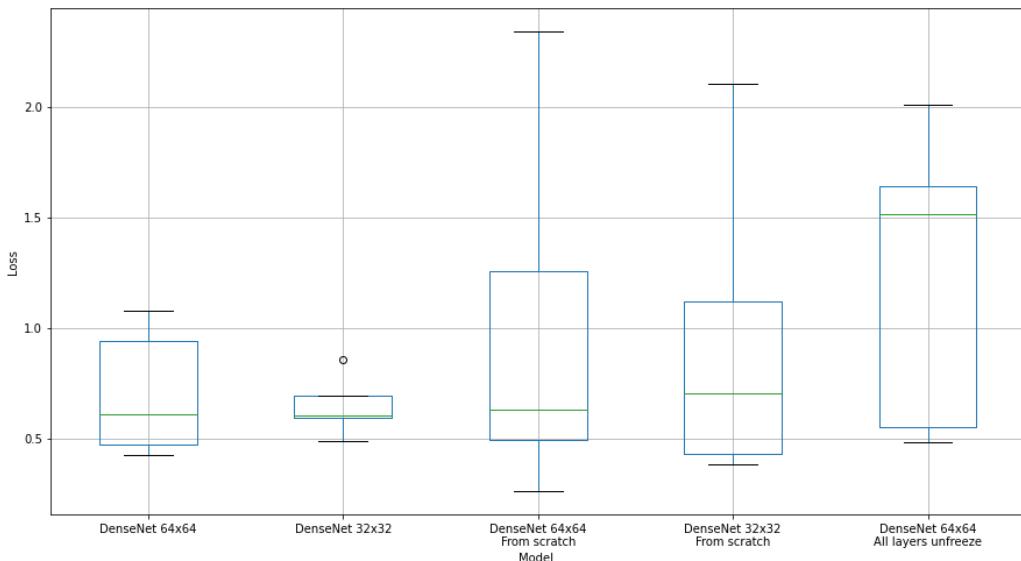


Figure 24 : Café contre autre - Loss modèles validation croisée – spatial

Explication des modèles sur les graphiques de gauche à droite :

- Modèle Transfer Learning DenseNet201, images 64x64
- Modèle Transfer Learning DenseNet201, images 32x32
- Modèle DenseNet201 From Scratch, images 64x64
- Modèle DenseNet201 From Scratch, images 32x32
- Modèle DenseNet201 avec toutes les couches dégelées, image 64x64

Résultats sur le jeu de test

Les trois modèles suivants ont été évalués sur le jeu de test (de gauche à droite):

- Modèle Transfer Learning DenseNet201, images 64x64
- Modèle From Scratch DenseNet201, images 64x64
- Modèle DenseNet201 Fine tuner, images 64x64

Le modèle entraîné « from scratch » obtient les moins bons résultats. Les deux modèles restants ont un F1-Score presque équivalent, mais le modèle entraîné avec toutes les couches dégelées obtient un Macro F1-Score supérieur.

F1-Score Coffee: 0.4559
F1-Score Other: 0.6216
Macro F1-Score : 0.5387

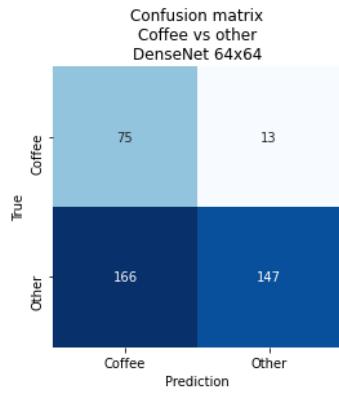


Figure 26 : Score – Café –
DenseNet 64x64 – Spatial

F1-Score Coffee: 0.1424
F1-Score Other: 0.4217
Macro F1-Score : 0.2821

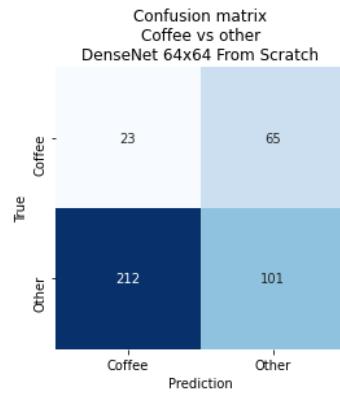


Figure 25 : Score – Café –
DenseNet 64x64
From Scratch - Spatial

F1-Score Coffee: 0.4549
F1-Score Other: 0.7459
Macro F1-Score : 0.6004

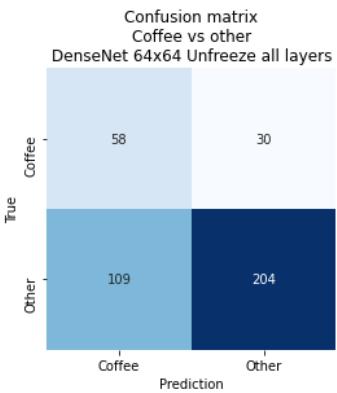


Figure 27 : Score – Café –
DenseNet 64x64
Toutes couches dégelées –
Spatial

Le meilleur F1-Score obtenu pour cette expérience est de : **45.59%** et un Macro F1 de : **60.04%**.

7.2.2. Culture contre non-culture

Models Accuracy

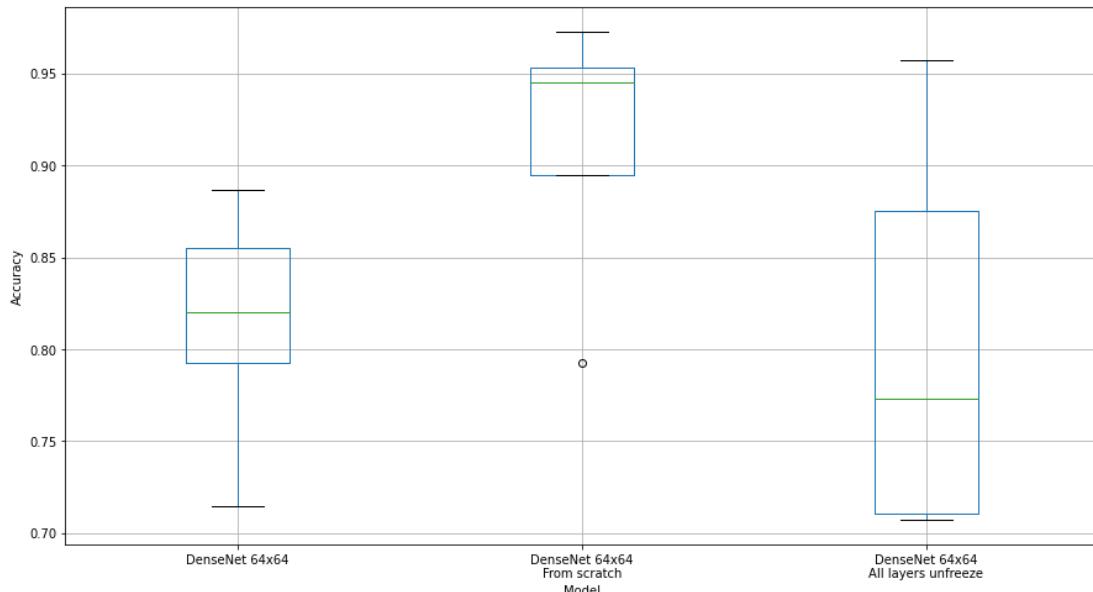


Figure 28 : Culture contre non-culture - Accuracy modèles validation croisée – spatial

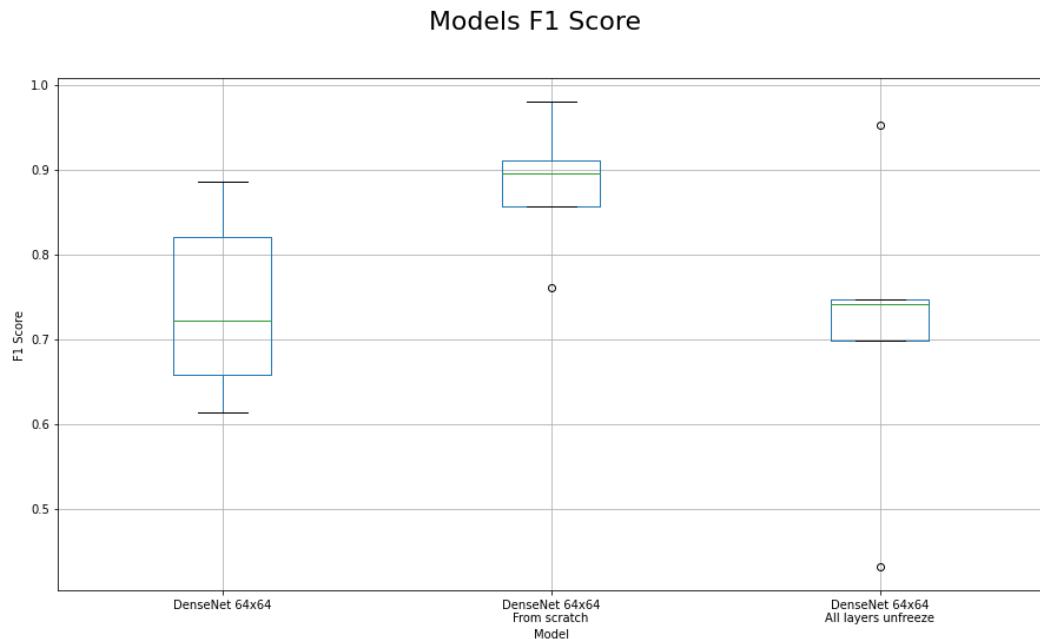


Figure 29 : Culture contre non-culture - Loss modèles validation croisée – spatial

Explication des modèles sur les graphiques de gauche à droite :

- Modèle Transfer Learning DenseNet201, images 64x64
- Modèle DenseNet201 From Scratch, images 64x64
- Modèle DenseNet201 Toutes couches dégelées, images 64x64

Résultats sur le jeu de test

Les deux modèles suivants ont été évalués sur le jeu de test (de gauche à droite):

- Modèle Transfer Learning DenseNet201, images 64x64
- Modèle From Scratch DenseNet201, images 64x64

Le modèle entraîné « from scratch » obtient les moins bonnes performances, suivi du modèle entraîné avec toutes les couches dégelées et finalement le modèle de transfer learning standard qui obtient les meilleures scores.

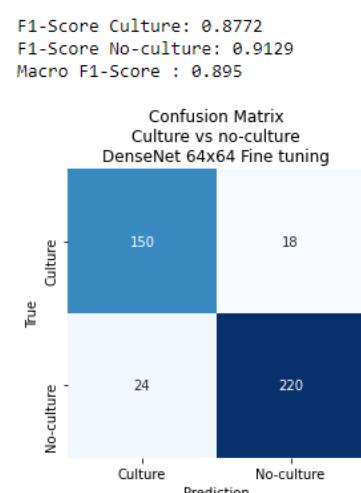
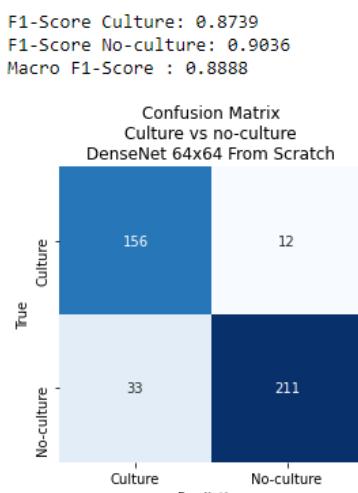
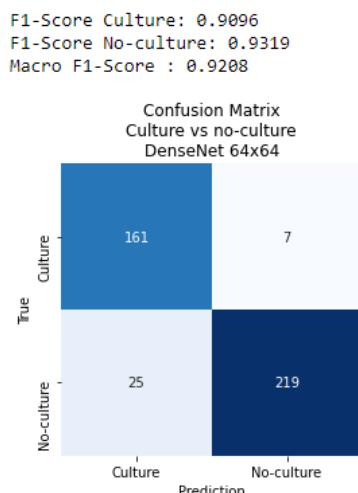


Figure 30 : Score – Culture –
DenseNet 64x64 –
Spatial

Figure 31 : Score – Culture –
DenseNet 64x64 From scratch –
Spatial

Figure 32 : Score – Culture –
DenseNet 64x64 Fine tuning –
Spatial

Le meilleur F1-Score obtenu pour cette expérience est de : **90.96%** et un Macro F1 de : **92.08 %**.

7.3. Variation de bandes

7.3.1. DenseNet 3 bandes

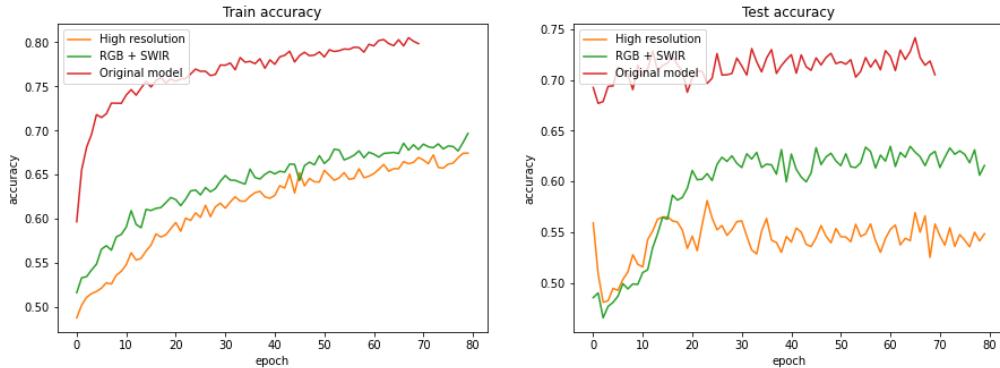


Figure 33 : Variation de bandes – DenseNet 3 bandes accuracy

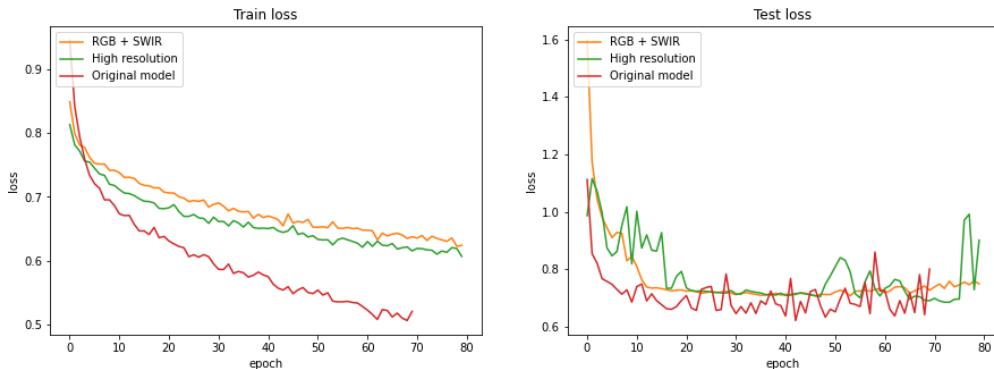


Figure 34 : Variation de bandes – DenseNet 3 bandes loss

Le modèle de base utilisé pour la comparaison est le modèle DenseNet201 entraîné sur les images de taille 64x64. Au vu des résultats obtenus, les modèles ne sont pas entraînés avec la validation croisée.

7.3.2. DenseNet 13 bandes

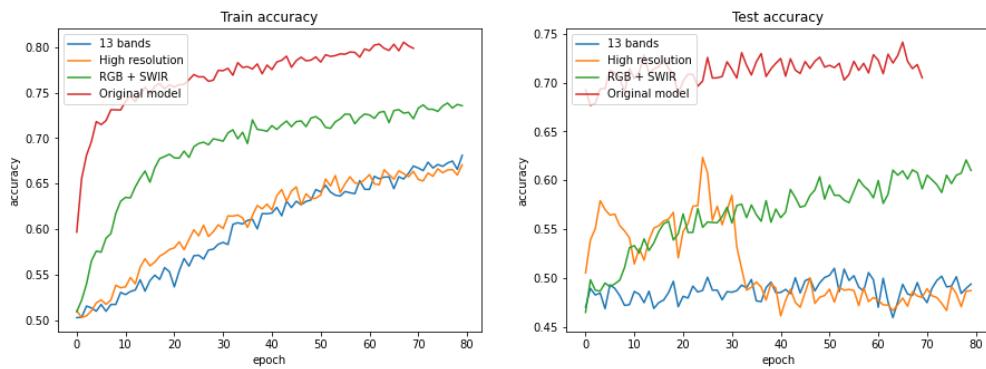


Figure 35 : Variation de bandes – DenseNet 13 bandes accuracy

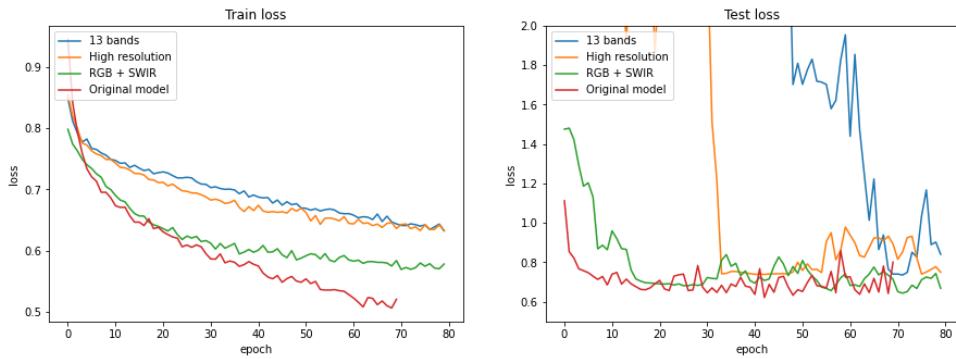


Figure 36 : Variation de bandes – DenseNet 13 bandes loss

Le modèle de base utilisé pour la comparaison est le modèle DenseNet201 entraîné sur les images 64x64. Au vu des résultats obtenus, les modèles ne sont pas entraînés avec la validation croisée.

7.1. Modèle multi labels

Pour cette expérience uniquement la séparation géographique des données est utilisée. Le modèle décrit au chapitre 5.9 Modèle *multi labels* est utilisé.

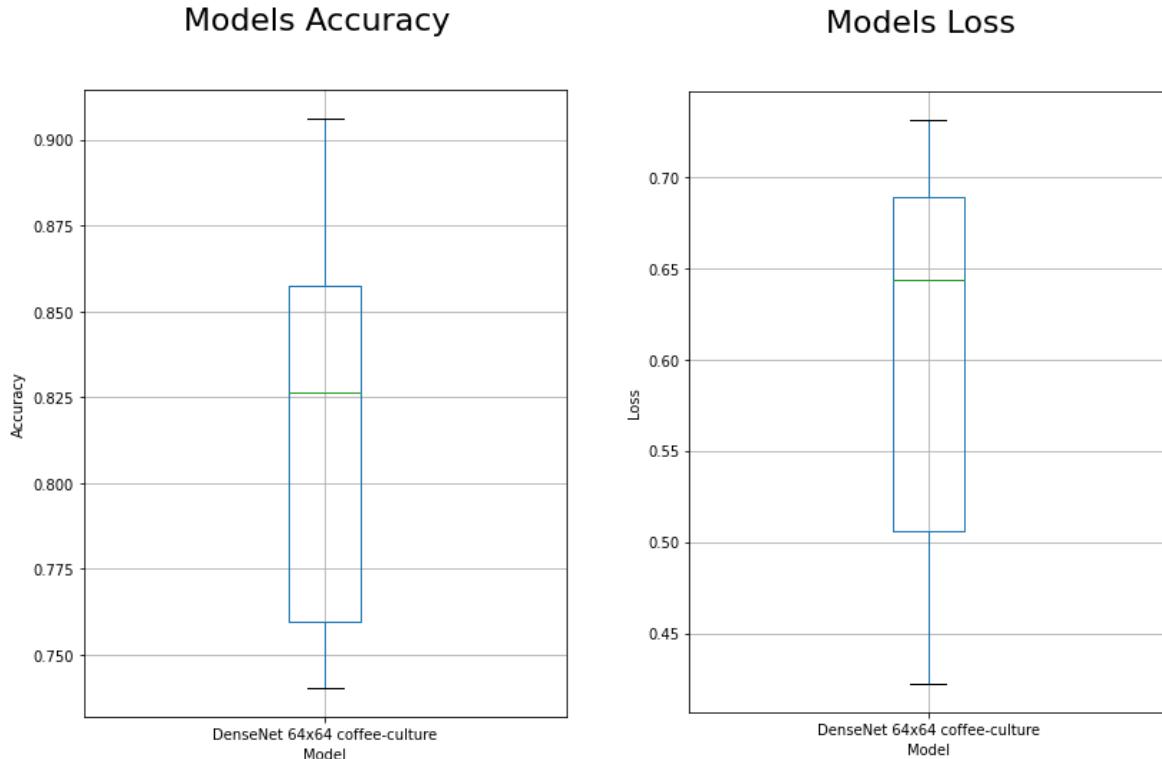


Figure 38 : Multi labels –
Accuracy validation croisée – spatial

Figure 37 : Multi labels –
Loss validation croisée – spatial

Résultats sur le jeu de test

Culture vs no-culture
 F1-Score Culture: 0.6708
 F1-Score No culture: 0.5597
 Macro F1-Score : 0.6152

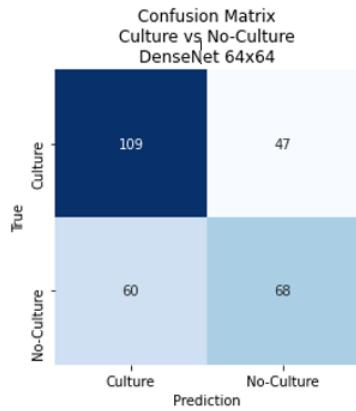


Figure 39 : Score – Multi labels culture – DenseNet 64x64 –Spatial

Coffee vs other
 F1-Score Coffee: 0.2991
 F1-Score Other: 0.8373
 Macro F1-Score : 0.5682

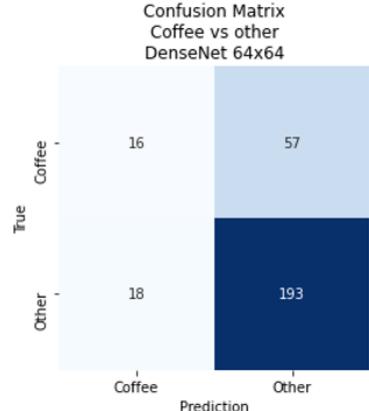


Figure 40 : Score – Multi labels café – DenseNet 64x64 –Spatial

- La classification culture contre non-culture obtient un F1-Score de : **67.08%** et un Macro F1-Score de : **61.52%**.
- La classification café contre reste obtient un F1-Score de : **29.91%** et un Macro F1-Score de : **56.82%**.

8. Discussion

8.1. Variation de bandes

On constate que cette expérience donne de moins bons résultats que le modèle original (DenseNet201 64x64 en transfer learning). L'accuracy obtenue est inférieur au modèle de base. Ces différentes solutions n'ont donc pas été retenues et pas entraînées avec la validation croisée. Ces résultats peuvent probablement s'expliquer par le fait que le modèle est entraîné à recevoir soit des images avec 3 ou 13 bandes et qu'il n'arrive pas bien généraliser sur des images avec un autre nombre de bandes.

8.2. Café contre reste

À partir des résultats sur la validation croisée, on constate que les modèles DenseNet201 (13 bandes) performent mieux que les modèles ResNet50 (3 bandes) autant pour les images de tailles 64x64 que 32x32. On peut donc conclure que les modèles tirent donc plus d'informations sur les images avec 13 bandes que les images avec 3 bandes et que l'architecture DenseNet201 semble mieux convenir à notre tâche.

Les modèles avec les images de taille 64x64 donnent légèrement de meilleurs résultats que les images de taille 32x32. Pour le modèle ResNet50 de Tensorflow Hub, ceci provient sûrement du fait que le modèle de base est entraîné avec des images de taille 224x224. Les images de taille 64x64 sont donc plus proches du modèle initial que les images 32x32. Pour le modèle DenseNet201, les meilleures performances peuvent venir du fait que le modèle de préentraîné est entraîné sur des images 64x64.

Les modèles DenseNet201 utilisant la normalisation z-norm avec la moyenne et l'écart-type calculés sur toutes les images Eurosat performe mieux que le modèle avec la moyenne et l'écart-type calculés sur les images du jeu de données du Vietnam. Ces résultats peuvent s'expliquer par le fait que le modèle performe mieux avec les paramètres sur lequel il a été entraîné initialement.

Pour les deux modèles retenus pour une évaluation sur le jeu de test, on constate que le modèle entraîné sur 300 epochs avec la Focal Loss donne de moins bons résultats que le modèle de transfer learning DenseNet201 de base. La focal loss ne semble donc pas plus aider les classes sous représentées que l'attribut *class_weight* de la méthode *fit()* de Keras utilisé sur les autres expériences.

Le meilleur F1-Score obtenu pour cette expérience est de **74.7%**.

8.2.1. Séparation géographique

À partir des résultats sur le jeu de test, on remarque que, le modèle entraîné de manière « from scratch » donne de moins bons résultats que le modèle de transfer learning classique. Au vu de la comparaison des scores obtenus lors de la validation croisée et lors du test on peut dire que le modèle « from scratch » surajuste. Ceci est cohérent avec le fait que le modèle a énormément de paramètres (19,340,673) et est entraîné sur très peu d'images.

Le modèle de transfer learning entraîné avec toutes les couches dégelées obtient un F1-Score similaire, mais un Macro F1-Score supérieur d'environ **6%** par rapport au modèle de base. De plus, la matrice de confusion semble indiquer une meilleure séparation des classes.

Les modèles entraînés avec une séparation géographique donnent des résultats bien inférieurs au modèle entraîné avec une séparation aléatoire stratifiée. Au vu des résultats des trois modèles sur le jeu de test, on peut

donc dire que cette expérience n'est pas concluante et que les modèles ne peuvent pas être utilisés de manière fiable pour prédire si une image contient un champ de café ou non. Il est également important de noter que les scores obtenus varient beaucoup d'une fold à l'autre (cf. Annexe 2). L'écart-type des scores entre les folds pour cette expérience est élevé

On peut donc vraisemblablement dire que les modèles entraînés sur le jeu séparé aléatoirement utilisent le fait que certaines images géographiquement proches sont très similaires. Cependant les folds sont assez déséquilibrés en défaveur du café ce qui peut expliquer cette différence dans les scores entre les deux expériences.

Le meilleur F1-Score obtenu avec la séparation géographique est de **45.59%**. On constate donc une très grande différence de performances entre les modèles entraînés avec un jeu de données séparé aléatoirement et un jeu de donnée séparé géographiquement.

On constate que malgré toutes les expériences réalisées pour le café contre le reste, les modèles n'arrivent pas à dépasser un F1-Score de **74.7%**. Ceci peut s'expliquer par le fait que les images de café sont visuellement très proches des autres images de culture comme le montre la capture ci-dessus. La variabilité interclasses est faible. Même un humain aurait de la peine à classifier les images dans la bonne catégorie. Cependant, cette remarque est valable uniquement avec l'utilisation des trois bandes RGB, le modèle peut tirer d'autres informations des dix bandes restantes invisibles pour un humain.

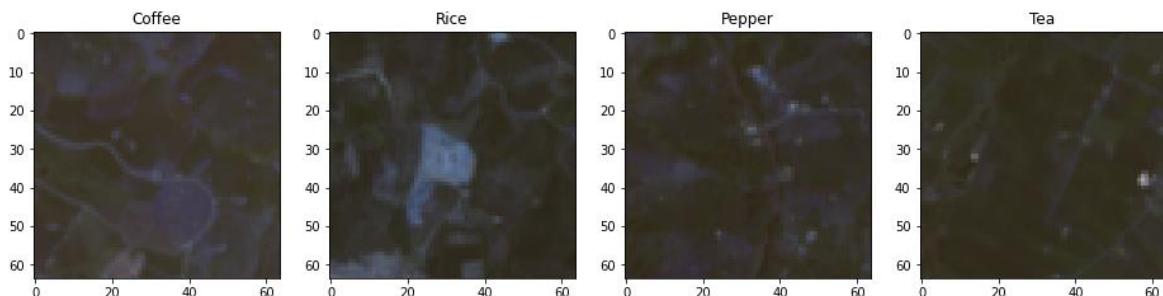


Figure 41 : Comparaison images – Café contre reste

On constate de plus sur la capture ci-dessous que les images d'Eurosat qui ont permis d'entraîner le modèle utilisé pour le transfer learning ont une grande variabilité interclasses. Ceci peut expliquer pourquoi les modèles entraînés sur les images Eurosat obtiennent de meilleurs résultats (cf Chapitre 2.1). De plus dans le cas d'Eurosat les classes sont équilibrées contrairement à notre cas.



Figure 42: Images Eurosat

8.3. Culture contre non-culture

Avec les scores obtenus lors de la validation croisée, les mêmes conclusions peuvent être tirées que pour les modèles café contre reste. Les modèles DenseNet201 performent mieux que les modèles ResNet50. Les modèles avec des images de taille 64x64 performent mieux que des modèles avec des images de taille 32x32.

Comme précédemment l'expérience de normalisation des images avec la moyenne et l'écart-type des images du Vietnam n'a pas donné de bon résultat, cette expérience n'est pas réitérée ici.

Lors de l'évaluation sur le jeu de test, le modèle de transfer learning DenseNet201 avec les images de taille 64x64 obtient un F1-Score de **93.49%**.

8.3.1. Séparation géographique

À partir des résultats sur le jeu de test, on remarque que le modèle entraîné « from scratch » obtient de moins bons résultats sur le jeu de test que le modèle de transfer learning, malgré les meilleurs résultats obtenus lors de la validation croisée. On peut donc conclure que comme précédemment le modèle « from scratch » surajuste.

En comparaison à l'expérience café contre reste on constate que l'entraînement du modèle avec toutes les couches dégelées n'a pas amélioré les performances du modèle initial.

On remarque que comparé à l'expérience café contre reste, les modèles culture non-culture entraînées avec un jeu de données séparé géographiquement obtiennent de bons résultats comparés à la séparation aléatoire. Ceci s'explique par le fait que dans cette expérience les images présentent une plus grande variabilité interclasse comme le montre la capture ci-dessous (plus proche de celle présente dans le jeu de données Eurosat).

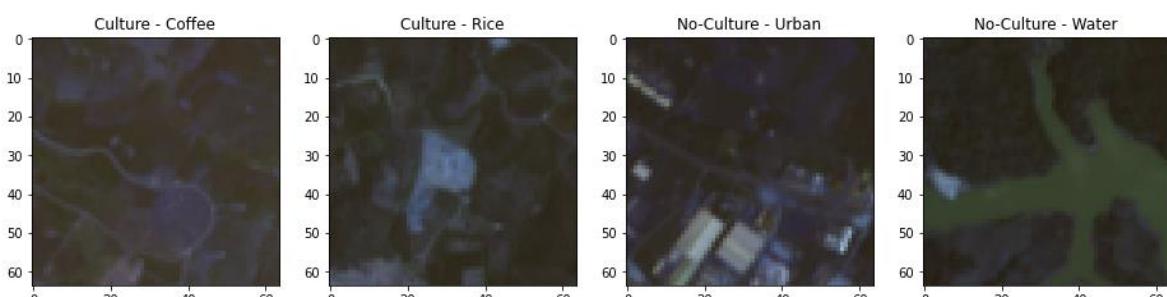


Figure 43 : Comparaison images – Culture contre non culture

On peut donc remettre en cause la conclusion faite précédemment qui dit que le modèle utilise probablement le fait que les images proches géographiquement sont assez similaires. Il semble que les mauvais résultats obtenus sur l'expérience du café proviennent du déséquilibre des classes et de la faible variabilité interclasse.

Le meilleur F1-Score obtenu pour cette expérience est de : **90.96%** avec le modèle de transfer learning DenseNet201 64x64.

8.4. Modèle multi labels

Pour le café contre le reste, on constate que cette expérience n'obtient pas de meilleurs résultats qu'observé précédemment. Le F1-Score est faible et en inspectant la matrice de confusion on constate que le modèle a tendance à classifier tous les champs de café dans la classe autre.

Pour l'expérience culture contre non-culture, on constate que le score est inférieur à toutes les expériences réalisées précédemment.

Ces résultats peuvent s'expliquer par le fait que lors de la création des folds géographique, il est très compliqué de respecter le balancement des classes à la fois pour le café contre le reste et pour culture contre non-culture.

Cette expérience n'est donc pas retenue au vu de ces résultats obtenus.

8.5. Comparaison des scores

Le but de ce chapitre est de comparer les scores obtenus dans ce projet avec ceux de J. Rod[1]. Pour ce chapitre uniquement les scores des modèles entraînés avec la séparation aléatoire sont retenus. Il n'y a pas d'intérêt à comparer les scores de la séparation géographique, car J.Rod[1], n'a pas utilisé cette méthodologie.

Pour rappel les meilleurs scores obtenus par J.Rod[1] sont :

- F1-Score de **76.21%** sur les images Sentinel 1C.
- F1-Score de **86.66%** sur les images Sentinel 2A.

Les scores obtenus dans ce projet pour les modèles entraînés avec la séparation aléatoire sont :

- F1-Score de **74.7%** et Macro F1-Score de **81.89%** pour café contre reste.
- F1-Score **93.49%** et Macro F1-Score de **92.72%** pour culture contre non-culture.

On constate que pour l'expérience café contre reste J.Rod[1] obtient un F1-Score légèrement supérieur pour les images de type 1C. En utilisant le Macro F1-Score, le modèle entraîné durant ce projet obtient de meilleurs résultats pour les images Sentinel 1C. Cette différence de résultats peut s'expliquer par deux points :

- J.Rod[1] utilise des images de taille 15x15 les images représentent donc une plus petite zone que les images de taille 64x64 qui peuvent contenir des informations superflues.
- En découpant les images avec une taille 15x15 on obtient plus d'images qu'avec une taille de 64x64 ce qui peut aider le modèle à mieux généraliser.

Pour finir, on constate que l'expérience **culture contre non-culture** obtient de meilleures performances comparées à toutes les autres expériences. On peut donc conclure que les modèles arrivent donc mieux à classifier les images cultures non-cultures que café contre reste.

9. Récapitulatif des résultats

Café contre reste :

| Modèle | Jeu de données | F1-Score |
|--|---------------------------------|---------------|
| DenseNet201 transfer learning, 64x64 | Séparation stratifiée aléatoire | 74.7% |
| DenseNet201 toutes les couches dégelées, 64x64 | Séparation géographique | 45.59% |

Tableau 7 : Meilleurs scores - Café

Culture contre non-culture :

| Modèle | Jeu de données | F1-Score |
|--------------------------------------|---------------------------------|---------------|
| DenseNet201 transfer learning, 64x64 | Séparation stratifiée aléatoire | 93.49% |
| DenseNet201 transfer learning, 64x64 | Séparation géographique | 90.96% |

Tableau 8 : Meilleurs scores – Culture

10. Conclusion

Pour conclure, on constate que la classification de champs de café n'est pas une tâche triviale. Plusieurs raisons expliquent cela, les champs de café sont visuellement très proches des autres cultures, la variabilité interclasses est donc faible. De plus, le nombre de champs de café est déséquilibré par rapport à la classe autre. Ce projet a également montré que les méthodes utilisées pour la séparation du jeu de données sont importantes et peut amener à des résultats très variables en fonction des techniques utilisées. En effet, le meilleur modèle avec une séparation des données aléatoires obtient un F1-Score de **74.7%** contre **45.59%** pour la séparation géographique, ce qui est une différence notable. Ceci s'explique par le fait que le café n'est pas réparti uniformément sur la carte et certaines fold se trouvent déséquilibrées.

En comparant ces résultats avec les scores obtenus dans le travail de J.Rod[1] on constate que les F1-Score obtenus dans ce projet sont légèrement plus faibles avec une différence d'environ **1.5%** pour les images Sentinel1C. Ceci peut s'expliquer par le fait que les modèles de J.Rod[1] utilisent des images de tailles 15x15 plus nombreuses et avec moins d'informations superflues.

Les expériences réalisées comme la variation de bande, l'utilisation d'image de taille 32x32, l'utilisation du réseau ResNet50 avec 3 bandes, l'entraînement d'un modèle multilabel et l'utilisation de la focal loss n'ont pas permis d'améliorer les performances par rapport au modèle de transfer learning **DenseNet201 sur 13 bandes** avec les images de taille 64x64.

Une des principales thématiques de ce projet est l'utilisation du transfer learning et la comparaison avec un modèle entraîné « from scratch ». On constate que pour les deux expériences du projet, le modèle de transfer learning a surperformé le modèle entraîné «from scratch ». Un gain d'environ **31%** pour le F1-Score est obtenu pour l'expérience café contre reste et environ **3.6%** de F1-Score pour culture contre non-culture. On peut donc dire que le transfer learning permet dans ce cas-ci d'améliorer les performances des modèles. On constate également pour le café contre le reste que l'entraînement du modèle de transfer learning avec toutes les couches dégelées permet d'améliorer les performances d'environ **6%** sur le Macro F1-Score.

En ce qui concerne l'expérience culture contre non-culture, on constate que de bien meilleurs résultats sont obtenus. Pour la séparation des données aléatoires, un F1-Score de **93.49%** est obtenu et pour la séparation des données géographiquement un F1-Score de **90.96%** est obtenu. La différence entre ces deux scores est bien plus minime que pour l'expérience précédente. Ces résultats s'expliquent par le fait que la variabilité interclasse est plus grande et les classes mieux équilibrées.

Cette dernière expérience semble être une bonne alternative, on pourrait imaginer par exemple utiliser ce réseau neurone en amont pour repérer les différentes cultures et ensuite un humain pourrait classifier manuellement les cultures restantes ou utiliser un autre réseau entraîné à différencier les différents types de cultures entre elles.

Neuchâtel, Juin 04, 2021

Capocasale Romain

R. Capocasale

11. Table des figures

| | |
|--|----|
| Figure 1 : P. Helber & al. results on their two best-performing fine-tuned CNNs[1, p. 6]..... | 10 |
| Figure 2: Architecture réseau de neurones E.Ransom | 12 |
| Figure 3 : Répartition des classes..... | 13 |
| Figure 4 : Séparation géographique du jeu de données | 14 |
| Figure 5 : Séparation jeu de données par classe sur 10 folds – visualisation..... | 15 |
| Figure 6 : Séparation jeu de donnée, méthode des 10 folds, séparation classe par classe | 15 |
| Figure 7: Séparation jeu de donnée, méthode des 10 folds, résultat final | 16 |
| Figure 8 : Récitatif, étapes de préprocessing..... | 17 |
| Figure 9 : Histogramme des pixels normalisés avec les paramètres Eurosat et Vietnam pour la bande B7 | 19 |
| Figure 10 : Architecture modèle Tensorflow Hub..... | 20 |
| Figure 11 : Architecture modèle transfer learning | 20 |
| Figure 12 : Exemple architecture couche “trainable” 13 bandes | 21 |
| Figure 13 : Exemple architecture couche “trainable” RGB 3 bandes | 21 |
| Figure 14 : Architecture modèle transfer learning – toutes couches dégelées..... | 22 |
| Figure 15 : Architecture modèle multi labels..... | 24 |
| Figure 16 : Café contre autre - Accuracy modèles validation croisée – stratifié..... | 27 |
| Figure 17 : Café contre autre - Loss modèles validation croisée – stratifié..... | 27 |
| Figure 18 : Score – Café – DenseNet 64x64..... | 28 |
| Figure 19 : Score – Café – DenseNet 64x64..... | 28 |
| Figure 20 : Culture contre non-culture – Accuracy modèles validation croisée - stratifié | 29 |
| Figure 21 : Culture contre non-culture - Loss modèles validation croisée - stratifié | 29 |
| Figure 22 : Score – Culture – DenseNet 64x64..... | 30 |
| Figure 23 : Café contre autre - Accuracy modèles validation croisée – spatial | 30 |
| Figure 24 : Café contre autre - Loss modèles validation croisée – spatial | 31 |
| Figure 26 : Score – Café –..... | 32 |
| Figure 25 : Score – Café –..... | 32 |
| Figure 27 : Score – Café –..... | 32 |
| Figure 28 : Culture contre non-culture - Accuracy modèles validation croisée – spatial..... | 32 |
| Figure 29 : Culture contre non-culture - Loss modèles validation croisée – spatial..... | 33 |
| Figure 30 : Score – Culture –..... | 33 |
| Figure 31 : Score – Culture –..... | 33 |
| Figure 32 : Score – Culture –..... | 33 |
| Figure 33 : Variation de bandes – DenseNet 3 bandes accuracy..... | 34 |
| Figure 34 : Variation de bandes – DenseNet 3 bandes loss | 34 |
| Figure 35 : Variation de bandes – DenseNet 13 bandes accuracy..... | 34 |
| Figure 36 : Variation de bandes – DenseNet 13 bandes loss | 35 |
| Figure 37 : Multi labels –..... | 35 |
| Figure 38 : Multi labels –..... | 35 |
| Figure 39 : Score – Multi labels culture – | 36 |
| Figure 40 : Score – Multi labels café – | 36 |
| Figure 41 : Comparaison images – Café contre reste | 38 |
| Figure 42: Images Eurosat | 38 |
| Figure 43 : Comparaison images – Culture contre non culture | 39 |

12. Références

- [1] Rod Julien, L'intelligence artificielle au service de l'agriculture durable, 2020
- [2] Ransome Edward, Deep learning for automatic mapping from satellite imagery, 2020
- [3] Helber, Patrick and Bischke, Benjamin and Dengel, Andreas and Borth, Damian, 2019 Eurosat: A novel dataset and deep learning benchmark for land use and land cover classification IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing
<https://arxiv.org/abs/1709.00029>
- [4] Géron Aurélien, Machine Learning avec Scikit-Learn (1ère édition). Paris, France : Dunod, 2017
- [5] Géron Aurélien, Deep Learning avec TensorFlow (1ère édition). Paris, France : Dunod, 2017
- [6] Abdishakur 2018, Land use/Land cover classification with Deep Learning,
<https://link.medium.com/ncZaHz2YLeb>
- [7] fchollet, 2020 Transfer learning & fine-tuning, https://keras.io/guides/transfer_learning/
- [8] Saptashwa Bhattacharyya, A Loss Function Suitable for Class Imbalanced Data: “Focal Loss”,
<https://towardsdatascience.com/a-loss-function-suitable-for-class-imbalanced-data-focal-loss-af1702d75d75>
- [9] Jason Brownlee, Transfer Learning in Keras with Computer Vision Models,
<https://machinelearningmastery.com/how-to-use-transfer-learning-when-developing-convolutional-neural-network-models/>

13. Annexes

Liste des annexes :

1. BandsVariabilityWaltherSimon
2. AccLossPlots

bandsVariability

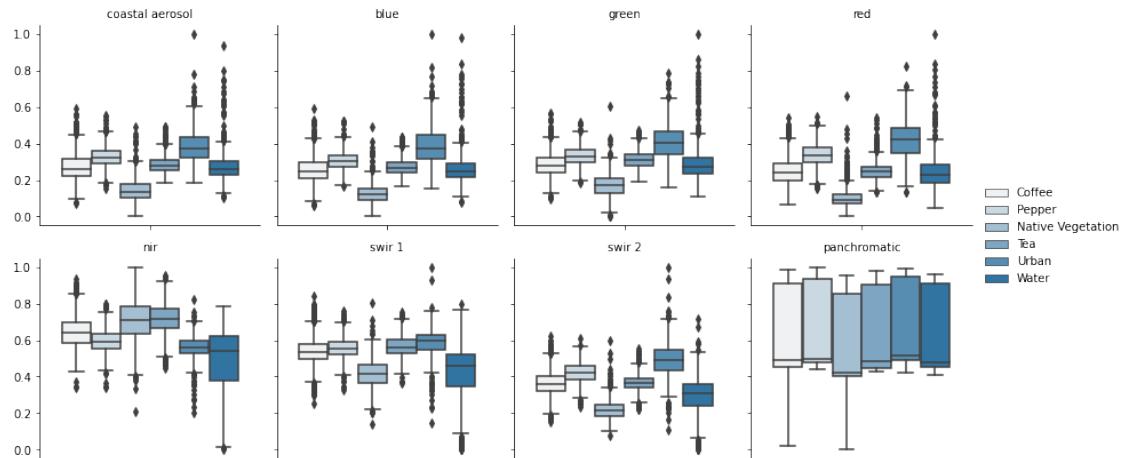
March 15, 2021

1 Coffee comparison to other entities

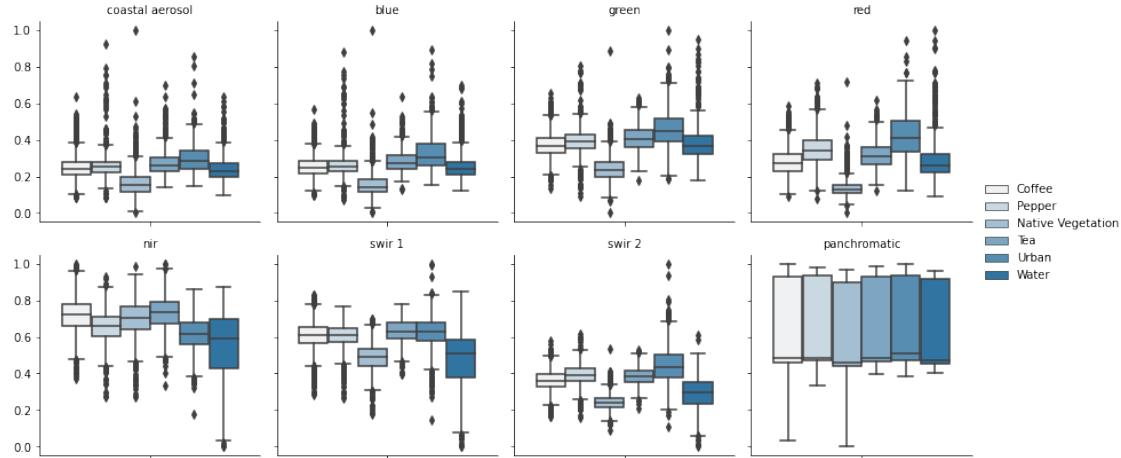
Il semble plus facile de distinguer le café des autres labels pendant la saison sèche. En effet, en regardant les différentes boîtes à moustache, on remarque qu'il est plus facile de différencier les labels avec les canaux *coastal aerosol*, *blue*, *green*, *red* et *nir* en saison sèche qu'en saison humide.

On remarque aussi que le café prend des valeurs proches du poivre, du thé et de l'eau. Toutefois, le café s'écarte un peu de poivre avec le canal *red* et du thé et de l'eau avec le canal *nir* même s'il reste encore un assez grand chevauchement entre leurs données.

1.1 Dry Season (2016 - 2018)



1.2 Wet Season (2016 - 2018)



2 Coffee comparison by season

Au Vietnam, il y a deux saisons : la saison sèche et la saison humide. La saison sèche commence en novembre et fini en avril. La saison humide, quant à elle, commence mai et fini en octobre [1].

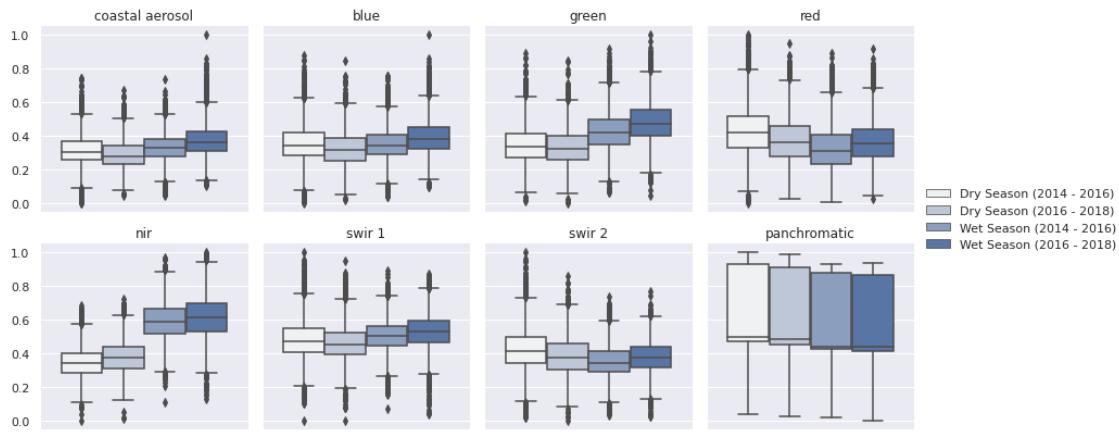
Si l'on étudie la variabilité des données du café à travers ces deux saisons, on observe que la variabilité est tantôt meilleur en saison sèche et tantôt meilleure en saison humide.

Sur les années 2014 à 2016, la saison humide à moins de variabilité à part pour le canal *nir* alors que pendant les années 2016 à 2018, la variabilité est plus faible en saison sèche pour tous les canaux sauf *red*, *swir1*, *swir2*, et *panchromatic*.

Pour ce qui est des valeurs médianes, on remarque une net augmentation du canal *nir*, de ~0.35 à ~0.6, en saison humide. Le canal *green* augmente lui aussi, mais en 2016-2018 plus fortement, passant de ~0.33 à ~0.42 pour les années 2014 à 2016 et ~0.47 en 2016 à 2018.

Tous les canaux ont tendances à légèrement augmenter à part *red*, *panchromatic* et *swir2* qui baissent légèrement en saison humide.

[1] <https://www.britannica.com/place/Vietnam/Climate>, Climate of Vietnam, Britannica



Dry Season (2014 - 2016)

| | coastal aerosol | blue | green | red | nir | \ |
|-------|-----------------|-------------|-------------|-------------|-------------|---|
| count | 4109.000000 | 4109.000000 | 4109.000000 | 4109.000000 | 4109.000000 | |
| mean | 0.315933 | 0.356673 | 0.350217 | 0.435539 | 0.341912 | |
| std | 0.088299 | 0.108875 | 0.115619 | 0.146328 | 0.089545 | |
| min | 0.000000 | 0.000000 | 0.015665 | 0.000000 | 0.000000 | |
| 25% | 0.257335 | 0.282380 | 0.270376 | 0.332943 | 0.280460 | |
| 50% | 0.305068 | 0.345214 | 0.336434 | 0.417811 | 0.339281 | |
| 75% | 0.367959 | 0.419007 | 0.416699 | 0.518538 | 0.398197 | |
| max | 0.740810 | 0.874754 | 0.892815 | 1.000000 | 0.685079 | |

| | swir 1 | swir 2 | panchromatic |
|-------|-------------|-------------|--------------|
| count | 4109.000000 | 4109.000000 | 4109.000000 |
| mean | 0.484461 | 0.426828 | 0.677409 |
| std | 0.115691 | 0.123650 | 0.229472 |
| min | 0.001010 | 0.019111 | 0.041952 |
| 25% | 0.407098 | 0.343363 | 0.471789 |
| 50% | 0.469673 | 0.413509 | 0.495829 |
| 75% | 0.544720 | 0.498556 | 0.928610 |
| max | 1.000000 | 1.000000 | 1.000000 |

Dry Season (2016 - 2018)

| | coastal aerosol | blue | green | red | nir | \ |
|-------|-----------------|-------------|-------------|-------------|-------------|---|
| count | 4109.000000 | 4109.000000 | 4109.000000 | 4109.000000 | 4109.000000 | |
| mean | 0.288847 | 0.325419 | 0.333349 | 0.376259 | 0.374908 | |
| std | 0.080218 | 0.097992 | 0.105681 | 0.132098 | 0.094972 | |
| min | 0.049725 | 0.020611 | 0.000000 | 0.027756 | 0.014178 | |
| 25% | 0.230753 | 0.252954 | 0.257982 | 0.277295 | 0.309963 | |
| 50% | 0.278692 | 0.315050 | 0.322332 | 0.364772 | 0.373983 | |

| | | | | | |
|-----|----------|----------|----------|----------|----------|
| 75% | 0.341769 | 0.389978 | 0.399580 | 0.460714 | 0.436920 |
| max | 0.674045 | 0.847252 | 0.845386 | 0.948544 | 0.722836 |

| | swir 1 | swir 2 | panchromatic |
|-------|-------------|-------------|--------------|
| count | 4109.000000 | 4109.000000 | 4109.000000 |
| mean | 0.462713 | 0.386058 | 0.659705 |
| std | 0.104501 | 0.112223 | 0.229017 |
| min | 0.000000 | 0.000000 | 0.022941 |
| 25% | 0.391493 | 0.305633 | 0.451776 |
| 50% | 0.454183 | 0.376616 | 0.486349 |
| 75% | 0.524536 | 0.459071 | 0.906854 |
| max | 0.948932 | 0.855568 | 0.982871 |

Wet Season (2014 - 2016)

| | coastal | aerosol | blue | green | red | nir | \ |
|-------|-------------|-------------|-------------|-------------|-------------|-----|---|
| count | 4109.000000 | 4109.000000 | 4109.000000 | 4109.000000 | 4109.000000 | | |
| mean | 0.332593 | 0.349723 | 0.427760 | 0.329392 | 0.590160 | | |
| std | 0.079613 | 0.090444 | 0.110569 | 0.129649 | 0.112163 | | |
| min | 0.049228 | 0.040402 | 0.065175 | 0.007257 | 0.106750 | | |
| 25% | 0.279720 | 0.289545 | 0.350838 | 0.235438 | 0.516336 | | |
| 50% | 0.328660 | 0.344509 | 0.419769 | 0.312444 | 0.589291 | | |
| 75% | 0.380440 | 0.404790 | 0.497603 | 0.405442 | 0.665999 | | |
| max | 0.732990 | 0.755197 | 0.913133 | 0.892162 | 0.964828 | | |

| | swir 1 | swir 2 | panchromatic |
|-------|-------------|-------------|--------------|
| count | 4109.000000 | 4109.000000 | 4109.000000 |
| mean | 0.505678 | 0.353650 | 0.626539 |
| std | 0.093960 | 0.092534 | 0.226390 |
| min | 0.073028 | 0.034229 | 0.018568 |
| 25% | 0.442582 | 0.290128 | 0.423963 |
| 50% | 0.501194 | 0.344258 | 0.441329 |
| 75% | 0.561539 | 0.411576 | 0.875130 |
| max | 0.889102 | 0.733909 | 0.925576 |

Wet Season (2016 - 2018)

| | coastal | aerosol | blue | green | red | nir | \ |
|-------|-------------|-------------|-------------|-------------|-------------|-----|---|
| count | 4109.000000 | 4109.000000 | 4109.000000 | 4109.000000 | 4109.000000 | | |
| mean | 0.374850 | 0.394145 | 0.482655 | 0.366102 | 0.610847 | | |
| std | 0.094700 | 0.100936 | 0.118304 | 0.124267 | 0.126298 | | |
| min | 0.107039 | 0.097391 | 0.044705 | 0.027140 | 0.131636 | | |
| 25% | 0.310921 | 0.325820 | 0.401366 | 0.279256 | 0.526551 | | |
| 50% | 0.365076 | 0.383706 | 0.474335 | 0.355034 | 0.612705 | | |
| 75% | 0.426780 | 0.452402 | 0.554965 | 0.441414 | 0.695031 | | |
| max | 1.000000 | 1.000000 | 1.000000 | 0.914939 | 1.000000 | | |

| | | |
|--------|--------|--------------|
| swir 1 | swir 2 | panchromatic |
|--------|--------|--------------|

| | | | |
|-------|-------------|-------------|-------------|
| count | 4109.000000 | 4109.000000 | 4109.000000 |
| mean | 0.528439 | 0.376791 | 0.613532 |
| std | 0.100725 | 0.092426 | 0.225916 |
| min | 0.036048 | 0.024233 | 0.000000 |
| 25% | 0.464190 | 0.313966 | 0.411387 |
| 50% | 0.530125 | 0.371236 | 0.436530 |
| 75% | 0.592034 | 0.436533 | 0.860215 |
| max | 0.866677 | 0.767457 | 0.931040 |

2.1 Coffee comparison each two months (2014-2020)

Ici, on compare les valeurs des canaux du café deux mois par deux mois en prenant une médiane de toutes les données récoltées entre 2014 à 2020.

On remarque alors quelles sont les périodes qui ont le moins de variabilité et l'évolution des canaux à travers les mois.

2.1.1 Contexte

Au Vietnam, la récolte du café commence en octobre [1] et prend fin de décembre à janvier [2], [3].

Après la récolte, les arbres à café sont taillés pour laisser plus lumière passer et c'est de janvier à avril où le cafetier est fortement arrosé [2].

En février se déroule le Têt, la fête du nouvel an lunaire, et il est possible que la café soit volontairement abondamment arrosé pour fleurir pendant Têt.

[1] <http://www.fao.org/3/ap301e/ap301e.pdf>, “Diversification by smallholder farmers: Vietnam Robusta Coffee”, AGSF, p.9

[2] ibid., p.19

[3] <http://amarin.com.vn/coffee-harvest-season>, “Harvest Season”, AmaRin

2.1.2 Évolution

C'est à partir de mai que le canal *nir* augmente significativement, la médiane augmentant de 0.299 à 0.549.

En mars à avril, le canal rouge est au plus haut. En effet la médiane passe à 0.406 alors qu'elle était à 0.318 et qu'elle retombe ensuite à 0.328.

On observe aussi que *panchromatic* augmente en mars à avril.

Le canal *green*, quant à lui, est à son plus haut en mai à juin et juillet à août.

2.1.3 Variabilité

Voici les mois qui ont le moins de variabilité pour certains canaux : - Novembre à décembre : *coastal aerosol*, *blue*, *green* et *red* - Septembre à octobre : *_swirl1*, *swirl2* - Juillet à août : *panchromatic* - Mars à avril : *nir*

Septembre à octobre et novembre à décembre sont dans les deux périodes avec la plus petite variabilité à part pour *nir* et *panchromatic*. Attention toutefois à ne pas oublier que la récolte du café commence en octobre et fini vers décembre.

La période de mai à juin n'est jamais la plage de temps avec la plus petite variabilité, mais se situe toujours en troisième ou quatrième position.

2.1.4 Proportion de nuage moyenne

Cette proportion a été calculée à partir du script de téléchargement avec :

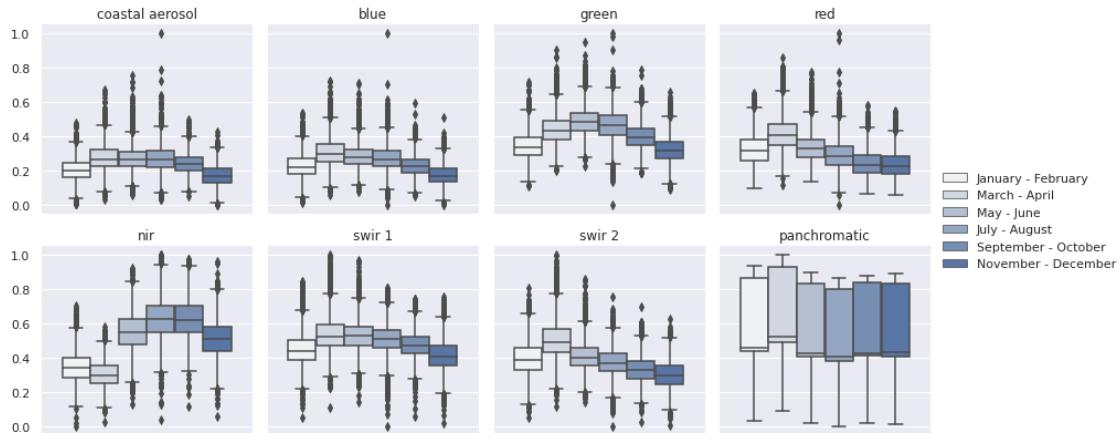
```
merged_image_collections.aggregate_stats('CLOUD_COVER').getInfo()
```

- Janvier à février : 28.71%
- Mars à avril : 23.30%
- Mai à juin : 37.40%
- Juillet à août : 45.11%
- Septembre à octobre : 39.66%
- Novembre à décembre : 38.90%

2.1.5 Réflexions

La plage de temps janvier-février est intéressante. Cette période a une faible couverture de nuage et se trouve en saison sèche ce qui devrait rendre le café plus facile à distinguer des autres labels. Malheureusement la variabilité est plutôt mauvaise sur la bande *red* et sur les bandes *swir1*, *swir2*, et *panchromatic*.

La période mai-juin pourrait aussi être intéressante, car sa variabilité est toujours plutôt bonne. Il y a néanmoins une plus grande quantité de nuage et elle se trouve en saison humide, ce qui pourrait peut-être rendre le café plus difficile à différencier. C'est dans cette période que la bande rouge est la plus prononcée.



coastal aerosol :

November - December : 0.0590617986994859
September - October : 0.05918915932984787
January - February : 0.061803873043750575
May - June : 0.07087521870499923
March - April : 0.07509162183899608
July - August : 0.07929351528365433

blue :

November - December : 0.05919015493495189
September - October : 0.05959641954030545
May - June : 0.0680424633207376
January - February : 0.06818768734453522
July - August : 0.07166327210972691
March - April : 0.07967923604446996

green :

November - December : 0.07515852210815206
September - October : 0.07575419896737919
January - February : 0.08027665296447384
May - June : 0.08180422764959856
March - April : 0.08631662384219274
July - August : 0.08772324831585153

red :

November - December : 0.07414631066769256
September - October : 0.07662710339562966
May - June : 0.08394594172612105
July - August : 0.08613764959082691
January - February : 0.09112963362367679
March - April : 0.09834233290193085

nir :

March - April : 0.074590665377683
January - February : 0.08985293542437296
November - December : 0.10861702071798451
May - June : 0.10975860225204752
September - October : 0.11295853424633111
July - August : 0.11762086840936019

swir 1 :

September - October : 0.07529911537698303
November - December : 0.0846938400696235
July - August : 0.08599370200096167

May - June : 0.08660030729961972
January - February : 0.09456424150170754
March - April : 0.10736032327229689

swir 2 :

September - October : 0.07393734450263158
November - December : 0.07938521757002268
July - August : 0.07958250545197741
May - June : 0.08390100012402185
January - February : 0.09906883534036365
March - April : 0.11269856287795753

panchromatic :

July - August : 0.21227356725713414
November - December : 0.2132577222273897
September - October : 0.2133299495723808
May - June : 0.21333667844463447
January - February : 0.21537276658230534
March - April : 0.21852505598330657

January - February

coastal aerosol blue green red nir \\\n

| | coastal | aerosol | blue | green | red | nir | \ |
|-------|-------------|-------------|-------------|-------------|-------------|-------------|---|
| count | 4109.000000 | 4109.000000 | 4109.000000 | 4109.000000 | 4109.000000 | 4109.000000 | |
| mean | 0.205271 | 0.226209 | 0.347082 | 0.327257 | 0.346079 | | |
| std | 0.061804 | 0.068188 | 0.080277 | 0.091130 | 0.089853 | | |
| min | 0.006410 | 0.014345 | 0.112500 | 0.095339 | 0.000000 | | |
| 25% | 0.163121 | 0.178943 | 0.290916 | 0.260596 | 0.283754 | | |
| 50% | 0.201644 | 0.221088 | 0.339634 | 0.318198 | 0.342547 | | |
| 75% | 0.246096 | 0.269115 | 0.395963 | 0.383964 | 0.400115 | | |
| max | 0.480913 | 0.535994 | 0.718308 | 0.649903 | 0.705151 | | |

swir 1 swir 2 panchromatic
count 4109.000000 4109.000000 4109.000000
mean 0.450472 0.396303 0.628867
std 0.094564 0.099069 0.215373
min 0.051965 0.053415 0.032945
25% 0.388023 0.326743 0.435475
50% 0.440641 0.386657 0.459993
75% 0.503114 0.460060 0.863965
max 0.845436 0.806638 0.933645

March - April

coastal aerosol blue green red nir \

| | | | | | |
|-------|-------------|-------------|-------------|-------------|-------------|
| count | 4109.000000 | 4109.000000 | 4109.000000 | 4109.000000 | 4109.000000 |
| mean | 0.275748 | 0.308479 | 0.439516 | 0.417639 | 0.304331 |
| std | 0.075092 | 0.079679 | 0.086317 | 0.098342 | 0.074591 |
| min | 0.034232 | 0.056690 | 0.199851 | 0.120458 | 0.023586 |
| 25% | 0.225272 | 0.253526 | 0.381701 | 0.348648 | 0.253140 |
| 50% | 0.268716 | 0.300086 | 0.431477 | 0.406208 | 0.299588 |
| 75% | 0.322626 | 0.355310 | 0.488332 | 0.474665 | 0.352681 |
| max | 0.669019 | 0.722803 | 0.901450 | 0.860890 | 0.578939 |

| | swir 1 | swir 2 | panchromatic |
|-------|-------------|-------------|--------------|
| count | 4109.000000 | 4109.000000 | 4109.000000 |
| mean | 0.538686 | 0.507133 | 0.688357 |
| std | 0.107360 | 0.112699 | 0.218525 |
| min | 0.107265 | 0.114216 | 0.092947 |
| 25% | 0.467721 | 0.428601 | 0.491740 |
| 50% | 0.520603 | 0.489807 | 0.520623 |
| 75% | 0.590016 | 0.566930 | 0.925755 |
| max | 1.000000 | 1.000000 | 1.000000 |

May - June

| | coastal | aerosol | blue | green | red | nir | \ |
|-------|-------------|-------------|-------------|-------------|-------------|-----|---|
| count | 4109.000000 | 4109.000000 | 4109.000000 | 4109.000000 | 4109.000000 | | |
| mean | 0.272504 | 0.287473 | 0.487732 | 0.338179 | 0.547978 | | |
| std | 0.070875 | 0.068042 | 0.081804 | 0.083946 | 0.109759 | | |
| min | 0.059742 | 0.080052 | 0.225073 | 0.134096 | 0.131439 | | |
| 25% | 0.228412 | 0.241942 | 0.431232 | 0.280282 | 0.475416 | | |
| 50% | 0.264083 | 0.279570 | 0.481633 | 0.328216 | 0.549400 | | |
| 75% | 0.307853 | 0.324223 | 0.534465 | 0.384685 | 0.622563 | | |
| max | 0.754840 | 0.709351 | 0.947510 | 0.777629 | 0.920093 | | |

| | swir 1 | swir 2 | panchromatic |
|-------|-------------|-------------|--------------|
| count | 4109.000000 | 4109.000000 | 4109.000000 |
| mean | 0.530546 | 0.410598 | 0.596937 |
| std | 0.086600 | 0.083901 | 0.213337 |
| min | 0.143832 | 0.141440 | 0.020587 |
| 25% | 0.471720 | 0.353928 | 0.407686 |
| 50% | 0.525188 | 0.402311 | 0.427720 |
| 75% | 0.581381 | 0.457123 | 0.832175 |
| max | 0.968151 | 0.854324 | 0.893512 |

July - August

| | coastal | aerosol | blue | green | red | nir | \ |
|-------|-------------|-------------|-------------|-------------|-------------|-----|---|
| count | 4098.000000 | 4098.000000 | 4098.000000 | 4098.000000 | 4098.000000 | | |
| mean | 0.271580 | 0.272188 | 0.465770 | 0.293403 | 0.626269 | | |
| std | 0.079294 | 0.071663 | 0.087723 | 0.086138 | 0.117621 | | |
| min | 0.030887 | 0.000000 | 0.000000 | 0.000000 | 0.042000 | | |

| | | | | | |
|-----|----------|----------|----------|----------|----------|
| 25% | 0.219126 | 0.225248 | 0.408283 | 0.232152 | 0.547986 |
| 50% | 0.263981 | 0.268722 | 0.462404 | 0.284007 | 0.626754 |
| 75% | 0.316597 | 0.314992 | 0.520380 | 0.345220 | 0.705020 |
| max | 1.000000 | 1.000000 | 1.000000 | 1.000000 | 1.000000 |

| | swir 1 | swir 2 | panchromatic |
|-------|-------------|-------------|--------------|
| count | 4098.000000 | 4098.000000 | 4098.000000 |
| mean | 0.506293 | 0.372849 | 0.571381 |
| std | 0.085994 | 0.079583 | 0.212274 |
| min | 0.000000 | 0.000000 | 0.000000 |
| 25% | 0.454371 | 0.320728 | 0.380770 |
| 50% | 0.505912 | 0.368101 | 0.406912 |
| 75% | 0.560643 | 0.422669 | 0.801878 |
| max | 0.804997 | 0.753548 | 0.861705 |

September - October

| | coastal | aerosol | blue | green | red | nir | \ |
|-------|-------------|-------------|-------------|-------------|-------------|-----|---|
| count | 4109.000000 | 4109.000000 | 4109.000000 | 4109.000000 | 4109.000000 | | |
| mean | 0.240116 | 0.228114 | 0.399952 | 0.244911 | 0.624911 | | |
| std | 0.059189 | 0.059596 | 0.075754 | 0.076627 | 0.112959 | | |
| min | 0.054111 | 0.056135 | 0.185716 | 0.067013 | 0.115255 | | |
| 25% | 0.198317 | 0.186444 | 0.346986 | 0.186697 | 0.547340 | | |
| 50% | 0.239213 | 0.226528 | 0.393025 | 0.234946 | 0.621657 | | |
| 75% | 0.278708 | 0.266347 | 0.447903 | 0.293136 | 0.701767 | | |
| max | 0.494379 | 0.591969 | 0.786637 | 0.580670 | 0.975233 | | |

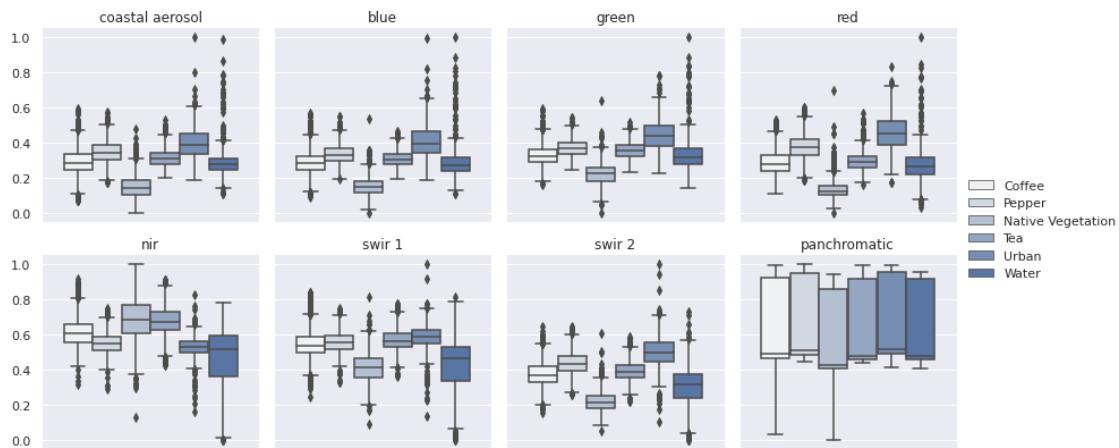
| | swir 1 | swir 2 | panchromatic |
|-------|-------------|-------------|--------------|
| count | 4109.000000 | 4109.000000 | 4109.000000 |
| mean | 0.474039 | 0.331126 | 0.600948 |
| std | 0.075299 | 0.073937 | 0.213330 |
| min | 0.060027 | 0.026972 | 0.021198 |
| 25% | 0.422851 | 0.278455 | 0.410614 |
| 50% | 0.471907 | 0.325535 | 0.424768 |
| 75% | 0.521494 | 0.377959 | 0.836031 |
| max | 0.740442 | 0.695715 | 0.872972 |

November - December

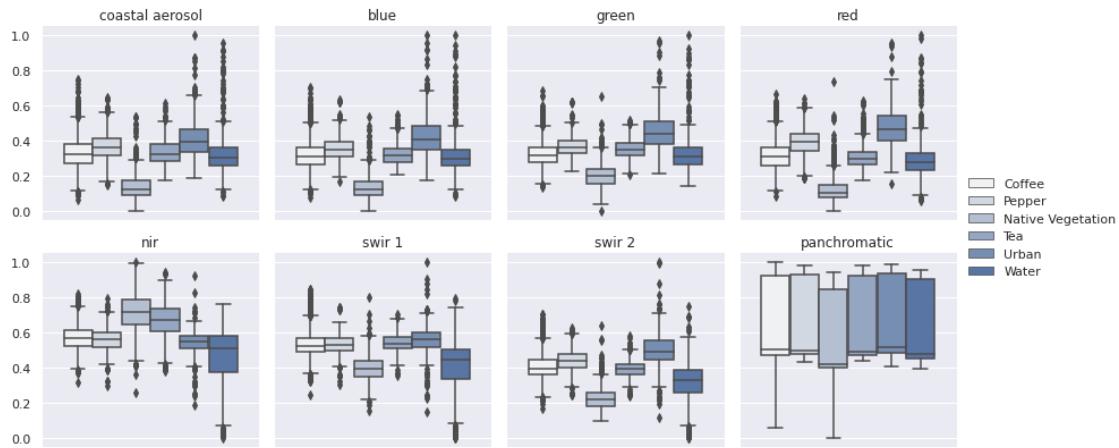
| | coastal | aerosol | blue | green | red | nir | \ |
|-------|-------------|-------------|-------------|-------------|-------------|-----|---|
| count | 4109.000000 | 4109.000000 | 4109.000000 | 4109.000000 | 4109.000000 | | |
| mean | 0.174870 | 0.175798 | 0.322915 | 0.235903 | 0.508411 | | |
| std | 0.059062 | 0.059190 | 0.075159 | 0.074146 | 0.108617 | | |
| min | 0.000000 | 0.003788 | 0.091173 | 0.060365 | 0.056011 | | |
| 25% | 0.132392 | 0.135990 | 0.271258 | 0.183532 | 0.437334 | | |
| 50% | 0.167286 | 0.170474 | 0.317711 | 0.226445 | 0.507622 | | |
| 75% | 0.213563 | 0.213347 | 0.368838 | 0.282654 | 0.581698 | | |
| max | 0.425051 | 0.513210 | 0.660567 | 0.550003 | 0.961989 | | |

| | swir 1 | swir 2 | panchromatic |
|-------|-------------|-------------|--------------|
| count | 4109.000000 | 4109.000000 | 4109.000000 |
| mean | 0.415326 | 0.301931 | 0.595940 |
| std | 0.084694 | 0.079385 | 0.213258 |
| min | 0.020563 | 0.005229 | 0.011425 |
| 25% | 0.357402 | 0.244759 | 0.403864 |
| 50% | 0.408920 | 0.294628 | 0.429236 |
| 75% | 0.468861 | 0.353162 | 0.828469 |
| max | 0.751630 | 0.621975 | 0.887803 |

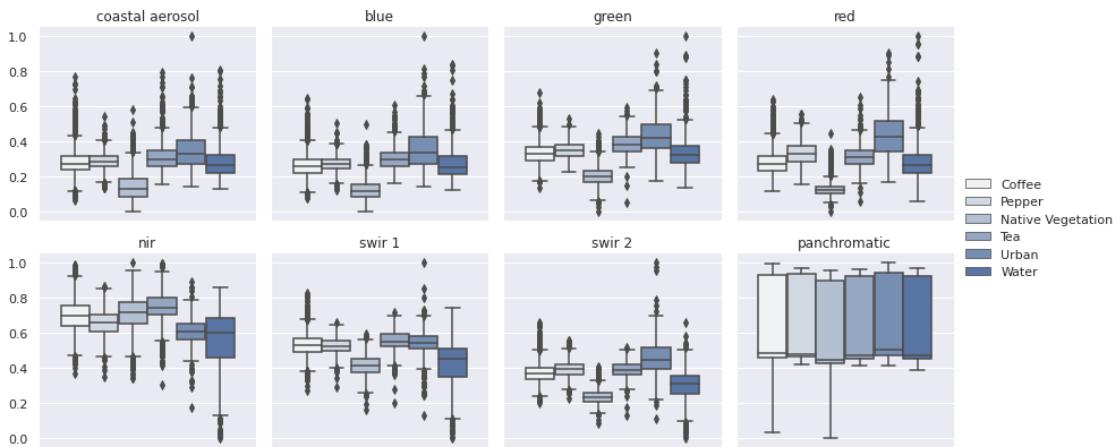
January - February



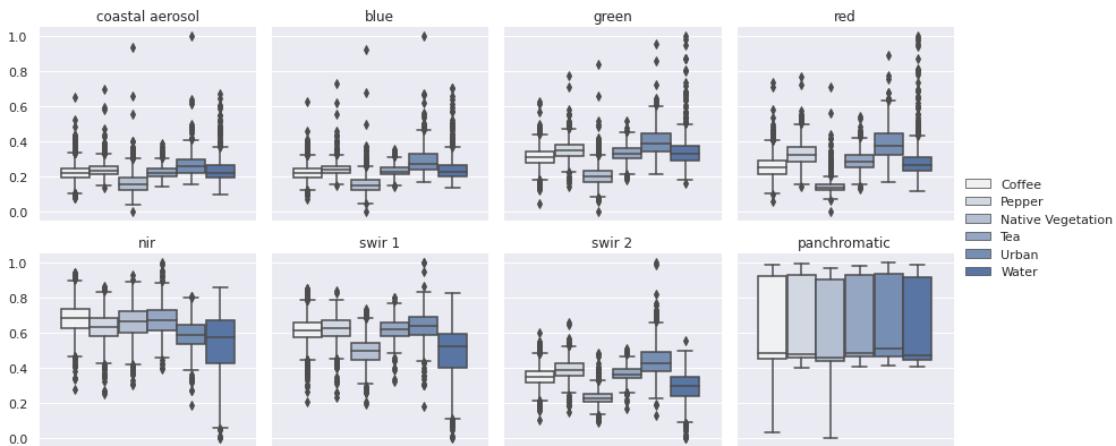
March - April



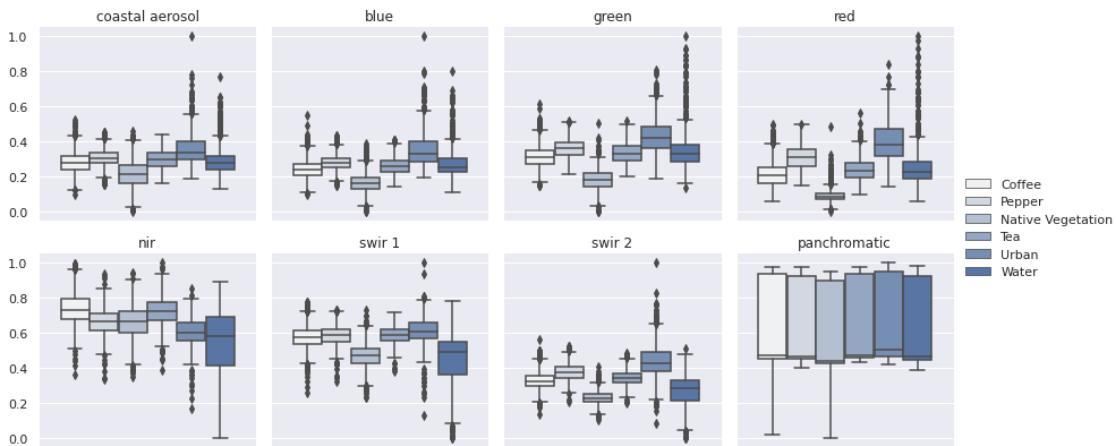
May - June



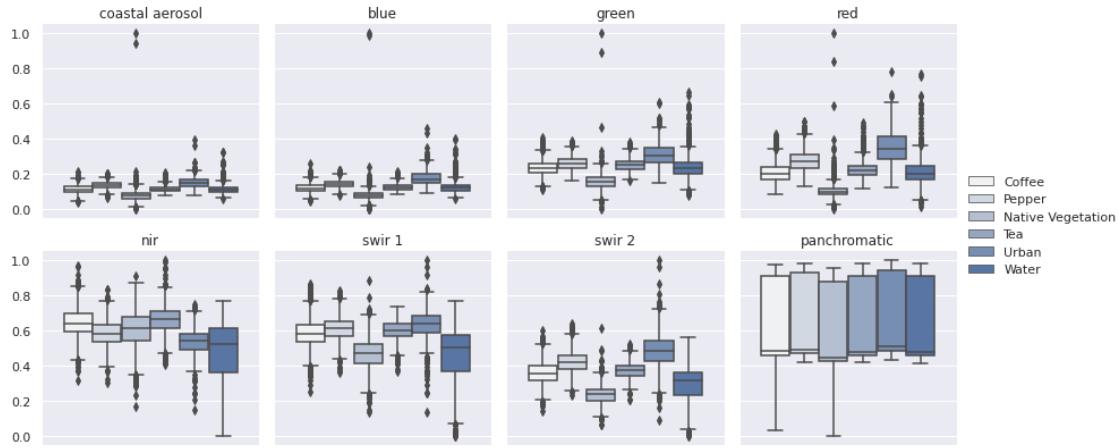
July - August



September - October

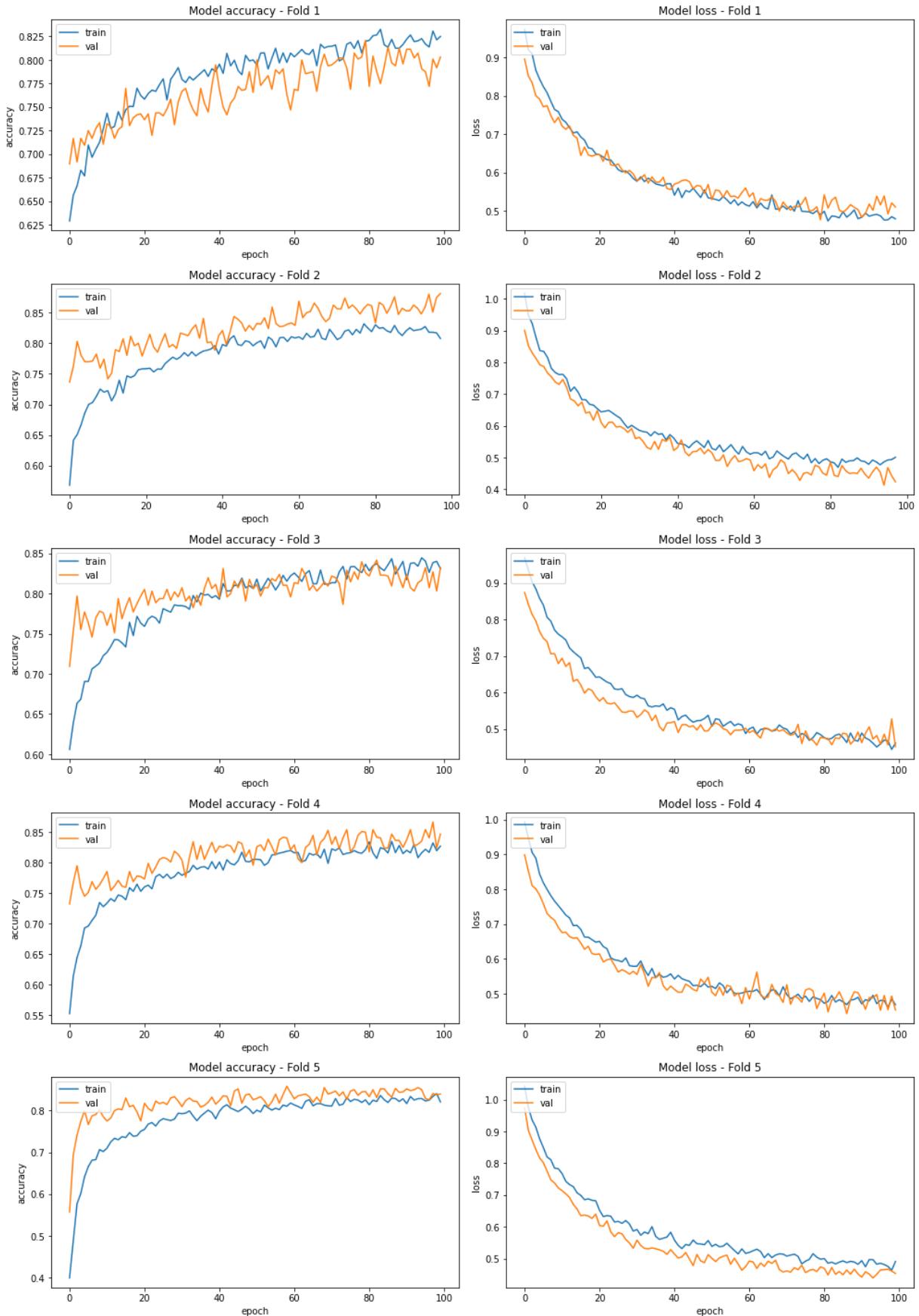


November – December

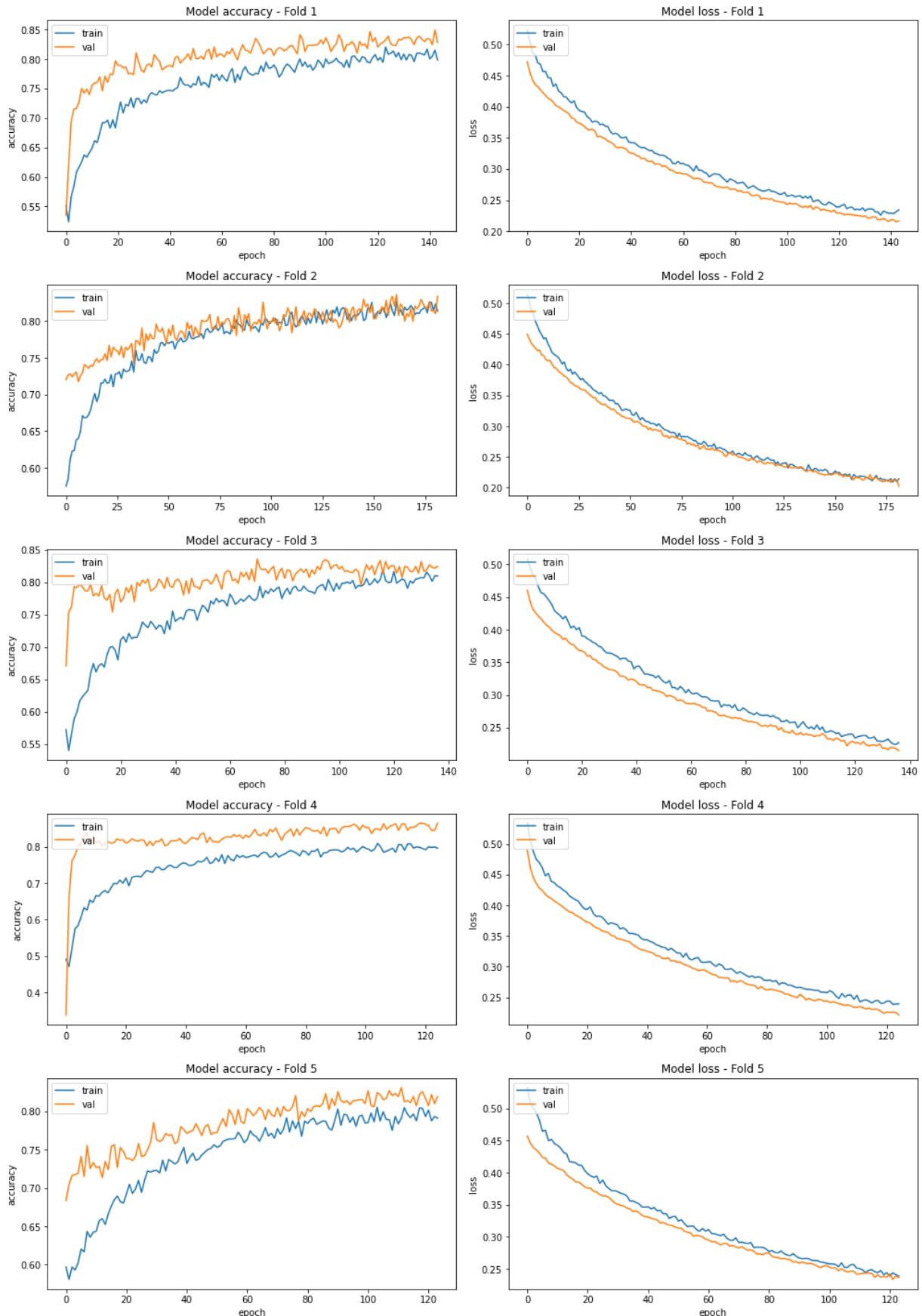


In [1]:

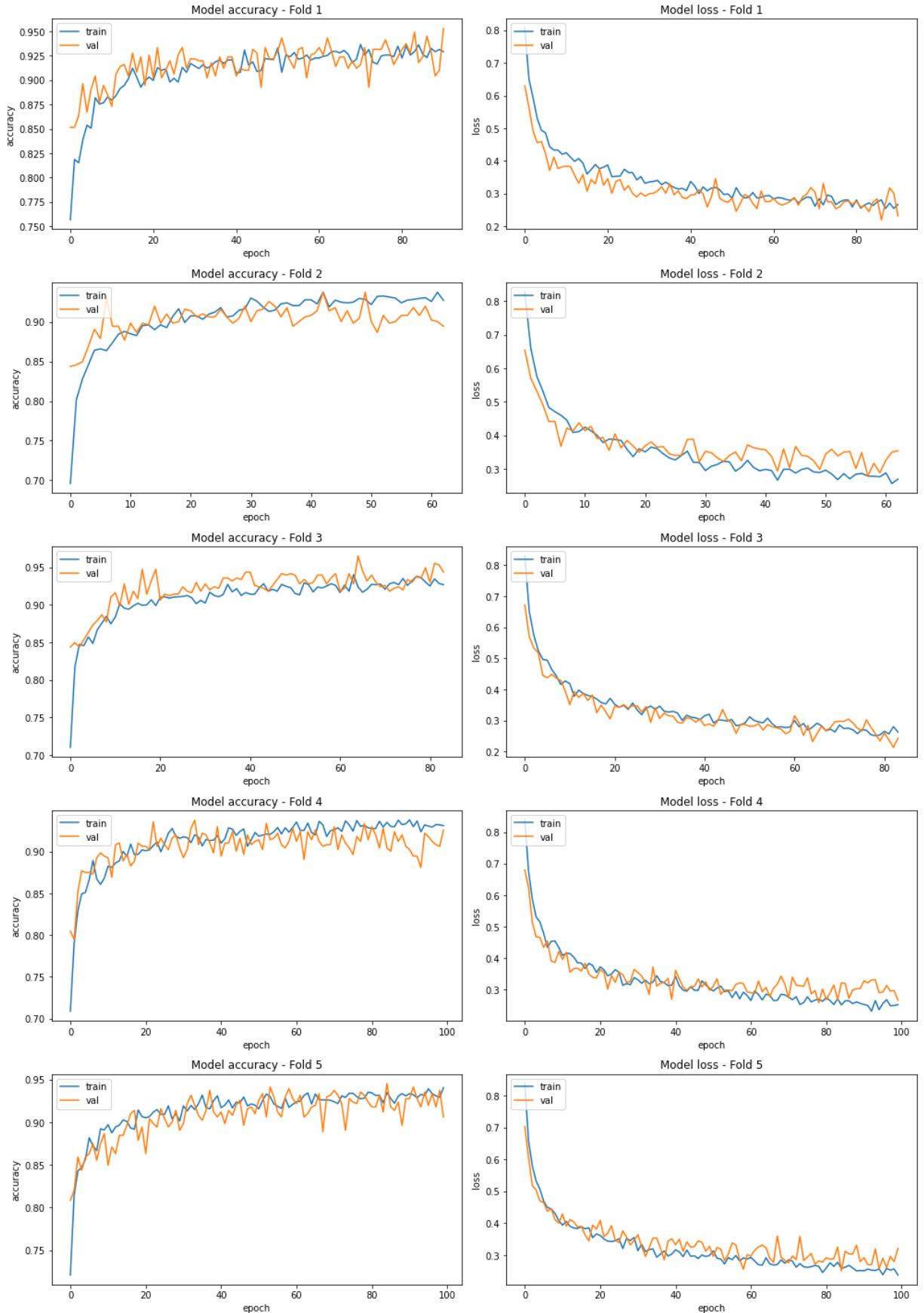
Dense 64x64 - Coffee - Stratified



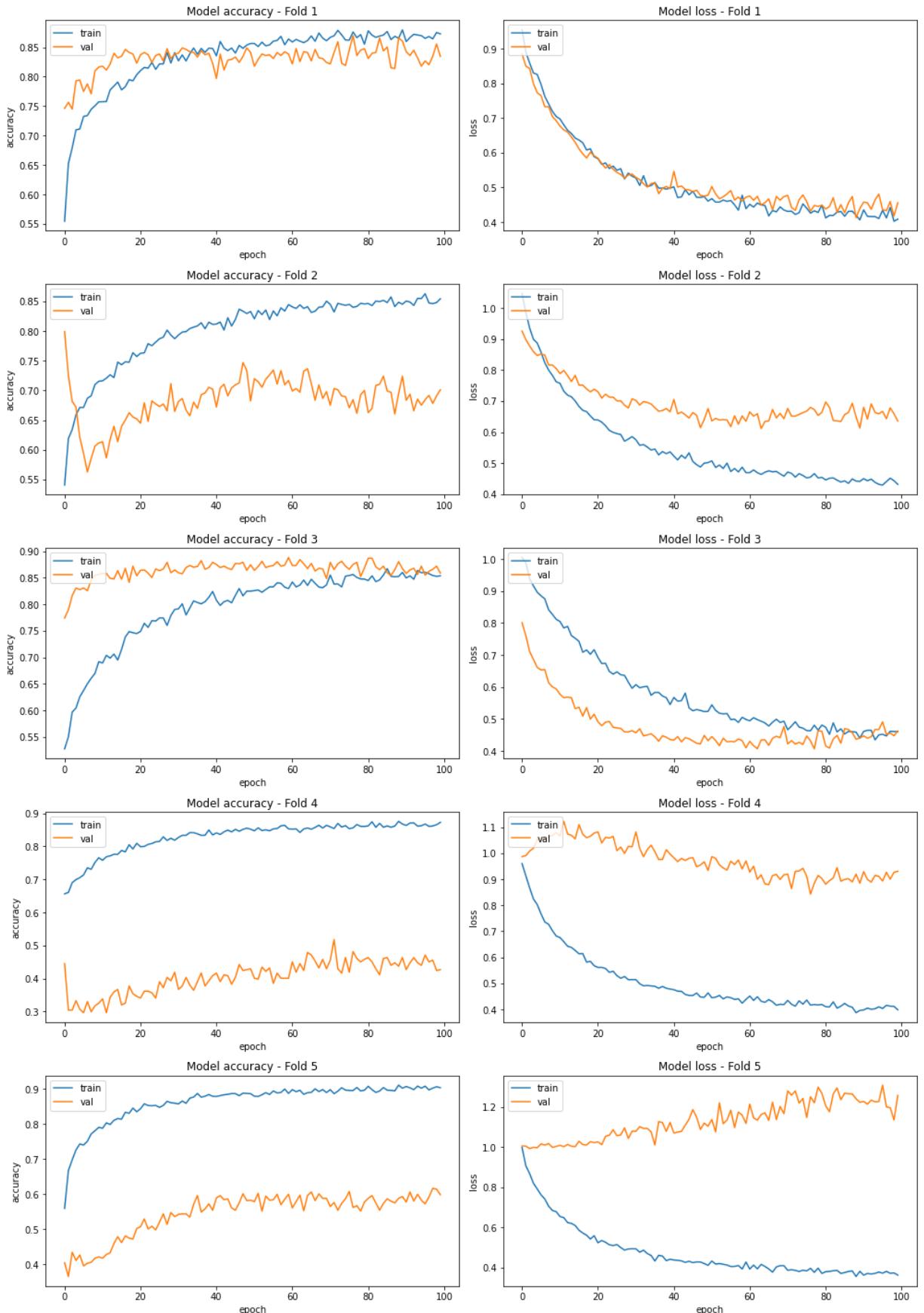
Dense 64x64 Focal Loss 300 Epoch - Coffee - Stratified



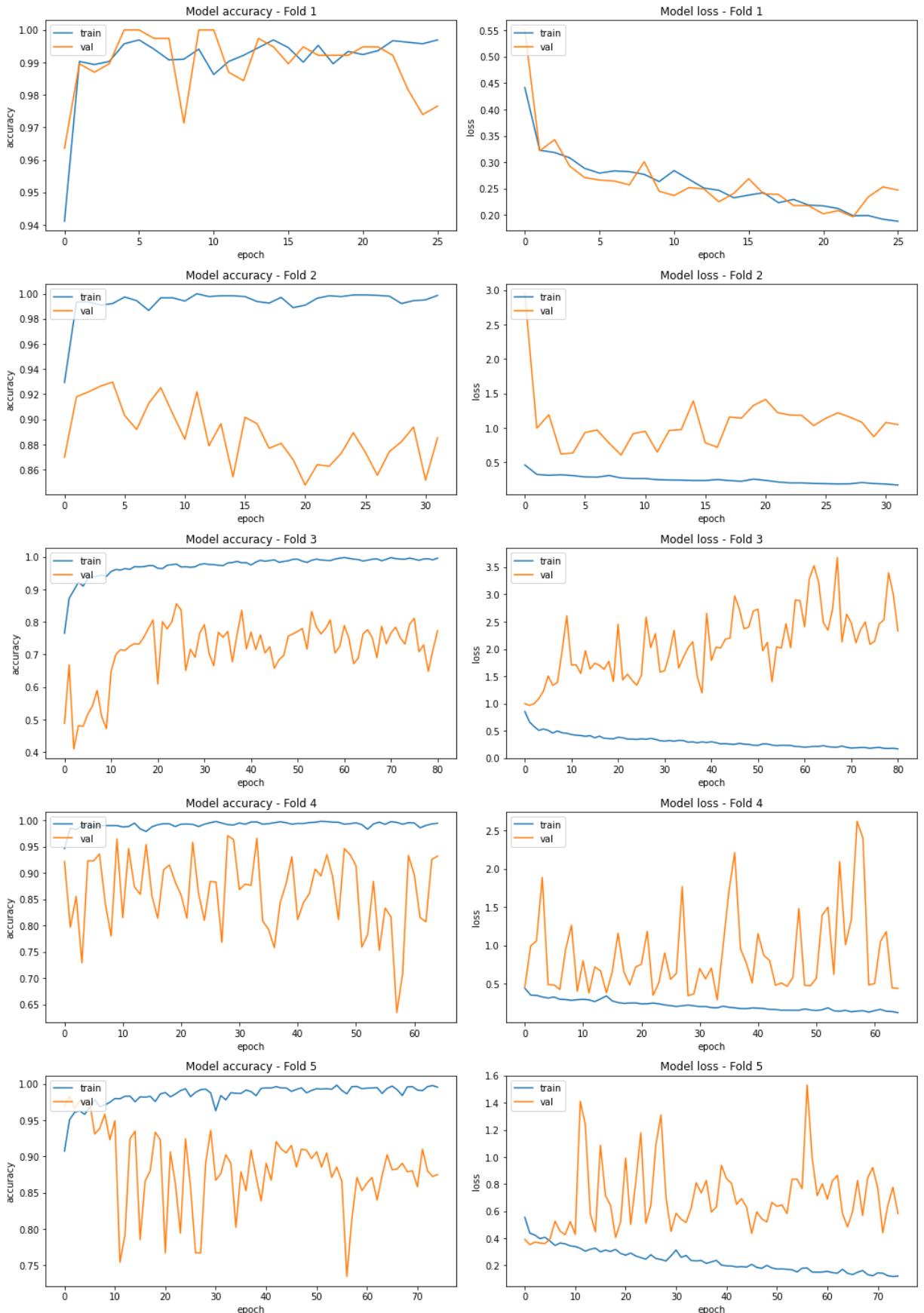
Dense 64x64 - Culture - Stratified



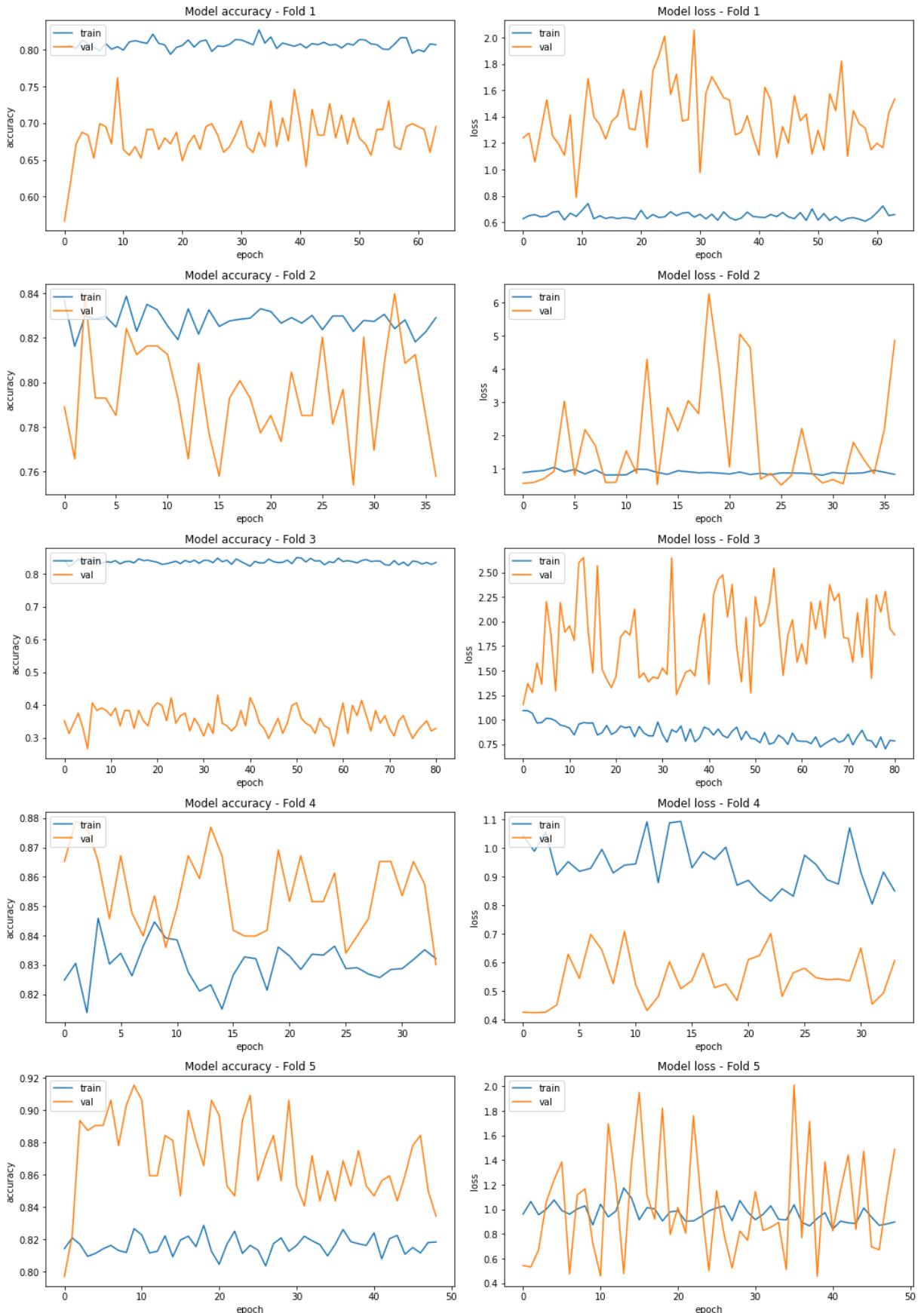
DenseNet 64x64 - Coffee - Spatial



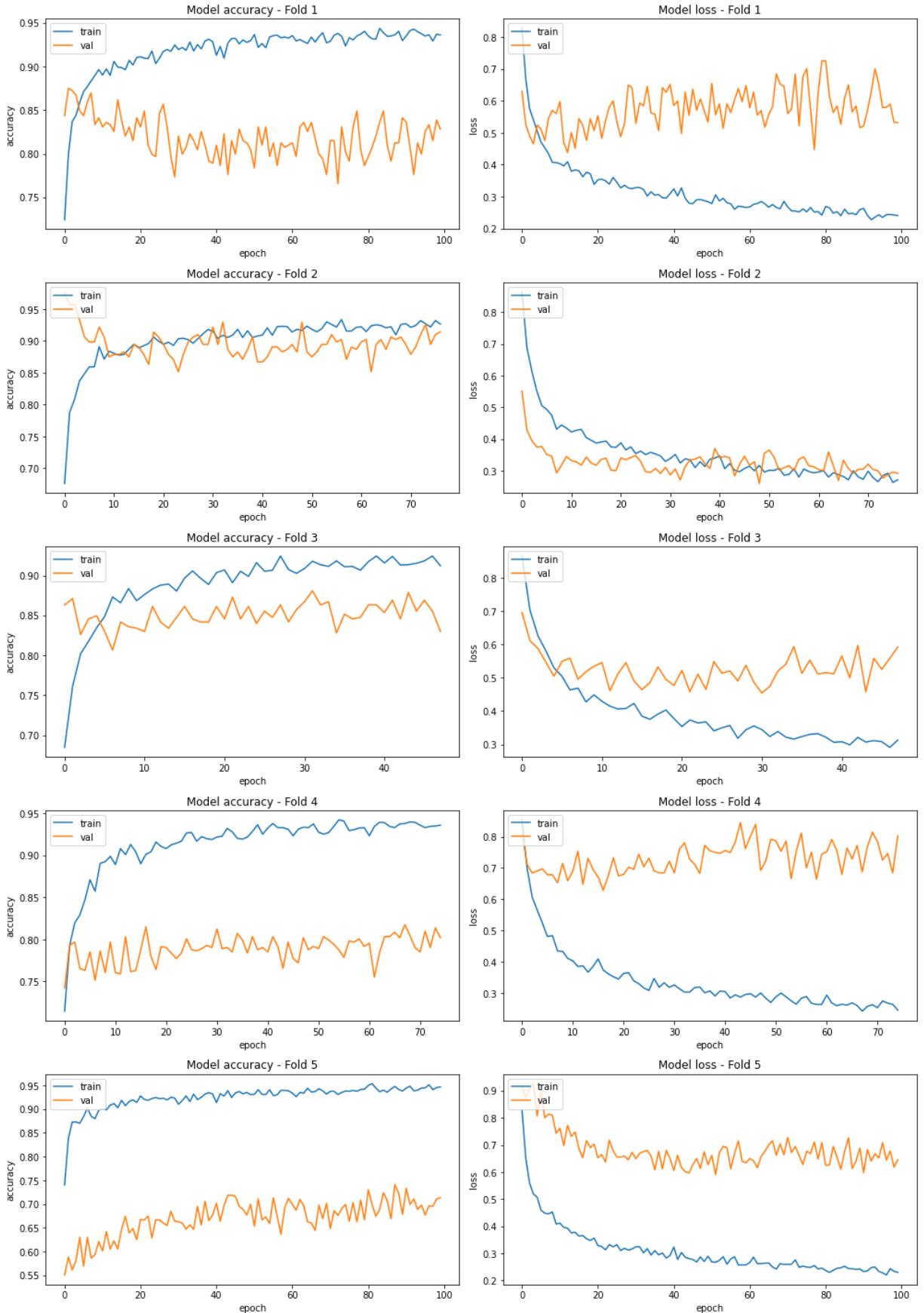
DenseNet 64x64 From Scratch - Coffee - Spatial



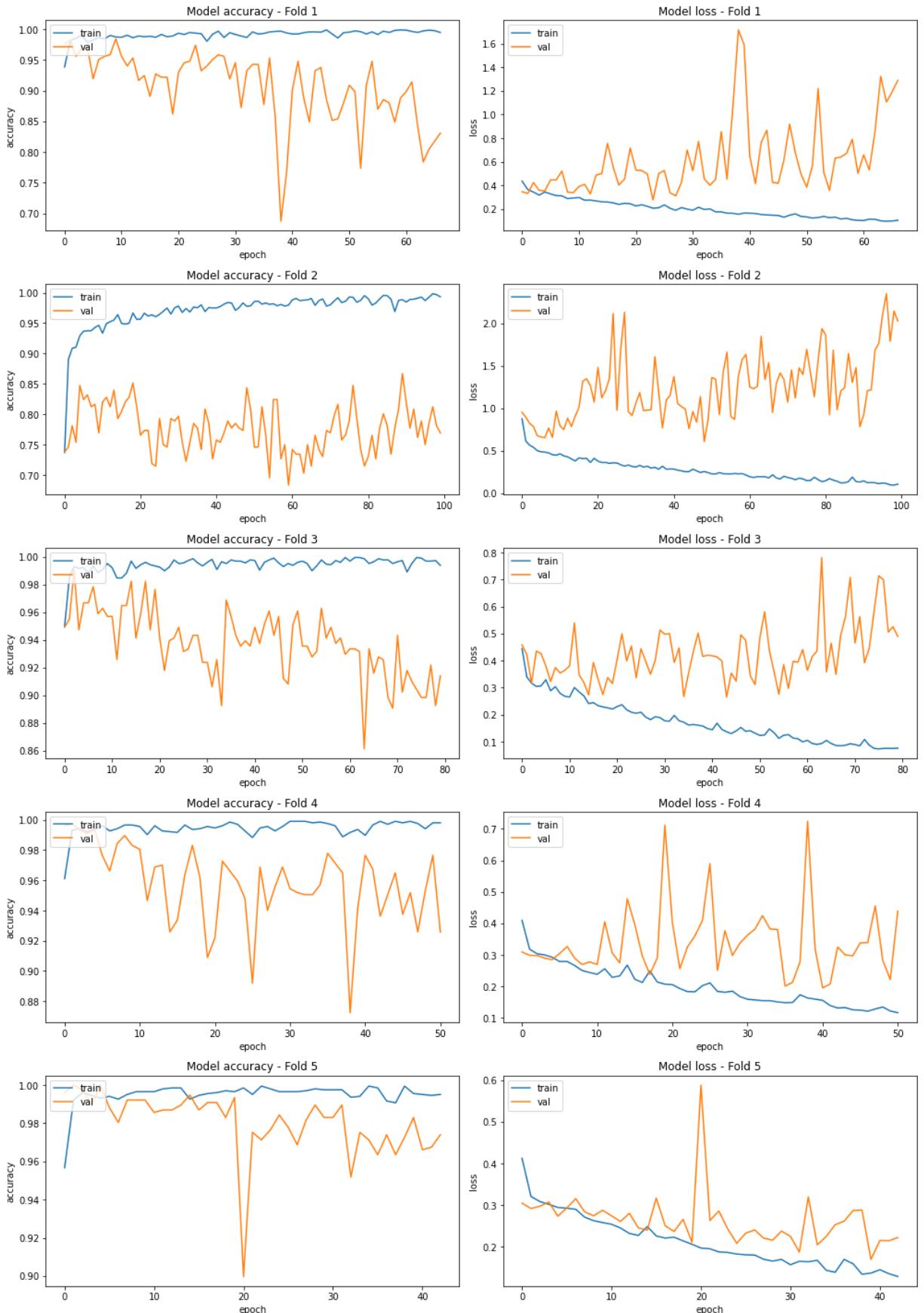
DenseNet 64x64 Unfreeze all layers - Coffee - Spatial



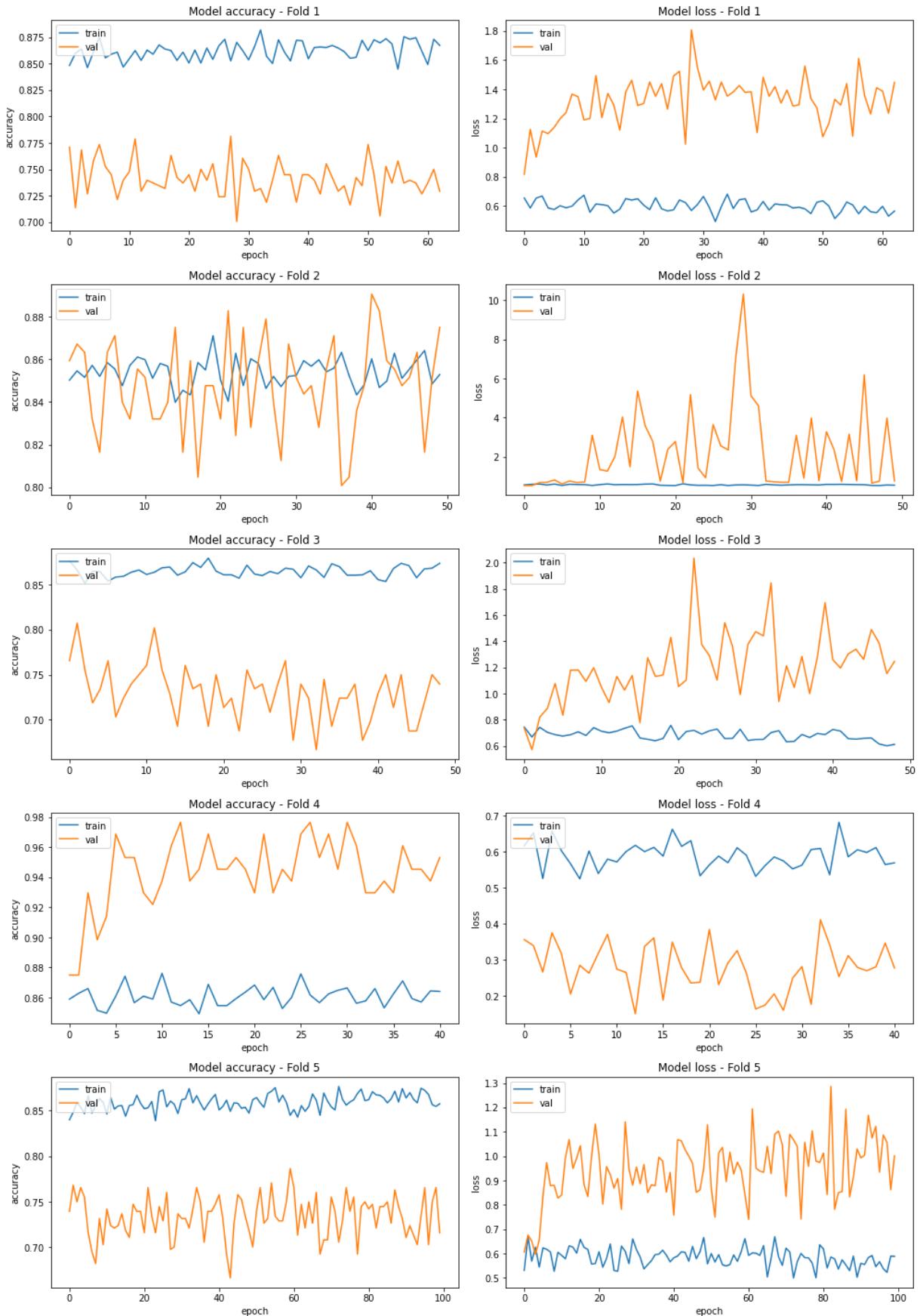
DenseNet 64x64 - Culture - Spatial



DenseNet 64x64 From Scratch - Culture - Spatial



DenseNet 64x64 Unfreeze all layers - Culture - Spatial



DenseNet 64x64 - Multi label - Spatial

