

## HANDS ON AI – Certificat IA de l'UMONS

### Rapport Défi 1 : IA et analyse d'images : détection et localisation sur images

- **Etudiant 1** : Nicolas Melaerts 253882
- **Etudiant 2** : Manu Mathey 254427

#### I. Description succincte du problème : choix d'options (classification et XAI + options facultatives : localisation de feu ou Vision Transformers ou pas d'option 😊)

L'objectif de ce projet est de classer des images afin de déterminer si elles contiennent un feu, un départ de feu, ou l'absence de feu. Pour cela, nous avons utilisé des réseaux de neurones profonds.

Les étapes principales du projet incluent :

1. Préparation des données : organiser et prétraiter les données nécessaires à l'entraînement.
2. Développement d'un modèle : concevoir un réseau de neurones avec des paramètres adaptés pour effectuer la classification des images.
3. Analyse et interprétation des résultats : utiliser des techniques d'explicabilité de l'intelligence artificielle (XAI) pour expliquer les performances du modèle.

En complément, nous avons également réalisé les points facultatifs proposés pour ce projet :

- La localisation des feux et de la fumée à l'aide du modèle YOLO.
- L'utilisation de Vision Transformers (ViT) pour la classification des images forestières.

Enfin, nous avons testé nos modèles en temps réel sur des vidéos téléchargées depuis YouTube. Ces tests nous ont permis d'évaluer les performances de nos approches en conditions réelles. Deux vidéos démontrant les résultats de la classification et de la localisation des feux ont été créées et mises en ligne sur YouTube. Les liens sont disponibles dans la section "Test des modèles sur une vidéo".

#### II. Description et évaluation des données (Small, BD1, BD2, BD3 + données personnelles si d'application)

Donner une analyse de chaque base de données avec évaluation de son influence sur le résultat de classification

Small : Cette base de données contient environ 300 images de fire et no\_fire, mais seulement 150 images de start\_fire. Ce déséquilibre crée un biais car la catégorie start\_fire est sous-représentée. De plus, le faible nombre total d'images limite la précision des résultats.

DB1 : Cette base est fortement déséquilibrée, avec neuf fois plus d'images dans la catégorie fire et cinq fois plus dans la catégorie no\_fire par rapport à start\_fire. Ce déséquilibre entraîne un modèle biaisé, qui aura tendance à prédire davantage les catégories fire ou no\_fire, car les données de start\_fire sont insuffisamment représentées pendant l'entraînement.

DB2 : Cette base de données est très équilibrée, avec 501 images par catégorie. Cependant, elle présente un problème majeur : de nombreux doublons dans chaque catégorie. Ces doublons introduisent un biais important, car le modèle s'entraîne plusieurs fois sur les mêmes images, ce qui affecte sa capacité à généraliser.

DB3 : C'est la meilleure des quatre bases de données pour l'entraînement. Les données sont équilibrées (501 images par catégorie) et contiennent peu ou pas de doublons. Grâce à ces caractéristiques, cette base permet d'obtenir les meilleurs résultats parmi les quatre bases de données testées.

Base de données personnelle :

Combinaison des quatre bases de données fournies :

Nous avons créé une grande base de données regroupant toutes les images issues de Small, DB1, DB2, et DB3. Pour garantir l'unicité des données, nous avons appliqué un algorithme afin de supprimer les doublons et conserver uniquement des images uniques. Ensuite, grâce à un processus de data augmentation et data réduction, nous avons équilibré les trois catégories pour obtenir un même nombre d'images dans chaque classe. Ce rééquilibrage était particulièrement nécessaire, car la catégorie start\_fire contenait beaucoup moins d'images comparée aux catégories fire et no\_fire.

Création de notre propre base de données :

Nous avons également conçu une base de données personnalisée.

Pour télécharger les images nécessaires à notre base, nous avons utilisé une extension Google Chrome nommée "Image Downloader – pictures and photos saver". Cette extension nous a permis de rechercher des images pertinentes (par exemple, en recherchant "feu de forêt") et de trier les résultats selon des critères spécifiques, comme le format et la taille. Grâce à cet outil, nous avons pu rapidement sélectionner et télécharger des images pertinentes à ajouter à notre base de données.

### III. Gestion des paramètres du modèle

**Décrire l'influence des hyperparamètres sur les résultats du modèle**

**a. Batch Size :**

**Le batch size détermine le nombre d'exemples utilisés pour calculer le gradient avant la mise à jour des poids.**

**Petit batch size : Génère des gradients plus bruités, ce qui peut améliorer la généralisation, mais ralentit la convergence.**

**Grand batch size : Stabilise l'entraînement, mais augmente les besoins en mémoire GPU (ce qui est une contrainte importante sous Google Colab). De plus, un batch size trop élevé peut entraîner un surajustement.**

**Dans notre cas, nous avons limité le batch size à un maximum de 64 pour éviter des problèmes de mémoire.**

**b. Learning Rate :**

**Le learning rate contrôle la vitesse d'ajustement des poids du modèle pendant l'entraînement.**

**Taux élevé : Accélère la convergence, mais peut manquer le minimum global ou provoquer une divergence.**

**Taux faible : Ralentit l'apprentissage, mais permet d'atteindre des solutions plus précises.**

**Pour optimiser ce paramètre, nous avons fait des test avec un scheduler tel que ReduceLROnPlateau, qui ajuste dynamiquement le learning rate en fonction de la performance sur les données de validation.**

**c. Nombre d'époques :**

**Le nombre d'époques correspond au nombre de passages complets sur l'ensemble des données d'entraînement.**

**Trop peu d'époques : Peut entraîner un sous-apprentissage, le modèle n'étant pas suffisamment optimisé.**

**Trop d'époques : Peut entraîner un surajustement, le modèle devenant trop spécifique aux données d'entraînement.**

**d. Optimiseur :**

L'optimiseur détermine la manière dont les gradients ajustent les poids.

**SGD (Stochastic Gradient Descent) :** Favorise une convergence plus lente mais offre souvent une meilleure généralisation.

**Adam :** Accélère l'apprentissage grâce à un ajustement adaptatif des gradients, particulièrement utile pour des modèles complexes ou des données bruitées.

**e. Régularisation (Dropout) :**

La régularisation, telle que le Dropout, désactive aléatoirement une fraction des neurones pendant l'entraînement pour limiter le surapprentissage (overfitting).

Nous avons utilisé un Dropout de 0,5 dans les couches pleinement connectées. Cela a amélioré la généralisation sur les données de validation, sans ralentir de manière significative l'entraînement.

**IV. Modèles proposés (modèle personnel et/ou existant avec Transfert Learning)**

Décrire le fonctionnement de votre meilleur modèle (vous pouvez inclure une photo d'architecture de votre modèle)

Notre meilleur modèle est le ResNet152, qui repose sur le principe de résidualité. L'architecture de ResNet utilise des blocs résiduels pour contrer le problème de dégradation de la performance. Le principe clé de ResNet est l'ajout de "skip connections" (connexions de contournement) entre les couches, permettant ainsi de transmettre directement des informations importantes d'une couche à une autre. Cela facilite l'apprentissage, surtout dans les réseaux très profonds, et aide à prévenir la perte d'informations critiques durant l'entraînement.

Avec ses 152 couches, ResNet152 est particulièrement puissant et performant pour notre tâche, car il peut extraire et identifier des caractéristiques très détaillées des images, ce qui est essentiel pour notre modèle de détection de feu.

Article des ingénieurs de Microsoft ayant présenté Resnet : <https://arxiv.org/pdf/1512.03385>

Exemple d'architecture de Resnet50 qui a une architecture moins large et détaillée que Resnet152 :

<http://ethereon.github.io/netscope/#/gist/db945b393d40bfa26006>

layer name	output size	18-layer	34-layer	50-layer	101-layer	152-layer
conv1	112×112	7×7, 64, stride 2				
conv2_x	56×56	3×3 max pool, stride 2				
		$\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$
conv3_x	28×28	$\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 8$
conv4_x	14×14	$\begin{bmatrix} 3 \times 3, 256 \\ 3 \times 3, 256 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 256 \\ 3 \times 3, 256 \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 23$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 36$
conv5_x	7×7	$\begin{bmatrix} 3 \times 3, 512 \\ 3 \times 3, 512 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 512 \\ 3 \times 3, 512 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$
	1×1	average pool, 1000-d fc, softmax				
FLOPs		$1.8 \times 10^9$	$3.6 \times 10^9$	$3.8 \times 10^9$	$7.6 \times 10^9$	$11.3 \times 10^9$

Pour les architectures Resnet les blocs résiduels sont indiqués entre crochet. La réduction de la résolution des données se fait dans les couches conv3\_1, conv4\_1 et conv5\_1, avec un pas de 2.

## V. Comparaison entre vos 5 meilleurs modèles (03 modèles au minimum, Modèle 5 : choix libre si d'application)

Caractéristiques	VGG16	ResNet152	GoogleNet	EfficientNet	BONUS : ViT
<b>Entraînement</b>					
Taille du modèle en MB	537.091	232.6	47.932	557.099	327.30
Training_Loss	0.14	0.10098	0.16598	0.15058	0.057738
Training_Accuracy	0.95512	0.9674	0.94131	0.95666	-
Validation_Loss	0.19791	0.20488	0.1999	0.13567	0.08045
Validation_Accuracy	0.93558	0.92791	0.93098	0.95859	0.97666
Temps (optionnel)	34 min	45 min	26 min	45 min	33 min
<b>Test 1 en Local (<a href="https://nextcloud.ig.umons.ac.be/s/RsZpqtYfDgYzKxr">https://nextcloud.ig.umons.ac.be/s/RsZpqtYfDgYzKxr</a>)</b>					
Test_Loss	0.2677	0.1320	0.3000	0.1455	-
Test_Accuracy	0.9133	0.9600	0.9000	0.9467	-
<b>Test 2 en ligne (<a href="https://nextcloud.ig.umons.ac.be/s/rYnCgdSmmRerJRa/download/TestData.zip">https://nextcloud.ig.umons.ac.be/s/rYnCgdSmmRerJRa/download/TestData.zip</a>)</b>					
Classement challenge en ligne	1 <sup>er</sup> (dimanche 24/11 à 20h)	-	-	-	-
Score en ligne	0.967	0.979	0.949	0.955	0.839

## VI. Test des modèles sur vidéo (lien vers la vidéo annotée)

Analyser/interpréter vos résultats (précision et performance) d'application du modèle sur une vidéo de votre choix

Les résultats montrent que le modèle est globalement performant et fiable. Lorsqu'il détecte un feu, il le fait avec une grande précision, tout comme lorsqu'il identifie de la fumée. Toutefois, on observe une légère faiblesse lorsqu'il y a un feu localisé dans une zone précise de l'image, accompagné d'une grande quantité de fumée. Dans ce cas, le modèle peut parfois classifier l'image comme un start\_fire au lieu de fire.

De plus, dans le cockpit de l'hélicoptère (une zone spécifique qui peut être difficile à interpréter pour le modèle), il arrive que le modèle alterne entre start\_fire et no\_fire lorsqu'il détecte de la fumée au loin.

En ce qui concerne le bonus avec YOLO, la détection des zones de feu et de fumée est particulièrement précise. Le modèle parvient à détecter facilement plusieurs zones de feu ou de fumée dans une même image.

Lien vers vidéos annotées :

Détection de fire, no\_fire, start\_fire : [https://youtu.be/\\_kJmNDs02vY](https://youtu.be/_kJmNDs02vY)

Bonus, localisation de fire et smoke : <https://youtu.be/vq-laiUJwV0>

Combinaison de détection de feu et départ de feu et localisation de feu et fumée : <https://youtu.be/bBL225O-4Jc>

## VII. Analyse des résultats et conclusion

Nous avons constaté d'excellents résultats avec le modèle ResNet lors de l'analyse de notre modèle sur une vidéo. Cependant, ce défi nous a montré que, peu importe le modèle utilisé parmi les quatre testés, il est possible d'obtenir de bonnes performances, à condition de bien choisir nos hyperparamètres et de disposer d'une base de données de qualité.

De plus nous avons également obtenu de très bons résultats avec les ViT, une optimisation plus conséquente pourrait encore nettement augmenter la fiabilité et donc les performances de ce type de modèle.

En ce qui concerne le choix des hyperparamètres, nous avons été limités par la contrainte d'une heure d'exécution sur Google Colab pour les expérimentations avec Optuna. Néanmoins, cela nous a permis d'obtenir des indications. Si nous avons concentré nos efforts sur un autre modèle que ResNet et que nous l'avons optimisé au maximum, il aurait pu devenir notre meilleur modèle.

De plus, nous étions limités par la taille de la mémoire GPU, ce qui ne nous permettait pas d'utiliser des tailles de batch supérieures à 64 pour certains modèles. Par conséquent, il serait intéressant de lancer des expérimentations avec des tailles de batch plus grandes, sans limite de temps pour l'exécution, et potentiellement un nombre d'époques plus élevé que ceux testés. Une analyse rigoureuse des images de notre base de données serait également nécessaire pour vérifier qu'il n'y a pas de biais significatifs dans nos résultats.

Ces ajustements pourraient considérablement améliorer la fiabilité et les performances de nos modèles.