Rapport Projet 3A
Analyse de traces de sang

Cléa Han, Yanis Labeyrie et Adrien Zabban

Mars 2024

1 Introduction

Notre projet 3A s'intéresse à l'analyse de traces de sang dans le cadre du travail de l'expert criminalistique Philippe Esperança. En effet, l'objectif est de réaliser une intelligence artificielle pour assister et faciliter le travail d'analyse de scènes de crimes présentant du sang.

Les travaux de Philippe Esperança sur l'analyse des traces de sangs [1], a abouti à la classification de ces traces en 19 classes distinctes qui sont listées dans la Table 1. Notre but est alors de faire un modèle de machine learning capable de prédire la classe pour une image de trace de sang.

Classe des types de trace de sang	
1- Modèles Traces passives	2- Modèles Goutte à Goutte
3- Modèle Transfert par contact	4- Modèle Transfert glissé
5- Modèle Altération par contact	6- Modèle Altération glissée
7- Modèle d'Accumulation	8- Modèle Coulée
9- Modèle Chute de volume	10- Modèle sang Propulsé
11- Modèle d'éjection	12- Modèle Volume Impacté
13- Modèle Imprégnation	14- Modèle Zone d'interruption
15- Modèle d'impact	16- Modèle Foyer de modèle d'impact
17- Modèle Trace gravitationnelle	18- Modèle Sang expiré
19- Modèle Trace d'insecte	

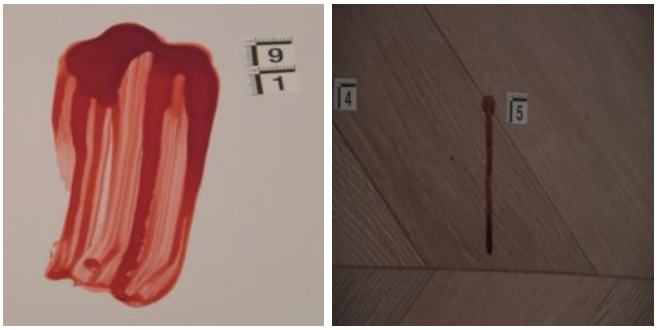
Table 1 – Liste des 19 modèles de trace de sang

2 Données

Philippe Esperança nous a fourni deux bases de données. La première contient des images de traces de sang reproduites en laboratoire. La deuxième correspond à des images issues de scène de crime. Cependant, il y avait la présence d'une classe trop minoritaire parmi ces 19 classes, qui est la classe de trace d'insectes possédant uniquement quatre images. Cette classe a donc été retirée afin de garder une certaine distribution relativement équilibrée. Nous avons donc travaillé avec 18 classes.

2.1 Données de laboratoire

Dans un premier temps, nous avons pu manipuler des données de laboratoire, c'est-à-dire des images de traces de sang reproduites en laboratoire sur des fonds réguliers, hors des scènes de crimes. Ces fonds sont de quatre types différents : bois, linoleum (lino), carrelage et papier. Ces données sont composées de 10978 images. La Figure 1 présente des images de taches de sang reproduites en laboratoire. La Table 8 en annexe montre un exemple de trace de sang pour chacune des 18 classes.



(a) Modèle Transfert glissé sur un fond de papier

(b) Modèle de Coulée sur fond de lino

FIGURE 1 – Deux images de laboratoire

Nous avons réparti ces données de laboratoire en 80% dans nos données d'entraînement, 10% dans nos données de validation et 10% dans nos données de test. La Figure 2 nous montre la distribution des données de laboratoire sur chacune des classes et chacun des datasets.

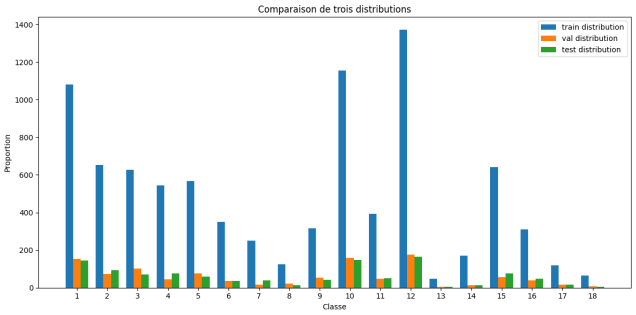


FIGURE 2 – Distribution des données de laboratoire sur les 18 classes selon le dataset d'entraînement, de validation et de test.

2.2 Données réelles issues de scènes de crime

Après l'élaboration de nos éventuels modèles pour la problématique traitée, nous avons pu manipuler des données dites réelles. Ce sont des données prises directement sur les scènes de crimes, qui sont composées de 245 images. Les

images sont alors relativement moins consistantes et plus hétérogènes que les données de laboratoire. En effet, ces images sont donc issues de prises réalisées le plus souvent par la police scientifique, qui ne prend pas en compte les conditions consistantes de prise de photo respectées dans les données de laboratoire prises par Philippe Esperança. La Figure 3 montre des exemples d'images réelles. On peut dès maintenant s'apercevoir que ces données vont être beaucoup plus compliquées à analyser pour nos modèles de deep learning au vu des nombreux objets présents sur les photos.

(a) Modèle Volume Impacté (b) Modèle d'impact

FIGURE 3 – Deux images de scènes de crime

Nous avons réparti ces données réelles à 60% dans nos données d'entraînement, à 10% dans nos données de validation et à 30% dans nos données de test. En effet, une plus grande proportion d'images a été attribuée au test des données réelles afin d'avoir un test relativement plus représentatif. Cet ensemble de données de test est composé de 73 images. La Figure 4 nous montre la distribution des données réelles sur chacune des classes et chacun des datasets.

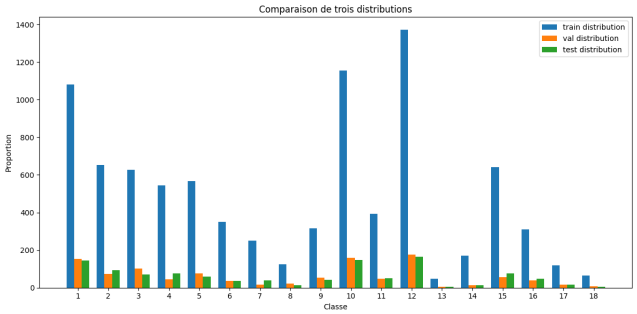


FIGURE 4 – Distribution des données de scène de crimes selon le dataset d'entraînement, de validation et de test.

2.3 Data Processing

Les images que l'on a reçues sont en couleurs et ont une taille variable allant de 2000×2000 à 10000×10000 pixels. Nous avons donc redimensionné toutes nos images en 256×256. Pour l'entraînement de nos modèles deep learning,

nous avons fait des symétries horizontales et verticales. Nous n'avons pas fait de rotation sur les images, car Philippe Esperança nous a expliqué qu'il ne prenait que des photos en face des taches de sang (donc sans rotation). Donc il n'est pas nécessaire d'apprendre les rotations des images. Nous avons aussi joué sur le contraste et la luminescence des images. Pour la validation, le test et l'inférence, nous n'avons pas mis de data augmentation.

3 Modèles

Pour aborder l'analyse de traces de sang, nous avons décidé d'utiliser le modèle Resnet 50 [2] pré-entraîné sur ImageNet selon certains poids indiqués dans la bibliographie [3], selon diverses approches. Nous avons utilisé la crossentropy pour notre fonction de coût, et Adam [4] pour l'optimizer.

3.1 ResNet linear probe

Nous avons dans un premier temps retiré la dernière couche dense du modèle, qui était destinée à classifier sur 1000 catégories, puis effectuer du linear probing. Nous avons donc gelé tous les poids du Resnet et nous avons remplacé la dernière couche dense par 2 couches denses (qui elles sont apprenables) pour avoir une dernière couche dense à 18 neurones [1]. La couche dense intermédiaire est de taille 64, et nous avons fixé un dropout à 0.1. La Figure 5 représente ce modèle linear probe ResNet. Les modèles utilisant le linear probing seront désignés par LP Resnet dans la suite.

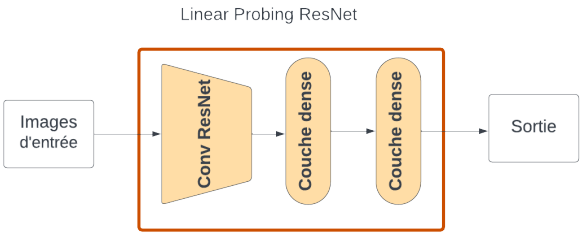


FIGURE 5 – Schéma du modèle LP Resnet

3.2 Réentraînement de tout le ResNet

Nous avons également testé la méthode LP ResNet sans geler les poids de convolution du ResNet. Les modèles concernés par cette méthode seront désignés par "AWL ResNet" pour All weight learnable.

3.3 Modèle Adversarial

Un des points important pour la classification d'image est de pouvoir détecter la tache de sang et de la détacher du fond (background). Nous avons alors

1. la sortie correspond alors aux 18 modèles de taches de sang.

implémenté un entraînement adversarial. En plus du modèle ResNet, nous avons rajouté un modèle de MLP (Multilayer Perceptron) composé de deux couches fully connected layer qui prend en entrée la sortie de l'avant-dernière couche dense de notre Linear Probe ResNet, et prédit le background. Le but est d'alors de faire en sorte que le modèle ResNet ne possède pas d'informations sur le background de l'image dans son espace latent. La Figure 6 montre ces deux modèles. Nous appellerons cette approche "modèle Adversarial".

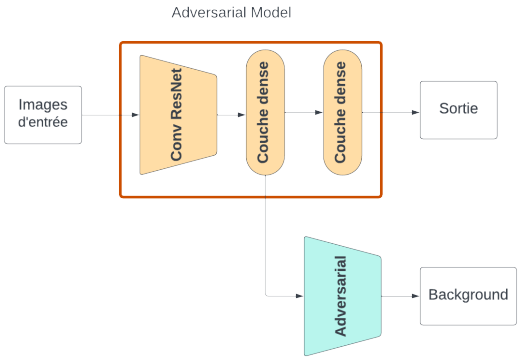


FIGURE 6 – Schéma du modèle Resnet adversarial

Nous avons fait un entraînement adversarial pour entraîner les deux modèles simultanément. Nous avons utilisé la fonction de coût définie dans la Formule 1, où CE_{tache} est la crossentropy entre la prédiction du modèle ResNet et la classe de la tache de sang, et $CE_{background}$ est la crossentropy de la prédiction du modèle adversarial et le background de l'image. Le paramètre α est un hyper-paramètre que l'on va optimiser dans la section 5.4.

$$L_{adv} = \frac{CE_{tache}}{\alpha CE_{background}} \tag{1}$$

La loss L_{adv} est rétro-propagée dans le modèle ResNet tandis que la loss $CE_{background}$ est alors propagée dans le modèle adversarial.

3.4 Fine-tune des modèles sur les données réelles

Une fois que nous avons appris nos modèles LP ResNet et AWL ResNet sur les données de laboratoires, nous avons fine-tune ces modèles sur les données réelles (données issues de scène de crime). Nous avons appelé ces modèles respectivement FT LP Resnet et FT AWL ResNet.

Nous n'avons pas pu fine-tune le modèle Adversarial, car les taches de sang ne sont pas sur les fonds unis, et par conséquent, il n'est pas possible de fine-tune le modèle prédicteur de background sur ces données.

4 Métriques

Pour comparer les performances de nos modèles, nous avons utilisé les métriques suivantes : l'accuracy micro (indique le pourcentage de réussite d'un

modèle), l'accuracy macro (indique la moyenne du pourcentage de réussite de chacune des classes), la précision, le rappel, le f1-score, le top 3 (indique le pourcentage de trouver la réponse parmi les 3 classes les plus prédites par le modèle). Toutes ces métriques sont des valeurs qui vont de 0 à 100, où 100 est le meilleur score. Les procédures de calcul de ces métriques sont explicitées en détail dans l'annexe B.1.

5 Apprentissage des modèles

5.1 Trouver le meilleur learning rate

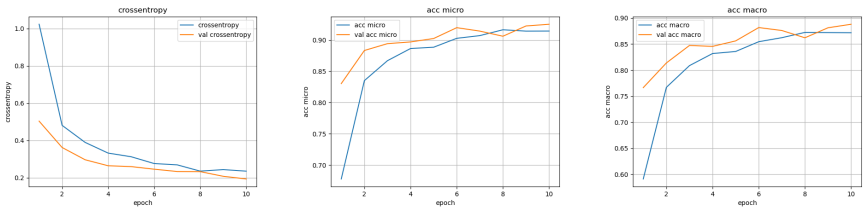
Afin de maximiser nos potentielles performances, nous avons commencé à effectuer un Grid-Search pour trouver le meilleur learning rate possible. Nous avons lancé l'entraînement du LP ResNet sur 5 epochs avec les learning rates suivants : $0.01, 0.005, 0.001, 0.0005, 0.0001$. La Table 2 montre alors les résultats de ce Grid-Search. Pour la suite, nous avons pris un learning rate à 0.0005.

learning rate	acc micro	acc macro	f1-score	top 3
0.01	85.1	78.8	78.1	49.4
0.005	89.1	84.7	83.6	**52.5**
0.001	90.4	85.7	84.8	50.5
0.0005	**91.1**	**86.8**	**86.0**	49.1
0.0001	84.9	78.6	77.4	50.4

TABLE 2 – Résultat de validation à la fin des entraînements des modèles LP ResNet avec différents learning rate.

5.2 Entraînement du modèle LP ResNet

Nous avons entraîné le modèle LP ResNet sur les données de laboratoire sur 10 epochs et un learning rate de 0.0005. La Figure 7 montre les courbes d'apprentissage de ce modèle.

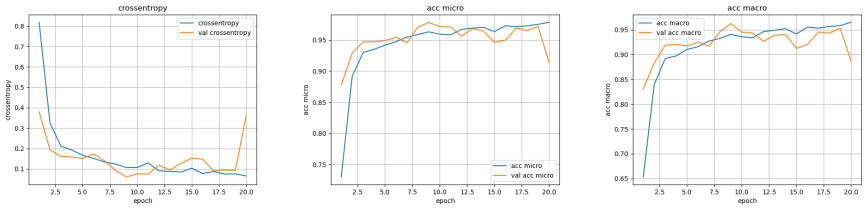


(a) crossentropy (b) acc micro (c) acc macro

FIGURE 7 – Valeurs de la loss et des accuracy d'entraînement (en bleue) et de validation (en orange) en fonction des epochs durant l'entraînement du modèle LP ResNet.

5.3 Entraînement du modèle AWL ResNet

Nous avons entraîné le modèle LP ResNet sur les données de laboratoire sur 20 epochs. Ici, nous avons mis un learning rate de 0.0001 pour qu'il apprenne moins vite et ne change pas trop ses paramètres dans les couches de convolutions. Nous n'avons pas pu faire une méthode de Grid Search comme le modèle de LP ResNet, en raison du temps d'apprentissage qui est plus long. La Figure 8 montre les courbes d'apprentissage de ce modèle.



(a) Crossentropy (b) acc micro (c) acc macro

FIGURE 8 – Valeurs de la loss et des accuracy d'entraînement (en bleue) et de validation (en orange) en fonction des epochs durant l'entraînement du modèle AWL ResNet.

5.4 Trouver les hyperparamètres pour le modèle Adversarial

On cherche ici à trouver le meilleur learning rate pour optimiser le ResNet lr_{res} et pour celui adversarial lr_{adv}. On cherche aussi le meilleur paramètre α, utilisé dans la formule 1. On a alors implémenté un Random Search [5] et fait 10 expériences de 5 epochs, en prenant à chaque fois des paramètres dans la Table 3 [2]

Paramètres	Valeurs possibles
lr_{res}	0.01, 0.005, 0.001, 0.005, 0.0001
lr_{adv}	0.01, 0.005, 0.001, 0.005, 0.0001
α	0.001, 0.1, 0.5, 1, 2, 5, 10, 100

TABLE 3 – Liste des valeurs possibles pour les hyperparamètres testés dans le Random Search

La Table 4 montre les résultats de ce Random Search. *res acc micro* représente l'accuracy micro de la partie ResNet (de même pour *res acc macro*), et *adv acc micro* représente l'accuracy micro de la partie Adversarial. Nous avons donc choisi de prendre les hyperparamètres de l'expérience numéro 2.

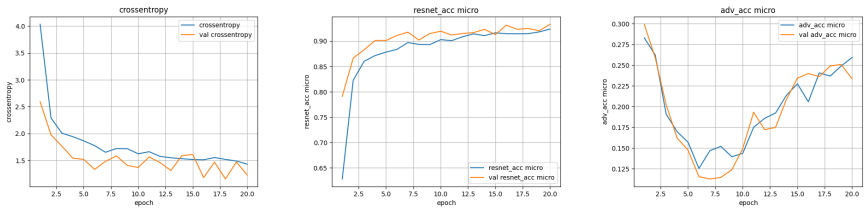
2. On a fait un Random Search et pas un Grid-Search, car il n'est pas possible de tester toutes les combinaisons possibles, qui sont au nombre de 200.

numéro	res acc micro	res acc macro	adv acc micro	lr_{res}	lr_{adv}	α
0	79.3	72.8	82.5	0.1	1	10
1	89.6	84.7	16.1	0.1	1	0.1
2	**90.8**	**86.2**	20.2	0.5	0.01	0.1
3	85.3	81.2	44.6	0.1	1	0.5
4	88	85.7	72	0.5	0.5	2
5	89.4	85.4	56.8	0.01	0.1	0.5
6	89.5	85.4	71	0.1	0.1	1
7	87.2	82.2	71.7	1	0.01	1
8	85.3	80.3	**85.3**	0.1	0.5	10
9	86.7	83.2	84.8	0.1	0.01	10

TABLE 4 – Résultat de validation à la fin des entraînements des modèles Adversarial avec différents hypermaramètres.

5.5 Entraînement du modèle Adversarial

Nous avons donc entraîné le modèle Adversarial sur les données de laboratoire sur 20 epochs avec les hyperparamètres trouvés dans la section 5.4. La Figure 9 montre les courbes d'apprentissages de ce modèle.

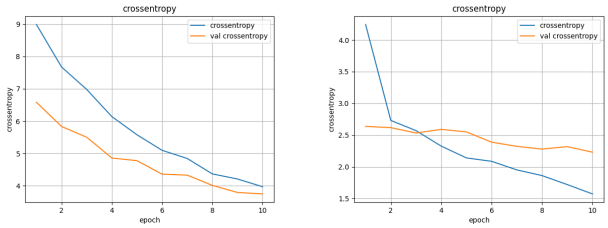


(a) Loss L_{adv} (b) res acc micro (c) adv acc micro

FIGURE 9 – Valeurs de la loss et des accuracy d'entraînement (en bleue) et de validation (en orange) en fonction des epochs durant l'entraînement du modèle Adversarial.

5.6 Fine-tune des modèles LB ResNet et AWL ResNet

Nous avons ensuite appris les modèles FT LB ResNet et FT AWL ResNet sur 10 epochs qui sont des modèles fine-tune sur les données réelles à partir des modèles respectivement LB ResNet et AWL ResNet sur 10 epochs. La Figure 10 montre les courbes d'apprentissages de ce modèle.



(a) crossentropy de FT LP ResNet (b) crossentropy de FT AWL ResNet

FIGURE 10 – Valeurs de la loss d'entraînement (en bleue) et de validation (en orange) en fonction des epochs durant l'entraînement des modèles fine-tune sur les données réelles.

6 Résultats

Les résultats de test sur les données de laboratoire présent dans la Table 5 indiquent de meilleures performances pour le modèle Resnet qui a été réentraîné sur tous ses poids, selon l'accuracy et le F1-score. On peut également constater que les modèles fine-tunés sur les données rélles ont des performances moins bonnes que les modèles réentraînés.

Modèles	Acc Micro	Acc Macro	F1-score	Top 3
LP ResNet	95.2	94.3	94.7	**99.9**
FT LP ResNet	83.9	86.2	80.4	98.3
AWL ResNet	**97.3**	**97.1**	**96.2**	**99.9**
FT AWL ResNet	76.4	76.1	70.7	93.8
Adversarial	93.4	91.8	91.8	**99.9**

TABLE 5 – Résultats de test sur les données de laboratoire

Les résultats de test sur les données de laboratoire présent dans la Table 6 indiquent de meilleures performances pour le modèle Resnet qui était entraîné entièrement sur les données de laboratoire, puis fine-tuné sur les données réelles, selon l'accuracy et le F1-score. Nous avons également une valeur de Top 3 à 51.7%, ce qui signifie qu'il y a une chance sur deux que la bonne réponse soit présente dans la prédiction.

Modèles	Acc Micro	Acc Macro	F1-score	Top 3
LP ResNet	12.9	6.0	4.0	30.1
FT LP ResNet	11.8	6.1	6.4	36.6
AWL Resnet	17.2	13.8	8.1	30.1
FT AWL ResNet	**41.9**	**33.4**	**26.9**	**67.7**
Adversarial	11.8	5.7	3.7	23.7

TABLE 6 – Résultats de test sur les données réelles

Le modèle Adversarial a des performances assez décevantes par rapport à un modèle Resnet, donc le modèle Adversarial n'est pas le plus adapté pour notre problématique.

Le AWL Resnet a étonnamment de meilleures performances que le modèle Resnet appliquant le Linear Probing.

7 Interprétabilité avec les cartes de saillance

Afin de répondre à l'aspect "boîte noire" des réseaux de neurones, nous avons implémenté une interprétabilité dans notre modèle à l'aide de Grad-CAM [6]. Cette méthode permet de fournir une explication visuelle vis-à-vis des décisions de classification issue de notre modèle, permettant ainsi de rajouter une certaine légitimité relative face à nos analyses de traces de sang faisant partie d'un processus judiciaire.

7.1 Visualisation avec Grad-CAM

L'algorithme Grad-CAM (Gradient-weighted Class Activation Mapping) est une technique utilisée pour rendre les réseaux de neurones convolutifs (CNN) plus interprétables dans le domaine de la classification. Il fonctionne en générant des cartes de chaleur (heatmaps) qui mettent en évidence les zones importantes d'une image qui contribuent le plus à la prédiction d'une classe spécifique. Pour ce faire, Grad-CAM calcule les dérivées des scores de sortie de la classe cible par rapport aux caractéristiques de la dernière couche convolutive du CNN. Ces dérivées sont ensuite globalement moyennées pour obtenir les poids d'importance de chaque carte de caractéristiques. Enfin, les cartes de caractéristiques sont pondérées par les poids d'importance et combinées pour obtenir la carte de chaleur Grad-CAM, qui peut être superposée à l'image d'origine pour une visualisation plus intuitive. Cette approche permet de mieux comprendre les décisions prises par le CNN et d'améliorer la confiance dans les prédictions du modèle. La Figure 11 montre un exemple d'une image de trace de sang avec sa carte de chaleur Grad-CAM superposée.

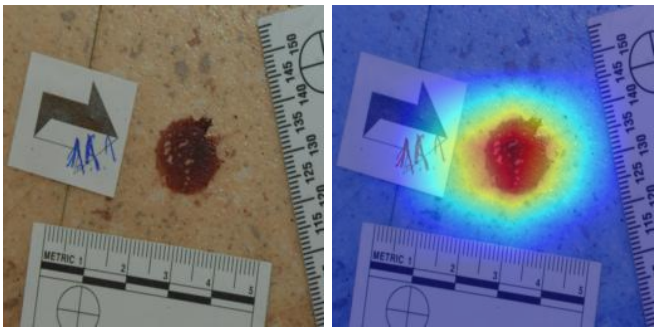


FIGURE 11 – Exemple d'une image de trace de sang (à gauche) avec sa carte de chaleur Grad CAM superposée (à droite).

En utilisant Grad-CAM, nous avons pu identifier que le modèle se focalisait parfois sur ces réglettes pour prendre sa décision, ce qui nous a permis de mieux comprendre les erreurs de classification, comme le montre la Figure 12.

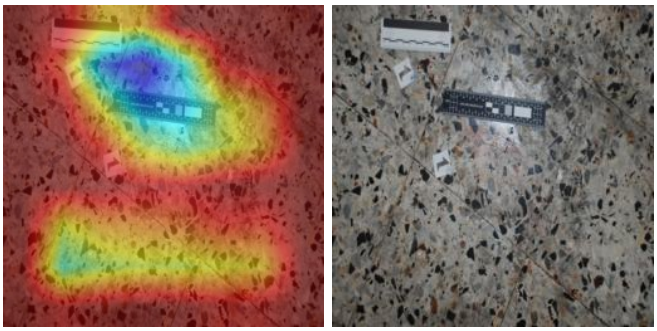


FIGURE 12 – Exemple d'une image de trace de sang (à droite) avec sa carte de chaleur Grad CAM superposée (à gauche), dans le cadre d'une attention portée à la réglette.

7.2 Métriques de saliency maps

De plus, nous avons également implémenté les métriques Average Drop, Average Increase et Average Gain [7], qui consiste à multiplier l'image de départ par la carte de saillence obtenue avec Grad-CAM puis de redonner cette image au modèle pour obtenir une nouvelle prédiction. On va alors comparer le changement de probabilité de la classe prédite avant et après le masquage de l'image. Ces métriques permettent de quantifier l'exactitude des cartes de saillance obtenues. La section B.2 présente les formules permettant de les calculer.

L'Average Drop (AD) quantifie la perte de pouvoir prédictif, mesurée en termes de probabilité de classe, lorsque nous masquons uniquement l'image. Elle est comprise entre 0 et 100, et une valeur plus faible est meilleure.

L'Average increase (AI) également connu sous le nom d'increase in confidence, mesure le pourcentage d'images pour lesquelles l'image masquée donne une probabilité de classe plus élevée que l'image originale. Elle est comprise en 0 et 100, et une valeur plus haute est la meilleure.

L'Average Gain (AG) quantifie le pouvoir prédictif, mesuré en tant que probabilité de classe, lorsque nous masquons l'image. Elle est comprise entre 0 et 100, et une valeur plus élevée est meilleure.

La Table 7 montre les scores de ses métriques sur la base de données de test des données réelles [3].

Métriques	Average Drop	Average Increase	Average Gain
AWL ResNet	91.6	0.0	0.0
FT AWL ResNet	87.6	0.0	0.0

TABLE 7 – Résultats des métriques sur les données réelles (de la base de données de test).

On peut voir dans la Table 7 des résultats très peu satisfaisant. En effet, les deux modèles ont une Average Increase et une Average Gain de 0, ce qui signifie qu'avec les notations de la section B.2, on a toujours $\forall i \in [1, N], p_i < o_i$.

3. On a effectuer les tests seulement sur les modèles AWL ResNet car c'est les seules dont les poids des couches de convolution ont été appris

Cela signifie qu'en multipliant l'image d'origine par la carte de saillance, on obtient toujours une probabilité de classe plus faible que l'image d'origine. Cela montre que ces cartes de saillance ne sont pas fiables. Une amélioration possible serait d'utiliser des méthodes de saliency maps plus performantes comme Grad-CAM++ [8] ou Score-CAM [9].

8 Conclusion

Le modèle sur lequel nous avons abouti représente un potentiel. Il présente de bonnes performances sur les données de laboratoires, 97% d'accuracy (micro) contre 41% avec les données réelles, sachant qu'il y a 10 000 images de laboratoire contre environ 300 images réelles. D'autres pistes d'améliorations possibles concernent le modèle Resnet utilisé. Celui qui a été manipulé est le Resnet50, qui est relativement petit par rapport à d'autres Resnet plus grands existants, comme ResNet50x4, ResNet50x16. Il y a également peu de résultats concluants et significatifs à travers les cartes de saillance avec le Grad cam utilisé. Il serait éventuellement plus judicieux d'utiliser des méthodes plus sophistiqué comme Grad cam ++ ou Score cam. Une autre approche qu'on pourrait envisager est d'utiliser de l'apprentissage auto-supervisé pour exploiter l'ensemble des images non classées que possède l'expert criminalistique. En effet, depuis 10 ans, il accumule une très grande quantité d'images de scènes de crime, qui n'a pas le temps de classer, qui pourraient être utilisées pour entraîner un modèle de deep learning.

Nous avons eu l'occasion de tester d'autres approches qui sont décrites plus en détail dans la section C. Parmi celles-ci, figurait l'utilisation de la clé de détermination élaborée par Philippe Esperança. Cette méthode consiste à analyser différentes caractéristiques observées sur l'image de la trace de sang, telles que la présence ou non d'une forme ovale, afin de classifier les traces de sang en différentes catégories selon les indications de la clé de détermination. Toutefois, les méthodes examinées n'ont pas donné de résultats concluants.

Notre encadrant a pu recevoir notre projet sous la forme d'une interface qui fonctionne localement sur son appareil de travail. Ce modèle d'analyse de traces de sang ne doit pas avoir accès à internet afin d'éviter tout risque d'attaque ou de fuite de données. Pouvoir faire tourner le projet en local pour l'expert criminalistique lui permet de nous faire confiance vis-à-vis de l'outil implémenté. Malgré des prédictions et des interprétations instables dans le contexte de notre projet, il a pu néanmoins apprécier et être satisfait de notre rendu dans son ensemble pour les objectifs qu'il remplit.

Notre projet va faire l'objet d'une reprise dans le cadre d'un sujet de stage au laboratoire LIS. Ainsi, notre projet possède un GitHub public ainsi qu'une documentation détaillée afin que les prochaines personnes en charge puissent reprendre aisément notre travail.

Références

[1] P. Esperança, *Morphoanalyse des traces de sang : Une approche méthodique*, Édition. PU POLYTECHNIQU, 2019.

[2] K. He, X. Zhang, S. Ren et J. Sun, « Deep Residual Learning for Image Recognition, » *CoRR*, t. abs/1512.03385, 2015. arXiv : 1512.03385. adresse : http://arxiv.org/abs/1512.03385.

[3] T. maintainers et contributors, *TorchVision : PyTorch's Computer Vision library*, https://github.com/pytorch/vision, 2016.

[4] D. P. Kingma et J. Ba, « Adam : A method for stochastic optimization, » *arXiv preprint arXiv :1412.6980*, 2014. adresse : https://arxiv.org/abs/1412.6980.

[5] J. Bergstra et Y. Bengio, « Random search for hyper-parameter optimization., » *Journal of machine learning research*, t. 13, n° 2, 2012.

[6] R. R. Selvaraju, A. Das, R. Vedantam, M. Cogswell, D. Parikh et D. Batra, « Grad-CAM : Why did you say that ? Visual Explanations from Deep Networks via Gradient-based Localization, » *CoRR*, t. abs/1610.02391, 2016. arXiv : 1610.02391. adresse : http://arxiv.org/abs/1610.02391.

[7] H. Zhang, F. Torres, R. Sicre, Y. Avrithis et S. Ayache, *Opti-CAM : Optimizing saliency maps for interpretability*, 2024. arXiv : 2301.07002 [cs.CV].

[8] A. Chattopadhyay, A. Sarkar, P. Howlader et V. N. Balasubramanian, « Grad-CAM++ : Generalized Gradient-based Visual Explanations for Deep Convolutional Networks, » *CoRR*, t. abs/1710.11063, 2017. arXiv : 1710.11063. adresse : http://arxiv.org/abs/1710.11063.

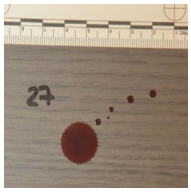
[9] H. Wang, M. Du, F. Yang et Z. Zhang, « Score-CAM : Improved Visual Explanations Via Score-Weighted Class Activation Mapping, » *CoRR*, t. abs/1910.01279, 2019. arXiv : 1910.01279. adresse : http://arxiv.org/abs/1910.01279.

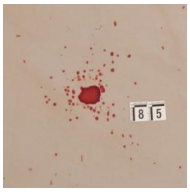
[10] O. Ronneberger, P. Fischer et T. Brox, « U-Net : Convolutional Networks for Biomedical Image Segmentation, » *CoRR*, t. abs/1505.04597, 2015. arXiv : 1505.04597. adresse : http://arxiv.org/abs/1505.04597.

[11] A. Radford, J. W. Kim, C. Hallacy et al., « Learning Transferable Visual Models From Natural Language Supervision, » *CoRR*, t. abs/2103.00020, 2021. arXiv : 2103.00020. adresse : https://arxiv.org/abs/2103.00020.

[12] A. Kirillov, E. Mintun, N. Ravi et al., *Segment Anything*, 2023. arXiv : 2304.02643 [cs.CV].

A Images de laboratoire


1- Traces passives


2- Goutte à Goutte


3- Transfert par contact


4- Transfert glissé


5- Altération par contact


6- Altération glissée


7- Accumulation


8- Coulée

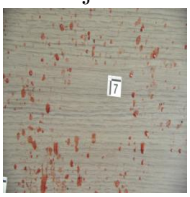

9- Chute de volume

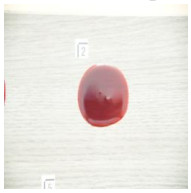

10- Sang Propulsé


11- éjection


12- Volume impacté


13- Imprégnation


14- Zone d'interruption


15- Modèle d'impact


16- Foyer de modèle d'impact


17- Trace gravitationnelle


18- Sang expiré

TABLE 8 – Classe des données de laboratoire et leur exemple en image

B Calcul des métriques

B.1 Evaluation des modèles

B.1.1 Accuracy micro

L'accuracy est le pourcentage de réussite d'un modèle. Elle est calculée par la formule suivante :

$$\text{Accuracy micro} = \frac{\text{Nombre de prédictions correctes}}{\text{Nombre total de prédictions}} \tag{2}$$

B.1.2 Accuracy macro

L'accuracy macro est la moyenne des accuracy de chaque classe. Elle est calculée par la formule suivante :

$$\text{Accuracy macro} = \frac{1}{N} \sum_{i=1}^{n} \frac{\text{Nombre de prédiction correctes de la classe } i}{\text{Nombre total de prédictions de la classe } i} \tag{3}$$

B.1.3 Précision

La précision est le pourcentage de prédictions correctes parmi les prédictions positives. Elle est calculée par la formule suivante :

$$\text{Précision} = \frac{\text{Nombre de vrais positifs}}{\text{Nombre de vrais positifs} + \text{Nombre de faux positifs}} \tag{4}$$

B.1.4 Rappel

Le rappel est le pourcentage de prédictions correctes parmi les vrais labels positifs. Il est calculé par la formule suivante :

$$\text{Rappel} = \frac{\text{Nombre de vrais positifs}}{\text{Nombre de vrais positifs} + \text{Nombre de faux négatifs}} \tag{5}$$

B.1.5 F1-score

Le F1-score est la moyenne harmonique de la précision et du rappel. Il est calculé par la formule suivante :

$$\text{F1-score} = 2 \times \frac{\text{Précision} \times \text{Rappel}}{\text{Précision} + \text{Rappel}} \tag{6}$$

B.1.6 Top 3

Le Top 3 est le pourcentage de trouver la réponse parmi les 3 classes les plus prédites par le modèle. Il est calculé par la formule suivante :

$$\text{Top 3} = \frac{\text{Nombre de prédictions correctes parmi les 3 premières prédictions}}{\text{Nombre total d'exemples}} \tag{7}$$

B.2 Evaluation des cartes de saillance

Soit $p_i = \arg\max_c p_i^c$ la probabilité de la classe prédicte par le modèle pour l'image x_i. Soit o_i la probabilité de cette même classe prédite par le modèle pour l'image maské de ca carte de saillance $x_i * s_i$ (où s_i est la carte de saillance de l'image x_i). On définie N le nombre totale d'image dans le test set. On note $[x]_+$ le maximun entre 0 et la valeur x ($[x]_+ = \max(0, x)$). On définie les métriques suivantes :

B.2.1 Averare Drop (AD)

L'Average Drop (AD) quantifie la perte de pouvoir prédictif, mesurée en termes de probabilité de classe, lorsque nous masquons uniquement l'image image. Elle est comprise entre 0 et 100, et une valeur plus faible est meilleure.

$$\text{AD} = \frac{1}{N} \sum_{i=1}^{N} \frac{[p_i - o_i]_+}{p_i} \cdot 100 \tag{8}$$

B.2.2 Average Increase (AI)

L'Average increase (AI) également connu sous le nom d'increase in confidence, mesure le pourcentage d'images pour lesquelles l'image masquée donne une probabilité de classe plus élevée que l'image originale. Elle est compris en 0 et 100, et une valeur plus haute est la meilleur.

$$\text{AI} = \frac{1}{N} \sum_{i=1}^{N} 1_{p_i < o_i} \cdot 100 \tag{9}$$

B.2.3 Average Gain (AG)

L'Average Gain (AG) quantifie le pouvoir prédictif, mesuré en tant que probabilité de classe, lorsque nous masquons l'image. Elle est comprise entre 0 et 100, et une valeur plus élevée est meilleure.

$$\text{AG} = \frac{1}{N} \sum_{i=1}^{N} \frac{[o_i - p_i]_+}{1 - p_i} \cdot 100 \tag{10}$$

C Pistes infructueuses

Au cours de ce projet, nous avons exploré plusieurs pistes qui se sont avérées infructueuses. Nous les présentons ici pour que le lecteur puisse comprendre les raisons pour lesquelles nous avons choisi de ne pas les poursuivre.

Un des objectifs de ce projet était de parvenir à interpréter les choix des modèles de machine learning que nous avons entrainé. Pour cela, nous avons tenté de nous inspirer du travail réalisé précédemment par l'équipe d'experts criminalistiques. En effet, leur approche s'inspire de la classification des espèces animales et végétales par les biologistes. Ils ont cherché à identifier des caractéristiques déterminantes pour chaque trace de sang et ainsi à classifier les traces de sang grâce à un arbre de décision portant sur ces caractéristiques (par exemple :

la forme, la taille, présence de tâches millimétriques près des tâches centimétriques, etc..). Nous avons donc tenté de reproduire cette approche en utilisant des techniques de feature ingenering pour extraire des caractéristiques fondamentales des traces de sang souis formes de critères mathématico-géométriques. Pour cela, nous avons d'abord tenté de réaliser un algorithme de segmentation non supervisé par détection de contours pour extraire les formes des taches de sang, avec des résultats peu concluants. Nous avons ensuite tenté de réaliser un algorithme mêlant des techniques de traitement d'images (seuils, etc..), d'apprentissage auto-supervisé (Unet [10]) et de géométrie pour extraire des caractéristiques géométriques telle la taille des tâches satellitaires par rapport à la tâche centrale, la forme des tâches (critère d'ovoïdité, de circularité, etc..). Ces tentatives ont néanmoins été infructueuses, car les masques de segmentation étaient trop imprécis du fait de la grande variabilité des images de taches de sang. Nous avons également tenté des méthodes de segmentation adversarial en tirant parti du fait que nous connaissions pour toutes les images la nature du support (bois, lino, carrelage, etc..) pour tenter de segmenter les taches de sang par une approche faiblement supervisée. Ces tentatives ont par ailleurs été infructueuses, car nous avions peu de donnée d'entrainement, ce qui rendait tout apprentissage trop instable de manière faiblement supervisée.

Nous avons aussi tenté d'extraire ces caractéristiques (masques de segmentation, critères géométriques) à l'aide de modèle de deep learning utilisant des techniques dites de "zero-shot classification" (modèles de type CLIP-ViT [11], etc..) ou zero-shot segmentation (SegmentAnything [12]). Ces tentatives ont également été infructueuses, car le manque de contraste sur certaines images de taches de sang ou leur trop grand éclatement rendait les masques de segmentation trop imprécis.