

ANALYSE AUTOMATIQUE DE TRACES DE SANG

GHANNAY Nesrine

KARKACH Clara

Aix-Marseille Université

M1 Informatique - parcours Intelligence Artificielle et Apprentissage Automatique

RESUMÉ

Le Laboratoire d'Analyse criminalistique (LAC) travaille en collaboration avec l'équipe Qarma du Laboratoire d'Informatique et Système (LIS), sur le développement d'une IA capable de catégoriser de manière interprétable les traces de sang provenant de scène de crimes. L'objectif principal de ce projet est d'améliorer les modèles existant sur des données produites en laboratoire et des données réelles de scènes de crimes. Nous explorons donc des méthodes d'adaptation et de transfert de domaines avec des architectures telles que DANN et ADDA, et des modèles comme t-ReX. Enfin, le système produit inclura une interface répondant aux besoins spécifiques du LAC, facilitant le traitement des données par les experts.

Mots clés— Vision par ordinateur, Deep Learning, Transfert et Adaptation de domaine, Détection d'objets, Carte de salliance.

1. INTRODUCTION

Le Laboratoire d'Analyses Criminalistiques (LAC) est un laboratoire spécialisé en criminalistique générale et en morphoanalyse des traces de sang. La méthodologie développée par Philippe Esperança, expert criminalistique au sein du LAC, permet d'analyser des traces de sang en extrayant leurs caractéristiques clés (forme, taille, dispersion, etc.). Elle permet finalement de déterminer de manière méthodique les événements à l'origine d'un crime et de mieux comprendre le déroulement des faits, ou encore le type d'arme utilisé. D'autres part, des précédentes études faites par le LAC, leur ont permis d'établir un Atlas, un document permettant de catégoriser les traces de sang selon une série de critères spécifiques [1]. Dans ce document une trentaine de classes sont définies, mais nous nous concentrerons sur la prédiction de 18 classes seulement et ce pour des raisons de manque de données. Ainsi, selon une description détaillée et précise des critères morphologiques d'une tâche, nous pouvons l'associer à une des 18 classes détaillées dans le tableau 4 en Annexes.

C'est dans ce contexte que le LAC, s'est tourné vers l'équipe Qarma du Laboratoire d'Informatique et Systèmes

(LIS), pour une collaboration dans le but de développer un système à l'aide de l'intelligence artificielle, destiné à classer les traces de sang de manière interprétable. Ceci afin de faciliter la prise de décision dans un contexte d'analyse de scène de crime. Par ailleurs, étant donné qu'une étude préliminaire a été réalisée sur ce projet par des élèves de l'École de Centrale Marseille [2], il est possible de définir plus en détail les trois principaux besoins auxquels nous essaierons de répondre durant notre stage :

Amélioration des Performances sur Données Réelles.

Premièrement, les modèles entraînés ont jusqu'à présent produit de très bons résultats sur une base de données conséquente, composée uniquement de photos reproduites en laboratoire. Cependant, lors de tests sur des données réelles provenant de scènes de crimes, les performances des modèles étaient insuffisantes. Nous explorerons ainsi d'autres architectures de réseau de neurones profonds afin de résoudre ce problème.

Amélioration de l'Interprétation des Prédictions. De plus, nous essaierons d'améliorer l'interprétation des prédictions, qui joue un rôle crucial dans la compréhension du fonctionnement du modèle d'analyse de traces de sang. Elle servira d'outil de vérification pour les experts.

Développement d'une Interface. Enfin, la dernière partie de ce projet se concentre sur la conception et le développement d'une interface conviviale et fonctionnelle, répondant aux besoins spécifiques du LAC.

Après avoir exposé les objectifs de notre stage ainsi que la méthodologie développée par Philippe Esperança pour l'analyse des traces de sang, il est essentiel de se pencher sur les technologies qui permettront d'atteindre ces objectifs.

2. ÉTAT DE L'ART

La vision par ordinateur représente le fondement technologique sur lequel repose notre projet de recherche. Elle est définie comme étant une sous-branche de l'intelligence artificielle qui a pour objectif de reproduire le comportement humain, vis à vis de leur façon de voir, reconnaître, et traiter les images. Pour cela sont utilisés les réseaux de neurones convolutionnels (CNN), qui sont des architectures de réseau de neurones permettant une reconnaissance et une analyse

efficace des informations de données visuelles. Nombreuses sont les architectures appartenant à cette catégorie de réseaux, et parmi les modèles les plus célèbres nous pouvons citer ResNet (ou Réseaux Résiduel). La particularité de ce CNN, est sa capacité à apprendre des identités résiduelles (ou résidu), permettant ainsi la formation de réseaux très profonds en évitant le phénomène de la disparition du gradient, et offrant ainsi de très bons résultats en généralisation [3].

D'autres approches ont également été combinées dans notre étude, dont les techniques d'adaptation et de transfert de domaine, à travers les architectures DANN et ADDA [4]. Ces méthodes devraient nous permettre de passer du domaine d'entraînement (noté D_s) à un domaine test cible (indiqué par la lettre D_t), et ce de manière à avoir tout de même de bon résultats en généralisation. L'emploi de cette méthode est très utile lorsqu'on a une grande quantité de données source labélisées, peu de données appartenant au domaine cible, et que ces deux domaines suivent des distributions différentes mais similaires. Parmi les méthodes d'adaptation de domaine, nous retrouvons **L'architecture DANN (Domain Adaptation Neural Network)** qui est composé de trois architectures clés (voir la figure 4 en Annexe):

1. L'Extracteur de caractéristique : Il permet d'apprendre les représentations utiles des images et capturer les caractéristiques les plus importantes pour la classification.
2. L'Invariance de domaine : Il permet de distinguer le domaine d'une entrée, en prenant en entrée la sortie de l'extracteur de caractéristique et prédit à partir de cela, à quel domaine la donnée appartient (D_s ou D_t).
3. Le prédicteur de classe : Permet de prédire la classe associée à une entrée à partir de sa représentation intermédiaire.

Ainsi, la combinaison de ces trois composantes permet de rendre les représentations des données moins dépendantes aux domaines, afin de mieux généraliser sur des images provenant des deux domaines à la fois. Le principe clé est d'inverser le gradient lors de sa rétro-propagation dans le réseau de neurones via une couche d'inversement du gradient. Sa fonction de perte est donc la suivante :

$$crossloss = regression_{loss} - \lambda \cdot adversarial_{loss}$$

Ceci permet au modèle d'apprendre des représentation qui minimisent à la fois la perte de classification ($regression_{loss}$) tout en maximisant la perte de domaine ($adversarial_{loss}$). [4][5].

L'architecture ADDA (Adaptation de Domaine Adversarial Discriminative) repose également sur l'extraction et l'adaptation des caractéristiques, mais diffère par son

approche et sa structure. Elle a deux encodeurs principaux (voir la figure 5 en Annexe):

1. Encodeur source (G_s): Pré-entraîné sur les données du domaine source D_s pour apprendre des représentations visuelles de ce domaine.
2. Encodeur cible (G_t): Initialisé aléatoirement avec les poids de l'encodeur source. Ensuite, un entraînement adversarial est effectué, impliquant un discriminateur (D) dont le rôle est de distinguer les deux domaines.

L'objectif d'ADDA est d'ajuster l'encodeur cible pour tromper le discriminateur, améliorant ainsi la généralisation. Lors des tests, les données du domaine cible (D_t) sont encodées par l'encodeur cible puis classifiées par un classifieur (C) déjà entraîné sur D_s . Cette méthode optimise les performances du modèle en réduisant l'écart entre les distributions des deux domaines, minimisant ainsi la perte de classification et maximisant la confusion du domaine [6]. La perte totale d'ADDA est la somme de la perte de classifieur sur le domaine source et de la perte de discrimination entre les données du domaine source et celles du domaine cible (1) .

$$\mathcal{L}_{ADDA} = \mathcal{L}_{src}(G_s, C) + \mathcal{L}_{adv}(D, G_t) \quad (1)$$

Enfin, la détection d'objets pourrait améliorer la reconnaissance des traces de sang. Cette technique identifie et localise automatiquement des objets dans une image ou une vidéo via des réseaux neuronaux profonds. Une méthode efficace est **t-ReX (Transformers for End-to-end Object Detection with feature eXchanges)**, qui utilise des prompts visuels plutôt que textuels. Les composants principaux de t-ReX incluent : un encodeur d'images qui extrait les caractéristiques visuelles de l'image cible et de référence, un encodeur de prompts qui utilise des indications visuelles pour extraire des prompts encodés, et un décodeur de boîtes qui combine ces informations pour produire des boîtes détectées avec leurs scores de confiance. [7]. Cette méthode produit des résultats précis et robustes.

3. MÉTHODOLOGIE

Tout d'abord, nous avons eu plusieurs réunions avec nos encadrants et l'expert M. Esperança, qui nous ont permis de répartir entre nous deux les différentes tâches afin d'atteindre nos objectifs, comme illustré dans le diagramme de Gantt en annexe, figure 7. Nous pouvons classer ces tâches en deux catégories : la reprises des travaux de l'équipe précédente [2] et les nouveaux travaux réalisés, les nouvelles démarches entreprises.

3.1. Travaux précédant

Nous avons d'abord analysé l'étude précédente [2] pour assimiler les notions clés de la vision par ordinateur et affiner nos objectifs. Ensuite, nous avons étudié le premier jeu de données du LAC, dont deux parties, 8186 images reproduites en laboratoire (données laboratoire) et 244 images de scènes de crime réelles triées par M. Esperança (données réelles). Mise à part le déséquilibre entre les données simulées et réelles, ainsi qu'entre les différentes classes, ce qui rajoute davantage de difficulté, nous avons repris les modèles de l'étude précédente. Ils avaient utilisé un ResNet18 pré-entraîné sur ImageNet, modifié pour classifier 18 catégories [8], produisant ainsi cinq modèles différents.

1. **LP ResNet (Linear Probing)** : ResNet adapté où l'on apprend les poids des deux dernières couches seulement.
2. **AWL ResNet (All Weight Learnable)** : Méthode LP ResNet sans geler les poids de convolution du ResNet (on apprend tous les poids).
3. **FT LP ResNet** : Fine-tuning de la méthode LP ResNet. On prend le modèle pré-entraîné sur les données laboratoire (LP ResNet) et on l'affine avec les données réelles.
4. **FT AWL ResNet** : Fine-tuning de la méthode AWL ResNet sur les données réelles.
5. **Adversarial** : Combinaison du modèle ResNet et un modèle MLP (Multilayer Perceptron), qui est un type de réseau neuronal artificiel organisé en plusieurs couches. Ce modèle comporte deux couches fully connected (couches entièrement connectées) qui prennent en entrée la sortie de l'avant-dernière couche dense de notre LP ResNet, et prédisent le background. Il permet ainsi de ne plus prendre en considération l'arrière plan de l'image.

Les données ont été divisées en ensembles d'entraînement, de test et de validation, tout en procédant par la suite à une augmentation des données par rotations, flips et ajustements de luminosité. Les modèles ont ensuite été testés séparément sur les données de laboratoire et les données réelles. Les meilleurs résultats ont été obtenus avec les modèles AWL ResNet, atteignant 97% de précision sur les données de laboratoire et 17,2% sur les données réelles, et FT AWL ResNet, avec 76,4% sur les données de laboratoire et 41,9% sur les données réelles. Nous avons constaté que la grande différence entre les résultats s'explique par la dispersion des données, le manque de données réelles (voir le tableau 5), et le fait que les données réelles sont très différentes des données laboratoire (voir un échantillon des données de vraies photos de sang figure 6 en Annexe).

3.2. Travaux réalisés

Les données : Pour remédier à ce problème, nous avons eu l'opportunité d'accéder à de nouvelles données réelles labellisées mais non triées. Ces données proviennent de cas réels de scènes de crime non encore triées, comprenant des images numérotées et des rapports d'expertise avec la référence de l'image et sa classe. Après avoir trié ces données, nous avons ajouté 664 nouvelles images. Nous avons ensuite effectué un pré-traitement des données en redimensionnant les images en 256×256 pixels, puis les avons divisées en ensembles d'entraînement, de validation et de test avec les proportions suivantes : 80%, 10%, 10% pour les données de laboratoire, et 60%, 10%, 30% pour les données réelles. Malgré ce travail, la différence de quantité entre les données de laboratoire et les données réelles reste significative, de même que leur répartition parmi les différentes classes (voir 5).

Les nouveaux modèles : Par la suite nous avons abordé les architectures de transfert de domaine, DANN et ADDA, en utilisant nos données de laboratoire comme domaine source et nos données de cas réels comme domaine cible. L'implémentation de DANN a par la suite été produite en reprenant le modèle adversarial de l'étude précédente de la manière suivante. Nous avons ajouté à un modèle ResNet18 de base, un modèle de MLP (Multilayer Perceptron) composé de deux couches fully connected et qui prend en entrée la sortie de l'avant-dernière couche dense du modèle Linear Probe ResNet. Il permet ainsi de prédire le domaine de l'image d'entrée : données réelle ou données laboratoire. Par la suite nous avons trouvé le meilleur learning rate pour optimiser le ResNet lr_{res} et celui pour la partie adversarial lr_{adv} , ainsi que le coefficient multiplicateur du gradient inversé : λ . Par manque de temps, nous avons seule utilisé un Random Search [9] à la place d'un GridSearch. Les hyper-paramètres de chaque modèles seront présentés dans section 3.

Par ailleurs, concernant ADDA et t-ReX nous sommes toujours en train de les implémenter, donc leurs performances ne seront pas renseignées dans ce rapport. [10].

Les Résultats : Afin de comparer l'ensemble des modèles cités nous avons effectué différents benchmarks en utilisant un cluster informatique (le cluster du LIS). Chacun des modèles a été entraîné et testé sur les données laboratoire et certains sur les données réelles, tels que les modèles affiné ou le modèle DANN. Les résultats de chaque modèle sont présentés dans les tableaux ci-dessous (voir la partie 6.1 en Annexes pour avoir un détail des métriques utilisées).

Les résultats obtenus sur les données de laboratoire, présent dans la table 1, nous montrent que le modèle AWL Resnet présente les meilleures performances. Cependant sur les données réelles (voir la table 2), les performances sont

Modèles	Acc Micro	Acc Macro	F1-score	Top 3
LP Resnet	89.6	88.4	89.3	97.3
AWL Resnet	97.1	95.7	96.5	99.8
FT LP Resnet	47.9	49.5	43.7	67.1
FT AWL Resnet	53	56.6	51.5	77.6
Adversarial	92.6	92.7	93.2	99.7
DANN	88.9	88.3	89.2	97.3
ADDA				
t-ReX				

Table 1: Résultats de test sur les données laboratoire

Modèles	Acc Micro	Acc Macro	F1-score	Top 3
LP Resnet	18	7	4	26.6
AWL Resnet	13.5	6.3	4.4	24.2
FT LP Resnet	32	11.7	10.1	45.1
FT AWL Resnet	32.4	11.3	8.4	57.4
Adversarial	13.1	7.1	6.8	30.7
DANN	32.7	12.7	12.5	42.6
ADDA				
t-ReX				

Table 2: Résultats de test sur les données réelles

globalement inférieures par rapport aux données laboratoire. Par ailleurs, le modèle DANN a obtenu de meilleurs résultats que la plus part des autres modèles, et nous pensons qu'avec une recherche plus approfondie des meilleures hyper-paramètres nous pourrions avoir de meilleures performances. *Les courbe de crossentropy sont disponible en Annexes, figure 3.*

Remarque : Nous n'avons pas pu reproduire exactement les résultats de l'étude précédente puisque l'équipe précédente a réalisé leurs tests avec un seul run par modèle à la place d'une moyenne de cinq run, qui apporterait moins de variations aléatoires. Par ailleurs, l'équipe n'a pas indiqué les seeds utilisés pour initialiser les poids des modèles et les générateurs aléatoires; il est donc impossible de garantir que les mêmes conditions de départ sont respectées.

L'interface : Parallèlement, M. Esperança nous a demandé d'améliorer l'interface déjà réalisée par le groupe précédent. Il souhaitait apporter trois améliorations majeures : qu'elle soit plus facile à prendre en main, qu'elle permette de corriger les prédictions, et de produire un rapport pour chaque session contenant pour chaque image : son nom, sa représentation, le label prédit ainsi que sa définition, et la carte de sillance. Nous avons réussi à intégrer la plupart de ces demandes avec le framework Streamlit [11] [12], tout en corrigeant des erreurs de la précédente implémentation, et en rendant l'interface plus esthétique et pratique.

4. SYNTHÈSE ET PERSPECTIVES

Notre projet d'identification automatique de traces de sang à l'aide de techniques de vision par ordinateur et de deep learning au sein de l'équipe Qarma, a pour but de fournir aux expert du LAC un outil de classification de traces de sang fiable et interprétable pour l'analyse de scène de crime. Les résultats préliminaires sont encourageants, mais soulignent également les défis posés par la généralisation aux données réelles.

La deuxième partie de notre stage, qui se déroulera jusqu'à mi-juillet, inclut alors plusieurs étapes clés. Tout d'abord, après avoir **amélioré des hyper-paramètres du modèle DANN**, nous continuerons à **implémenter la deuxième architecture ADDA** et à **développer le modèle t-ReX**, afin de déterminer si ces méthodes nous permettent d'obtenir de meilleurs résultats. D'autres part, puisque nos données réelles sont mal réparties, nous appliquerons des **pénalités aux classes sur représentées dans la loss**, pour obtenir un meilleur équilibre entre les classes. Nous effectuerons des benchmarks plus précis afin que les modèles soient davantage comparables. Par ailleurs, concernant les données nous effectuerons une phase d'**augmentation de données** avant la répartition des données en train, val, test. De même, une grande quantité d'images reste non étiquetées, et ces images pourront être utilisées dans les méthodes ADDA et DANN, étant donné que ces méthodes n'ont pas forcément besoin de données cibles étiquetées. En outre, toutes la partie amélioration de l'**interprétation des résultats** n'a pas pu être implémentée encore, nous nous concentrerons également sur cette tâche durant la seconde partie de notre stage, en étudiant les cartes de sillance avec Grad-CAM++ [13] et Score-CAM [14]. Enfin, nous continuerons à **améliorer l'interface** pour répondre plus précisément aux attentes supplémentaires du client.

Pour conclure, nous avons pu effectuer notre stage de Master 1 en tant que stagiaire en apprentissage automatique spécialisé dans le domaine de la vision par ordinateur. Durant ces un mois et demi, nous avons pu mettre en pratique nos connaissances théoriques dans ce domaine, acquises durant notre formation, tout en enrichissant notre savoir dans le domaine des réseaux de neurones profonds. Ce stage nous a permis de travailler sur des projets concrets, de développer notre compréhension des divers méthodes d'apprentissage automatique, de transfert de domaine et de collaborer avec des experts du (LIS). Nous avons également appris à produire des solutions innovantes pour des problèmes complexes. Enfin, malgré que toutes ces notions étaient nouvelles pour nous, et que nous avons pris du temps à les assimiler, cette expérience nous a permis d'enrichir notre parcours académique, mais également de renforcer notre motivation à poursuivre dans ce domaine.

5. REFERENCES

- [1] Philippe Esperança, “Atlas,” 2019.
- [2] C. Han, Y. Labeyrie, and A. Zabban, “Analyse de traces de sang,” Rapport Projet 3A, Mars 2024.
- [3] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, “Deep residual learning for image recognition,” *CoRR*, vol. abs/1512.03385, 2015.
- [4] Mei Wang and Weihong Deng, “Deep visual domain adaptation: A survey,” *CoRR*, vol. abs/1802.03601, 2018.
- [5] Yaroslav Ganin, Victor Lempitsky, Skolkovo Institute of Science, and Technology (Skoltech), “Unsupervised domain adaptation by backpropagation,” *CoRR*, vol. abs/1409.7495, 2014.
- [6] Eric Tzeng, Judy Hoffman, Kate Saenko, and Trevor Darrell, “Adversarial discriminative domain adaptation,” *CoRR*, vol. abs/1702.05464, 2017.
- [7] Qing Jiang, Feng Li, Tianhe Ren, Shilong Liu, Zhaoyang Zeng, Kent Yu, and Lei Zhang, “T-rex: Counting by visual prompting,” 2023.
- [8] TorchVision maintainers and contributors, “TorchVision: PyTorch’s Computer Vision library,” Nov. 2016.
- [9] James Bergstra and Yoshua Bengio, “Random search for hyper-parameter optimization,” *J. Mach. Learn. Res.*, vol. 13, no. null, pp. 281–305, feb 2012.
- [10] Mert Bulent Sariyildiz, Yannis Kalantidis, Karteek Alahari, and Diane Larlus, “No reason for no supervision: Improved generalization in supervised models,” in *International Conference on Learning Representations*, 2023.
- [11] “Streamlit: An open source python library for building web apps,” <https://streamlit.io/>, 2018.
- [12] “streamlit-option-menu: A simple streamlit component for menu selection,” <https://github.com/victoryhb/streamlit-option-menu>, 2022.
- [13] A. Chattopadhyay, A. Sarkar, P. Howlader, and V. N. Balasubramanian, “Grad-cam++ : Generalized gradient-based visual explanations for deep convolutional networks,” <http://arxiv.org/abs/1710.11063>, 2017.
- [14] H. Wang, M. Du, F. Yang, and Z. Zhang, “Scorecam : Improved visual explanations via score-weighted class activation mapping,” <http://arxiv.org/abs/1910.01279>, 2019.

6. ANNEXES

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

Fig. 1: Résultats de l'étude précédente des tests sur les données de laboratoire [2]

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

Fig. 2: Résultats de l'étude précédente des tests sur les données réelles [2]

Modèles	Epochs	Learning rate	Learning rate adversary	Lambda
LP Resnet	10	0.0005	-	-
AWL Resnet	20	0.0001	-	-
FT LP Resnet	20	0.0001	-	-
FT AWL Resnet	10	0.0001	-	-
Adversarial	20	0.005	0.0001	0.1
DANN	20	0.005	0.005	1

Table 3: Hyper-paramètres utilisés pour l'entraînement les différents modèles (trouvés par Grid Search, ou Random Search pour les modèles adversariaux)

Métriques Pour pouvoir comparer les modèles implémentés de manière rigoureuse, nous nous sommes concentré sur les 4 métriques suivantes :

L'Accuracy Micro : représente le pourcentage de réussite du modèle, le taux de bonne classification.

L'Accuracy Macro : représente la moyenne des pourcentages de réussite du modèle pour chaque classe.

La F1-score : représente la moyenne harmonique entre la précision (pourcentage de prédictions correctes parmi les prédictions positive), et le rappel (pourcentage de prédictions correctes parmi les vrais labels positifs).

Le score Top 3 : représente le pourcentage de bonnes prédictions parmi les 3 meilleures classes (c'est à dire celles qui prédisent le mieux). Cette métrique permet de répondre au problème de la très mauvaise répartition des données entre les classes, et qui affecte donc les résultats.

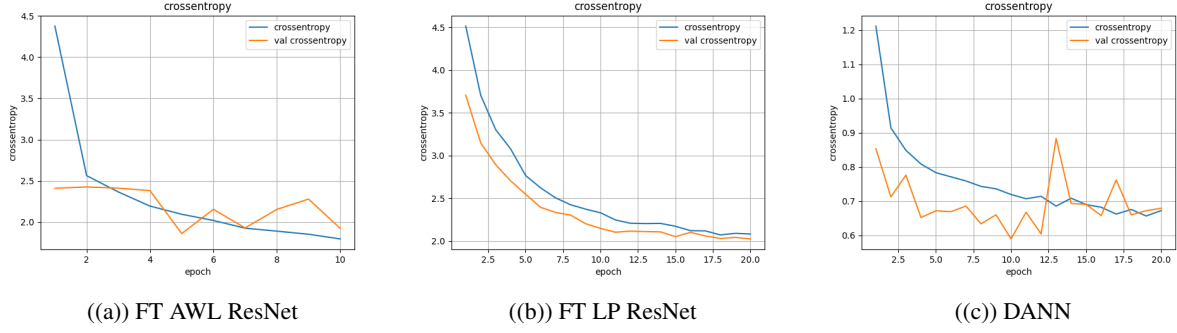


Fig. 3: Valeurs de la loss (en bleu) et de la validation (en orange) en fonction du nombre d'époch pour les trois meilleurs modèles entraînés

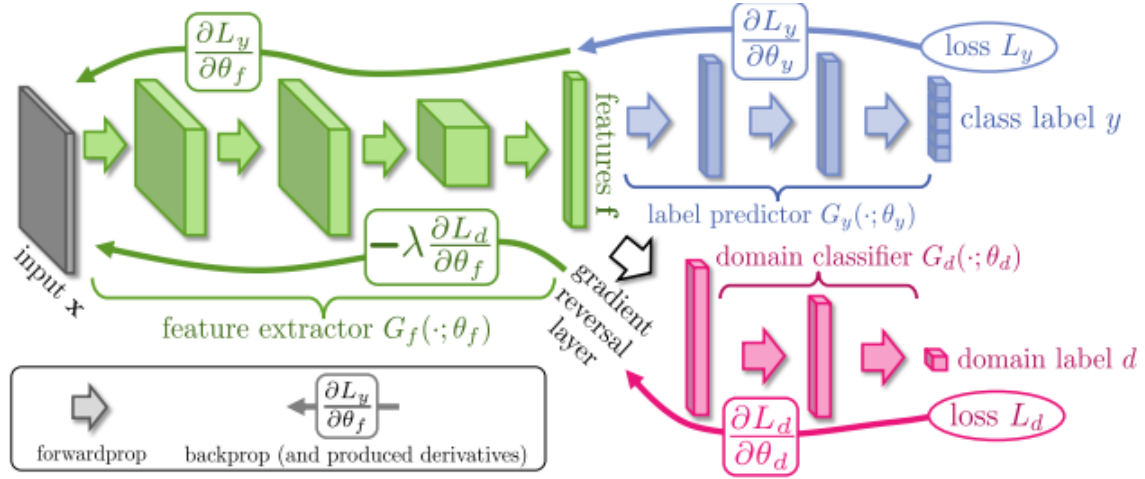


Fig. 4: The domain-adversarial neural network (DANN) architecture [5].

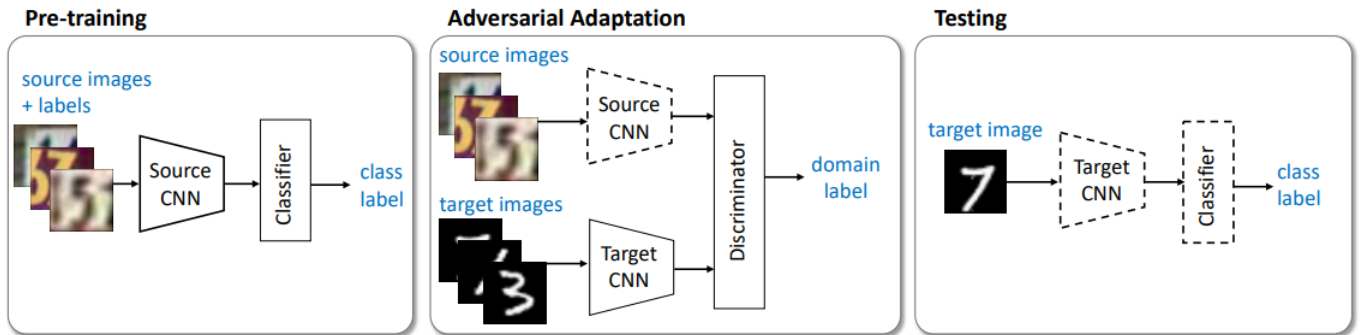
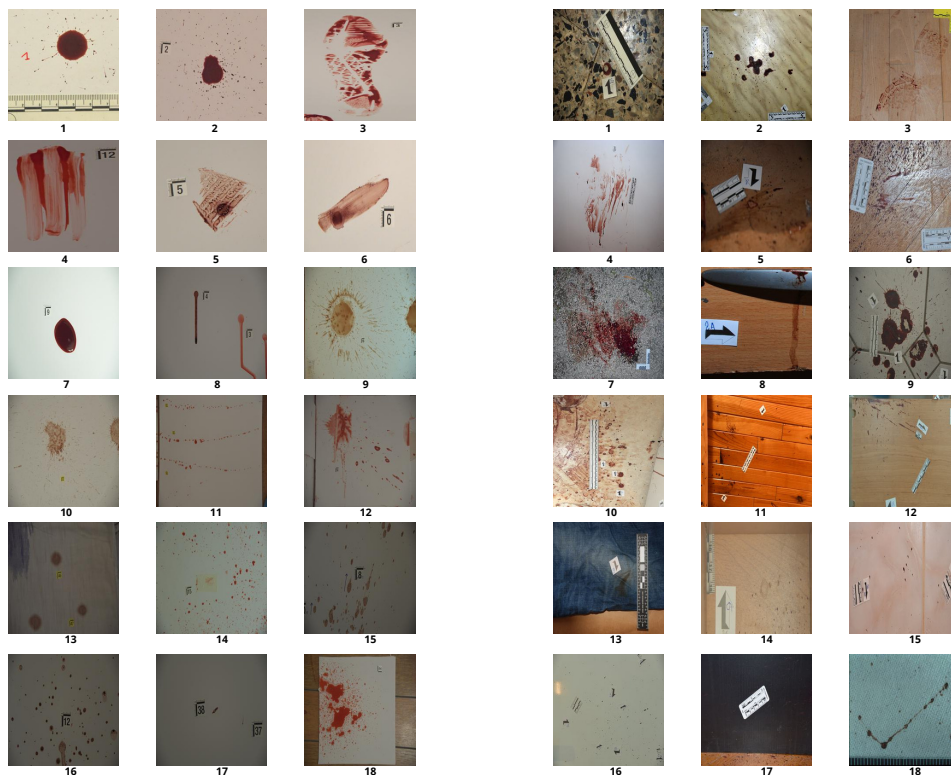


Fig. 5: The Adversarial discriminative domain adaptation (ADDA) architecture [6].



((a)) Photos laboratoires

((b)) Photos de cas réels

Fig. 6: Photos du jeu de données laboratoire et de cas réels pour chaque classes

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

Table 4: Tableau des 18 classes de traces de sang.

Classes	Nombre d'image du dataset labo.	Nombre d'image du dataset réelle n°1	Nombre d'image du dataset réelle n°2
1	793	27	86
2	403	5	8
3	389	30	101
4	358	20	57
5	399	2	3
6	302	3	10
7	304	4	5
8	158	4	13
9	409	10	30
10	1461	9	21
11	490	34	125
12	1164	5	15
13	61	25	37
14	198	8	13
15	635	26	238
16	398	18	82
17	98	8	52
18	144	6	12
TOTAL	8164	244	908

Table 5: Répartition des données

Projet GitHub

Vous trouverez ci-dessous un lien vers notre GitHub qui est pour le moment privé au vue de la sensibilité de nos modèles qui doivent pour le moment rester confidentiels : <https://github.com/NesrineGhannay/Blood-Stains-Classification.git>. A la fin du stage, ce dépôt git sera rendu accessible aux encadrant et aux prochaines personnes qui reprendront nos travaux.

Diagramme de Gantt Stage Blood Stains Classification

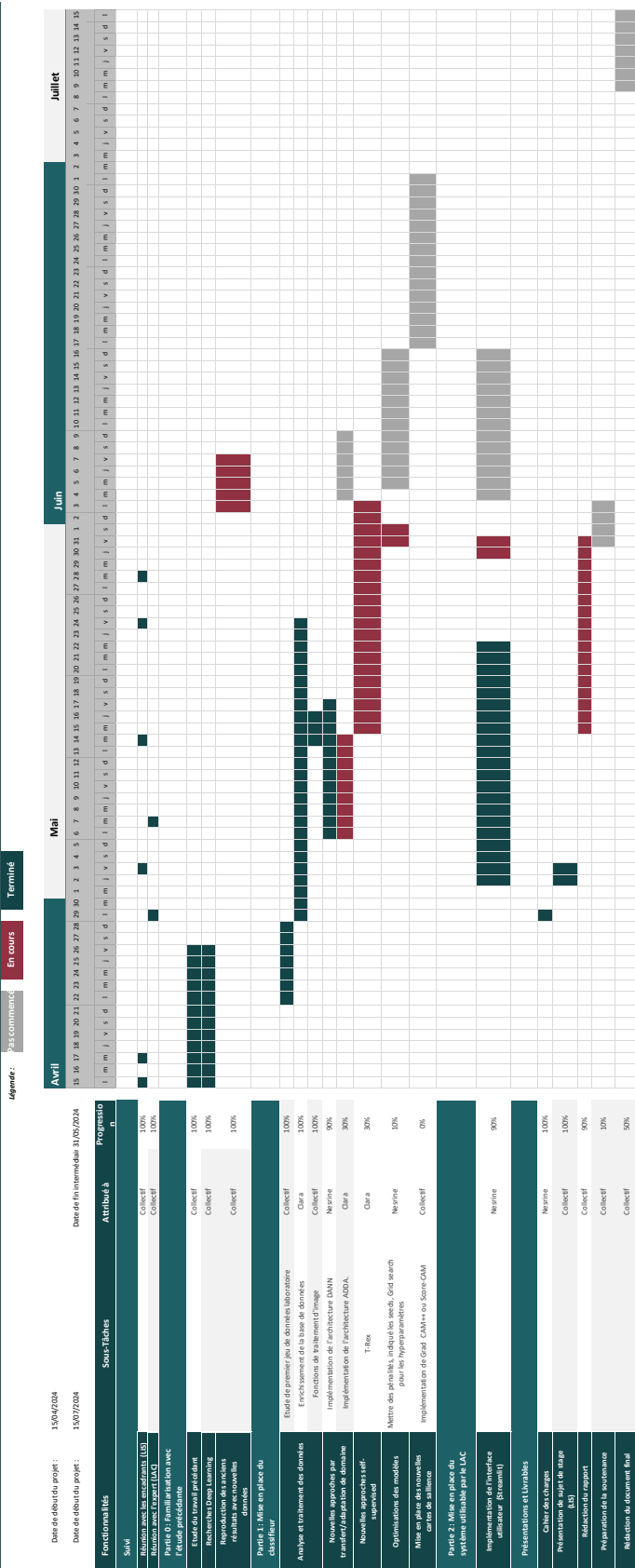


Fig. 7: Diagramme de Gantt