UNIVERSITÉ SAAD DAHLEB DE BLIDA

Faculté des sciences

Département d'informatique

MÉMOIRE DE MASTER

En Informatique

Option : Systèmes Informatiques et Réseaux

THÈME :

Solution basée sur l'apprentissage profond pour la modération de contenu vidéo

Réalisé par

KADDOUR Mohamed Ali

KESSI Ouassim

Encadré par

MANCER Yasmine

BOULEKNATER Sofiane

2024

Remerciements

Nous voudrions commencer par remercier Dieu Tout-Puissant, qui nous a donné la capacité d'accomplir notre travail de la meilleure façon, et nous sommes reconnaissants pour la force qu'il nous a donnée tout au long de cette période.

Nous exprimons nos reconnaissances à nos promoteurs, Madame Mancer Yasmine et Monsieur Bouleknater Sofiane, et nous les remercions d'avoir accepté de nous encadrer et de nous guider avec des conseils dans notre travail.

Nous remercions sincèrement nos chers parents et familles pour leur soutien continu, leurs encouragements et leur confiance en nous tout au long de notre travail.

Enfin, nous tenons à remercier les membres de notre jury d'avoir accepté d'évaluer et de juger notre travail.

ملخص

في الوقت الحاضر، مع ازدهار المنصات الرقمية وكذلك تطوير التلفزيون، أصبحت مراقبة محتوى الفيديو أمرًا بالغ الأهمية. استجابة للطلب المتزايد، تبث المنصات المزيد والمزيد من المحتوى، مما يتطلب مراقبة صحيحة لضمان تجربة ممتعة للمستخدمين.

يقترح هذا البحث عملية متعددة الوسائط باستخدام نماذج CNN المدربة مسبقًا مثل InceptionV3 و MobileNetV3-Large و DenseNet169 و VGGish للكشف عن العري والإباحة. لقد استخدمنا مجموعة البيانات LSPD وأنشأنا مجموعات بيانات إضافية للتدريب والتقييم. لقد حققت تجاربنا دقة بلغت 98,32 % على مجموعة البيانات NPDI-2K و 95,08 % بدمج التحليل البصري والصوتي.

الكلمات المفتاحية : مراقبة محتوى الفيديو ، Not Safe For Work ، التعلم العميق، تصنيف الفيديو، InceptionV3 ، MobileNetV3 ، DenseNet169 ، VGGish .

Résumé

De nos jours, avec l'essor des plateformes numériques ainsi que le développement de la télévision, la modération efficace du contenu vidéo est devenue cruciale. En réponse à la demande croissante, les plateformes diffusent de plus en plus de contenu, nécessitant une modération afin d'offrir une expérience agréable aux utilisateurs.
Ce mémoire propose une approche multimodale utilisant des modèles CNN pré-entraînés tels qu'InceptionV3, MobileNetV3-Large, DenseNet169 et VGGish pour la détection de la nudité et de la pornographie. Nous avons utilisé le dataset LSPD et créé des datasets supplémentaires pour l'entraînement et l'évaluation. Nos expériences ont atteint une Accuracy de 98,32% sur le dataset NPDI-2K et de 95,08% en combinant l'analyse visuelle et auditive.

Mots clés : modération de contenu vidéo, Not Safe For Work, apprentissage profond, classification vidéo, InceptionV3, MobileNetV3, DenseNet169, VGGish.

Abstract

Nowadays, effective video content moderation has become crucial with the rise of digital platforms and the development of television. In response to growing demand, platforms are broadcasting more and more content, necessitating moderation to provide a pleasant user experience.

This thesis proposes a multimodal approach using pre-trained CNN models such as InceptionV3, MobileNetV3-Large, DenseNet169, and VGGish for the detection of nudity and pornography. We used the LSPD dataset and created additional datasets for training and evaluation. Our experiments achieved an accuracy of 98.32% on the NPDI-2K dataset and 95.08% by combining visual and auditory analysis.

Keywords : video content moderation, Not Safe For Work, deep learning, video classification, InceptionV3, MobileNetV3, DenseNet169, VGGish.

Table des matières

Table des figures

Liste des tableaux

Liste des abréviations

ARAV Autorité de régulation de l'audiovisuel

CNN Convolutional Neural Network

EPTV Établissement Public de Télévision

FFNN Feed Forward Neural Network

LSTM Long Short-Term Memory

MFCC Mel-Frequency Cepstral Coefficients

NSFW Not Safe For Work

ReLU Rectificateur Linéaire Unitaire
RF Random Forest
RNN Recurrent Neural Network
RVB Rouge Vert Bleu

SFW Safe For Work
SOD Sensitive Object Detection
SVM Support Vectors Machine

ViSiL Video Similarity Learning

XP Extreme Programming

YOLO You Only Look Once

Introduction générale

Ces dernières années ont connu un développement remarquable concernant les plates-formes de diffusion de média en tout genre (audio, photo et vidéo), cet essor implique l'augmentation du nombre de médias comportant du contenu inapproprié. L'abondance ainsi que la diversité de ce contenu ont fait surgir un nouveau défi, celui de modérer ce contenu efficacement.

L'Établissement Public de Télévision (EPTV) se doit aussi de modérer son contenu afin de maintenir un visionnage respectueux et en harmonie avec les normes de la société. Ce défi déjà pas simple se complique encore plus avec la diversité et la quantité de média mis en ligne chaque jour.
En effet, l'EPTV compte neuf chaînes de télévision, qui diffusent du contenu en tout genre, que ce soit en direct ou bien en différé. Une erreur peut nuire à la réputation d'une société, provoquer des pertes et même des poursuites judiciaires. Contrôler ce contenu quelles que soient sa taille et la façon de procéder est l'une des responsabilités de l'EPTV, et est une tâche très importante.

En 2022, " L'Autorité de régulation de l'audiovisuel (ARAV) a décidé de fermer définitivement la chaîne TV 'Al Adjwaa' "[11], pour cause, la diffusion " de scènes offensantes et contraires aux valeurs de notre société et à notre religion " [11].

En 2023, " L'ARAV a décidé de suspendre 'tous les programmes' de la chaîne Essalam TV pour une durée de vingt (20) jours " [5], la cause est la diffusion de " contenus et de scènes contraires aux préceptes de l'islam et aux mœurs de la société algérienne " [5]. L'importance de la modération de contenu est primordiale, et une erreur à ce niveau peut causer des pertes ainsi que des problèmes juridiques et moraux.

Dans ce mémoire, nous emploierons le terme Not Safe For Work (NSFW) [1] pour parler de tout contenu offensant et non approprié, ce tag est très utilisé sur internet, et est mis pour mettre en garde les internautes de la présence d'un contenu inapproprié, le contenu NSFW englobe en général : de la nudité, de la pornographie, de la violence, du gore, des grossièretés, des discours de haine ou toute autre idéologie extrémiste. Le terme NSFW est défini en détails dans I.2.2.

1. Littéralement pas sûr pour le travail

Problématique

L'EPTV avec ses neuf chaînes et sa grande quantité de données ne peut plus se permettre de modérer son contenu manuellement, automatiser cette modération de façon efficace devient essentiel. L'automatisation de ce processus n'est pas possible, un modèle d'intelligence artificiel quelle que soit sa performance n'est jamais fiable à 100 %, sur tout en sachant les conséquences qu'une seule erreur peut avoir. Mais grâce à un modèle efficace, le travail manuel sera amoindri, et des économies de temps et d'argent peuvent être fait.

Le manque de solutions adaptées à nos standards et aux cultures et traditions locales est flagrant, développer une solution qui répond aux besoins des algériens, et des musulmans de façon générale est important. De ce fait, la définition du contenu NSFW est tout aussi importante que sa détection.
Dans ce mémoire, nous nous concentrons uniquement sur la détection de la nudité et la pornographie.

Objectifs du travail

Notre objectif est de trouver une solution basée sur l'apprentissage profond capable de détecter du contenu NSFW dans des vidéos, plus précisément, la détection de la nudité et de la pornographie.
Cet outil vise du moins à soulager et à faciliter le travail manuel de modération de l'EPTV.
Nos objectifs sont :

- Définir le contenu NSFW selon les traditions et la culture algérienne.
- Collecter les bases de données nécessaires.
- Concevoir un modèle permettant de détecter la nudité et la pornographie dans les vidéos.
- Entraîner ce modèle et l'évaluer.

Organisation du mémoire

Le présent mémoire est construit comme suit :
Chapitre I : Nous avons défini la modération de contenu. Nous avons aussi défini le contenu NSFW et présenté un état d'art où nous avons analysé les méthodes utilisées dans la littérature.
Chapitre II : Nous avons présenté la conception de notre outil en s'aidant de diagrammes et tableaux afin d'expliquer et d'illustrer le fonctionnement de notre solution.
Chapitre III : Nous avons montré et expliqué notre implémentation, l'environnement logiciel et le matériel utilisé, nous avons aussi présenté les résultats des différentes expériences tentées ainsi que notre interprétation des résultats.
Nous avons terminé avec une conclusion et des perspectives d'avenir.

Chapitre I

État de l'art

I.1 Introduction

Dans ce chapitre, nous allons définir ce qu'est la modération de contenu ainsi que le contenu à modérer.
Nous ferons aussi une revue de la littérature dans laquelle nous mentionnerons les anciennes techniques et leurs limitations, nous parlerons aussi des dernières méthodes et nous détaillerons leurs fonctionnements et performances tout en faisant des comparaisons. Nous parlerons aussi des datasets utilisés et disponibles.

I.2 Définitions

I.2.1 La modération de contenu vidéo

La modération du contenu, quelle que soit sa forme, est le processus de contrôle afin d'assurer que le contenu n'est pas "offensant, nuisible, trompeur, illégal ou inapproprié de n'importe quelle manière" [27].
Ce processus peut-être "automatisé, exécuté manuellement ou en utilisant une approche hybride qui inclut des humains dans la boucle" [27].

La modération de contenu vidéo est l'application de ce processus de contrôle sur un contenu vidéo, ce contenu peut provenir et être distribué sur divers supports (télévision, plateforme de streaming, film, réseaux sociaux).
Nous pouvons appeler le contenu jugé comme approprié : Safe For Work (SFW) [1], et le contenu jugé comme inapproprié : Not Safe For Work [2]. Ci-dessous sa définition.

1. Littéralement : sûr pour le travail.
2. Littéralement : pas sûr pour le travail.

I.2.2 Le contenu Not Safe For Work

Malgré l'absence de connaissance de l'origine exacte de cette abréviation (NSFW), elle est utilisée, sur internet principalement pour faire en sorte d'avoir une expérience sans contenu inapproprié, ce contenu englobe en général : de la nudité, de la pornographie, de la violence, du gore, des grossièretés, des discours de haine ou toute autre idéologie extrémiste.
Le dictionnaire d'Oxford définit l'abréviation NSFW comme : " un lien qui pointe vers un site ou une page web qui contient des images, du texte ou des vidéos que les gens peuvent considérer comme offensants " [12].

Bien que certaines parties de la modération de contenu soient universelles, il existe toujours des spécificités pour chaque région ou culture, il est clair que le contenu que des algériens musulmans considèrent comme NSFW peut être très différents que celui des occidentaux.
En absence d'étude sur ce que les algériens considèrent comme NSFW, et en prenant en compte le degré de conservation et de religion de notre peuple, nous avons défini le contenu NSFW algérien comme suit : nudité, pornographie, violence extrême, excès de sang, insultes, surexposition de la peau (personnes en maillot de bain court ou en sous-vêtement), exposition de forme ou attribut à notation sexuelle, scène de baiser.

Il existe une autorité algérienne nommée ARAV qui a pour fonction de réguler le contenu mis en ligne sur la télévision algérienne, connaître leur version du contenu NSFW est intéressant et plus percutant.

Nous allons nous concentrer dans ce mémoire uniquement sur la détection de la nudité et de la pornographie.

I.3 Revue de la littérature

Au début, le contenu NSFW était contrôlé manuellement, avec la présence d'employés dédiés à ce travail. Après le développement, de nouvelles capacités ont été découvertes pour effectuer ce travail automatiquement et offrir un confort aux travailleurs.
Aujourd'hui, l'approche de l'apprentissage profond est la plus utilisée et la plus performante, notamment grâce à l'utilisation de modèles pré-entrainés, mais il existait d'autres approches, parmi elles :

- Algorithmes de détection de peau, cette approche est assez limitée et peu performantes.
- Sac de mots, c'est-à-dire la conversion du contenu en mots pour ensuite faire une classification, cette approche est très couteuse du point de vue du temps d'exécution.
- Modèle basé sur les mouvements, bien que cette approche soit plus performante que les deux précédentes, elle peine à détecter du contenu NSFW statique.

En vue du dépassement des autres approches, nous nous sommes concentrés seulement sur les techniques utilisant de l'apprentissage profond.

Une vidéo est une combinaison d'une séquence d'image et d'un son, ça nous offre plusieurs types de média que nous pouvons travailler avec.

Par exemple certaines études ont utilisé seulement les images, cette technique a pour avantage d'être plus flexible, car elle marche pour les vidéos et les images et est moins gourmande par rapport à d'autres techniques.
Pour utiliser cette technique sur une vidéo, nous prenons chaque image de la vidéo et nous la traitons séparément, nous obtiendrons un score de NSFW pour chaque image par exemple, et nous ferons une moyenne sur l'ensemble des images de la vidéo afin de la classifier, ou nous choisissons que le fait qu'une seule image contienne un contenu NSFW suffit à classifier toute la séquence en tant que contenu NSFW, d'autres techniques existe utilisant un seuil limite permettant de classifier la vidéo. Le lissage des résultats permet d'avoir un résultat final plus lisible et ayant une meilleur continuité.
Afin de réduire le coût de calcul, nous pouvons décider que plutôt de traiter l'ensemble des images de la vidéo, nous traiterons chaque x images.

Nous pouvons utiliser la vidéo en tant que séquence d'image, par rapport à la technique précédente cette technique prend en considération la dimension temporelle, c'est-à-dire que l'ordre des images a une importance, ça permet au modèle de mieux percevoir la vidéo et de mieux comprendre les actions réaliser.

Nous pouvons aussi utiliser le son, c'est une technique plus complexe, car de façon générale le son a trop de bruit (musique de fond, son d'effet spéciaux) qui peut fausser le résultat.

Notons que le contenu NSFW peut être seulement sur le son ou seulement sur l'image, donc en utilisant seulement une seule source, nous pouvons passer à côté. Bien que dans le cas d'une vidéo, le son et l'image sont souvent synchronisés, il se peut que pour des raisons techniques ou artistiques, il ne le soit pas.

I.3.1 Détection par son

Comme le tableau I.1 l'indique, dans l'étude [26], les auteurs ont utilisé un modèle Convolutional Neural Network (CNN) et Feed Forward Neural Network (FFNN) entraîner sur le Spectre Log Mel et Mel-Frequency Cepstral Coefficients (MFCC) afin de détecter des sons pornographiques, ces solutions sont la norme pour la détection de parole, mais leurs applications sont une nouveauté pour la détection de pornographie.
Dans cette étude, les meilleures performances ont été obtenues en utilisant un modèle CNN entraîné sur le Spectre Log Mel à une taille de segments de 60 secondes avec une classification binaire soit il y a une pornographie ou non, le score F1 est de : 94,89 %, l'entraînement et les tests ont été réalisés sur le dataset NPDI-800 [7] (400 vidéos pornographiques / 400 vidéos non pornographiques).

Dans l'étude [36], les auteurs se sont concentré sur la détection de son acoustique pornographique, ils ont utilisé la technique Log Filter Banks pour extraire les caractéristiques qui ont servi

TABLE I.1: Comparaison des études sur la détection par son

Étude	Année	Technique	Classification	Précision	Dataset
[36]	2022	ResNet18 + Log Filter Banks	Porno/Non-porno	97,19% sur dataset privé	Dataset privé
[26]	2022	CNN entraîné sur un spectre Log Mel	Porno/Non-porno	94,89% sur NPDI-800	NPDI-800[7]

d'entrée pour un modèle CNN pré-entrainé ResNet18, après ça nous retrouvons des couches de classification.
Un dataset a été spécialement conçu pour cette étude, il est composé de 224127 sons pornographiques et 274206 sons normaux.
Les auteurs ont réussi à atteindre une précision de : 97,19 % avec tous les ajustements et améliorations combinés.

Nous avons remarqué un manque d'études concluantes sur la détection par son uniquement, cela est dû à la démocratisation de la vision par ordinateur, et un certain retard sur les techniques de traitement du signal audio.

I.3.2 Détection par image

Le tableau I.2 résume les techniques, datasets et résultats obtenus par ces études, à noter que sauf si le contraire est indiqué, la précision des modèles a été réalisée sur le dataset NPDI-2K.

L'étude [2] est la plus récente étude qui a fait un état de l'art, et parmi les rares études ayant fait des tests sur le dataset LSPD [14], ils ont ré-entraîné des modèles CNN conçus pour des tâches différentes et ont conclu que ConvNexT(tiny) entraîner sur LSPD a les meilleurs résultats sur l'ensemble des datasets, Inceptionv3 et MobileNetv3 (large) ont obtenu des résultats très similaires sur le dataset LSPD.
ConvNexT(tiny) a obtenu 94,90 % sur le dataset LSPD avec une classification sur cinq classes.

Dans l'étude [32], les auteurs ont utilisé une combinaison de modèle pré-entraîner pour l'extraction de caractéristique de bas et moyen niveau (MobileNet-V2 et DenseNet-169 est la meilleure combinaison), les caractéristiques extraites de ces modèles, ont été fusionnés et entraînés à nouveau sur le dataset afin de classifier les images, la classification ce fait grâce à un classificateur sigmoïde (une classification binaire soit : image obscène ou image non obscène). Le dataset proposé [32] contient 23000 photos obtenue grâce à une intelligence artificielle générative de photo par photo Pix-2-Pix Gan, deux autres datasets ont été utilisés pour les benchmarks (NPDI-800 [7] et NPDI-2K [29]). Le score F1 obtenu est 98,48 %.

Dans l'étude [23], les auteurs ont utilisé un modèle CNN au sein d'une machine de Boltzmann restreinte de type Gaussien-Bernoulli avec une classification binaire (contenu adulte ou non adulte), la précision de cette approche a atteint : 99,10 %, le dataset utilisé est NPDI-2K [29].

Dans l'étude [34], les auteurs ont utilisé You Only Look Once (YOLO)v3 et l'ont entraîné à

nouveau sur les datasets GVIS SOD [3] et GVIS APD-2M [4] où les organes et objets sexuels ont été annotés (neuf classes), les auteurs ont choisi de détecter les organes ou objets et les classifiés en tant que normal ou bien sexuel. Les auteurs ont aussi créé un modèle qui classifie 19 classes (plus de précision sur les actes et nombre de personnes présentes), on appelle ce type de modèle : modèle Sensitive Object Detection (SOD).
Les auteurs ont obtenu une précision de : 91,50 % sur le dataset NPDI-800 [7].

Dans l'étude [4], les auteurs ont utilisé YOLO pour détecter les humains dans la photo, pour ensuite les passer à un modèle CNN pré-entraîné sans les dernières couche ce qui permet d'extraire les caractéristiques, ils ont ensuite utilisé un classificateur afin de classifier les images en tant que normal ou contenant de la nudité (classification binaire).
Les meilleurs résultats dans cette étude ont été obtenus en utilisant YOLO + ResNet-101 + Random Forest (RF) pour la classification, les auteurs ont obtenu un score F1 de : 90,03 %, le dataset utilisé est NPDI-2K [29] augmenté, les limitations de cette méthode sont que si YOLO fait une erreur, il fausse complètement le résultat (par exemple YOLO ne détecte pas d'humain alors que dans la photo, il y a un humain nu, cette image sera classifiée en tant qu'une image normale).

Dans l'étude [9], les auteurs ont créé une combinaison de trois modèles qui permette d'extraire les caractéristiques d'une image, détecter les zones sensibles, extraire les caractéristiques de ces zones et les classifiés en inoffensive ou sexy ou bien pornographique, c'est-à-dire la classification ce fait sur trois classes. Les auteurs ont obtenu une précision de : 92,70 % sur le dataset NPDI-2K [29]. Les datasets utilisé sont NPDI-2K [29] et AIC Dataset[35].

Dans l'étude [14], les auteurs ont créé un dataset nommé LSPD, nous y reparlerons plus tard avec plus de détails dans la sous-section I.3.5.
Les auteurs ont aussi créé un modèle permettant de détecter les objets sexuels (quatre classes), mais ils ont aussi testé le modèle créé par l'étude [30] [5], il a obtenu une précision de 87,22 % sur la classification binaire sur images (pornographique ou non), le modèle a été testé sur les cinq classes du dataset séparément (sexy, hentai [6], pornographie, normal, dessin) et a obtenu en moyenne une précision de : 79,02 % avec la meilleure précision sur la classe pornographie qui a atteint une précision de : 91,27 %, ce modèle n'a pas eu de test sur la classification binaire pour les vidéos.

I.3.3 Détection par vidéo

Le tableau I.3 résume les techniques, datasets et résultats obtenus par ces études, à noter que sauf si le contraire est indiqué, la précision des modèles a été réalisé sur le dataset NPDI-2K.

3. `https://gvis.unileon.es/datasets-sod/`, consulté le 31/05/2024
4. `https://gvis.unileon.es/datasets-apd-2m/`, consulté le 31/05/2024
5. Il a obtenu une précision de 90,40 % sur le dataset NPDI [29]
6. Manga/dessin animé contenant de la pornographie ou du contenu explicite

TABLE I.2: Comparaison des études sur la détection par image

Étude	Année	Technique	classification	Précision	Dataset
[2]	2023	ConvNexT(tiny) ré-entraîné	porno/ normal/ sexy/ dessin/ hentai	94,90% sur LSPD	LSPD
[32]	2023	Mobile-Net V2 + DenseNet 169	Porno/Non-porno	99,15%	NPDI-800 et NPDI-2K et GGOI
[14]	2022	Modèle CNN [30] entraîné sur un autre dataset	Porno/Non-porno	79,02% sur LSPD	LSPD
[23]	2022	CNN au sein d'une machine de Boltzmann restreinte de type Gaussien-Bernoulli	Porno/Non-porno	99,10%	NPDI-2K et des images d'internet
[34]	2021	YOLOv3 ré-entraîné	9 classes d'organes/objets sexuels	91,50% sur NPDI-800	GVIS SOD et GVIS APD-2M
[4]	2020	YOLO3 + ResNet-101 + RF	Porno/Non-porno	87,75%	NPDI-2K(augmenté)
[9]	2019	3 modèles CNN combiné	Porno/ Sexy/ Normal	92,70%	NPDI-2K et AIC Dataset

Dans l'étude [18], les auteurs ont utilisé la vidéo en tant que séquence d'images, le son n'a pas été utilisé, les auteurs ont utilisé un modèle pré-entraîné pour l'extraction de caractéristiques ResNet-18 couplé à ConvNet (modèle CNN qui permet de travailler sur des séquences d'images) avec une classification binaire soit la vidéo est pornographique ou non.
Cette étude a obtenu une précision de 97,70 %, les datasets utilisé sont [7] et [29].

Dans l'étude [19], les auteurs ont utilisé la vidéo en tant que séquence d'images, le son n'a pas été utilisé, les auteurs ont utilisé un modèle CNN à deux flux avec une classification binaire, soit la vidéo est pornographique ou non, l'un contient les images clés, l'autres contient le flux optique.
Le score F1 obtenu est 95,10 %, le dataset utilisé est NPDI-2K [29], cette méthode, c'est montré assez efficace avec les vidéos non pornographiques difficiles (par exemple boxe, lutte et allaitement).

Dans l'étude [8], les auteurs ont utilisé la vidéo en tant que séquence d'images, le son n'a pas été utilisé. Les auteurs ont utilisé le modèle CNN pré-entraîné (MobileNetV2) en plus d'un modèle Recurrent Neural Network (RNN) pour classifier les vidéos au tant que vidéo pornographique ou non, et puis ils ont segmenter les vidéos pornographiques en une séquence d'image et isoler les images ayant de la nudité, ensuite, ils ont utiliser un modèle SOD pour détecter les organes sexuels sur les images qui comporte de la nudité, à la fin, ils ont utiliser une fonction de sévérité des séquences pour le classement de la gravité de la vidéo.
Le résultat obtenu montre que l'utilisation de vidéo à des meilleurs résultats que les images, la précision pour la détection de vidéo est de : 97,80 %, les datasets utilisé sont : NPDI-800 [7], le

dataset APD-VIDEO [7] a aussi été créé, il contient le dataset MPII [8] [6] en plus du dataset Kaggle Thumbzilla.
Cette étude a aussi remarqué la présence de logo sur 60 % des vidéos pornographiques et 16 % sur les vidéos normals, les auteurs ont pensé que la présence de ces logos peut fausser les résultats et donner au modèle une sorte de raccourci, ce n'était pas le cas, après des tests, ils ont remarqué une légère différence si on masquait les logos (-0.81 % sur la détection par vidéo et -2 % pour la détection par image), ces résultats renforcent l'observation de la robustesse du modèle pour la détection de vidéo par rapport aux images.

Dans l'étude [13], les auteurs ont utilisé la vidéo en tant que séquence d'images, le son n'a pas été utilisé. Les auteurs ont utilisé une combinaison de modèle CNN pré-entraîné (ResNet101 + DenseNet121) et une approche permettant de calculer la similitude inter-feature et intra-feature. Les auteurs ont divisé ce travail en deux branches, la branche intra-feature fait en sorte que les vidéos sont divisées en images, ces images sont regroupées selon leurs similitudes par des algorithmes de clustering non supervisés afin de garantir un temps de calcul optimal, la branche inter-feature utilise le modèle Video Similarity Learning (ViSiL) pour calculer les relations spatio-temporelles entre deux vidéos en comparant leurs similarités de cadre.
Finalement, les auteurs ont combiné les résultats de ces deux branches et leur ont appliqué une classification Support Vectors Machine (SVM) binaire.
Les auteurs ont obtenu une précision de 96,88 %. Le dataset utilisé est NPDI-2K [29].

Dans l'étude [16], les auteurs ont utilisé la vidéo en tant que séquence d'image et de son, les auteurs ont utilisé une combinaison de deux modèles CNN pré-entraîné (InceptionV3 pour extraire les caractéristiques visuelles des images et AudioVGG pour extraire les caractéristiques audio), pour la classification les auteurs ont utilisé un classificateur séquentiel Long Short-Term Memory (LSTM) afin de les classifier soit en tant que vidéos pornographiques ou non pornographiques.
Le score F1 obtenu est : 94,00 %, les datasets utilisé sont NPDI-2K [29] et XVideos [9] et des vidéos éducative brésilienne [10], nous avons découvert que le classificateur séquentiel LSTM a obtenu de bien meilleur résultat que d'autre classificateur non séquentiel.

Dans l'étude [33], les auteurs ont utilisé la vidéo en tant que séquence d'image et de son, les auteurs ont utilisé trois classificateurs (classificateur visuel + classificateur audio + combinaison des deux) afin de réduire les erreurs.
Ils ont utilisé VGG-16 + RNN bidirectionnelle(LSTM) pour extraire les caractéristiques visuelles, un CNN avec Mel-Scaled Spectogram pour extraire les caractéristiques audio, le dernier classificateur est une combinaison des deux, la classification est binaire (pornographique/non pornographique), la précision est : 92,33 %, le dataset utilisé est NPDI-2K [29], la performance du modèle audio est à améliorer.

7. `https://gvis.unileon.es/datasets-apd-video/`, consulté le 31/05/2024
8. `http://human-pose.mpi-inf.mpg.de/`, consulté le 31/05/2024
9. `https://info.xvideos.net/db`, non disponible en Algérie
10. `https://video.rnp.br/`, consulté le 31/05/2024

TABLE I.3: Comparaison des études sur la détection par vidéo

Étude	Année	Détection de	Technique	classification	Précision	Dataset
[18]	2023	Séquence d'images	CNN + ResNet-18	Porno/Non-porno	97,15%	NPDI-800 et NPDI-2k
[19]	2023	Séquence d'images	CNN à deux flux	Porno/Non-porno	95,20%	NPDI-2k
[8]	2022	Séquence d'images	MobileNetV2 + RNN	Porno/Non-porno	97,80% sur NPDI-800	NPDI-800 et APD-VIDEO
[13]	2022	Séquence d'images	ResNet101 + DenseNet121 et calcule de similitude (CNN + ViSiL)	Porno/Non-porno	96,88%	NPDI-2k
[16]	2020	Séquence d'images et son	AudioVGG et InceptionV3 + LSTM	Porno/Non-porno	94,00%	NPDI-2k et XVideos et RNP
[33]	2020	Séquence d'images et son	VGG-13 + LSTM + CNN sur l'échelle de Mel	Porno/Non-porno	92,33%	NPDI-2k

I.3.4 Discussion

L'étude [2] est la plus récente étude ayant fait un état de l'art complet, les auteurs comparent les résultats des différents modèles sur plusieurs datasets (LSPD [14] [11], NudeNet [12], Adult-Content [10]). Les résultats montrent que ConvNexT(tiny) qui a était entraîné sur le dataset LSPD[14] est le plus polyvalent, il a obtenu les meilleurs résultats sur les trois dataset avec une moyenne de score F1 de : 94,3 %, MobileNetv3 (large) et Inceptionv3 ont aussi obtenu de très bons résultats aussi, les auteurs soulignent la différence entre la catégorie sexy des datasets, en effet LSPD[14] et NudeNet[12] ont tous les deux une catégorie sexy, mais elles sont différentes.
Ceci est un lien [13] important qui comporte plusieurs sources vers des modèles et des datasets pour la modération de contenu [1], nous y retrouvons un résumé à jour sur les différentes techniques, modèles et datasets concerné.

Les techniques possibles sont soit de créer un modèle (CNN de façon générale) et l'entraîné, ou bien d'utiliser un modèle pré-entraîné en lui ôtant les dernières couches et en les ré-entraînant sur notre dataset. Nous pouvons bien évidement combiner plusieurs modèles pour accroître la précision ou pour combiner les caractéristiques du son et des images.
Si nous souhaitons détecter certains objets comme des cigarettes ou de l'alcool, c'est possible grâce à des modèles déjà existant comme YOLOv3, comme ça était fait par l'étude [28] mais

11. Images seulement
12. `https://archive.org/details/NudeNet_classifier_dataset_v1`, consulté le 31/05/2024
13. `https://github.com/fcakyon/content-moderation-deep-learning`

TABLE I.4: Comparaison des études

Étude	Année	Détection par	Technique	Classification	Précision
[32]	2023	image	Mobile-Net V2 + DenseNet 169	Porno/Non-porno	99,15%
[23]	2022	image	CNN au sein d'une machine de Boltzmann restreinte de type Gaussien-Bernoulli	Porno/Non-porno	99,10%
[8]	2022	vidéo	MobileNetV2 + RNN	Porno/Non-porno	97,80% sur NPDI-800
[36]	2022	son	ResNet18 + Log Filter Banks	Porno/Non-porno	97,19 % sur dataset privé
[18]	2023	vidéo	CNN + ResNet-18	Porno/Non-porno	97,15%
[13]	2022	vidéo	ResNet101 + DenseNet121 et calcul de similitude (CNN + ViSiL)	Porno/Non-porno	96,88%
[19]	2023	vidéo	CNN a deux flux	Porno/Non-porno	95,20%
[2]	2023	image	ConvNexT(tiny) ré-entraîné	porno/normal/ sexy/dessin/ hentai	94,90% sur LSPD
[26]	2022	son	CNN entraîné sur un spectre Log Mel	Porno/Non-porno	94.89% sur NPDI-800
[16]	2020	vidéo (images+son)	AudioVGG et InceptionV3 + LSTM	Porno/Non-porno	94,00%
[9]	2019	image	3 modèle CNN combiné	Porno/Sexy /Normal	92,70%
[33]	2020	vidéo (images+son)	VGG-13 + LSTM + CNN sur l'échelle de Mel	Porno/Non-porno	92,33%
[34]	2021	image	YOLOv3 ré-entraîné	9 classes d'organes/objets sexuels	91,50% sur NPDI-800
[4]	2020	image	YOLO3 + ResNet-101 + RF	Porno/Non-porno	87,75%
[14]	2022	image	CNN entraîné sur un autre dataset	Porno/non-porno	79,02% sur LSPD

TABLE I.5: Comparaison des études avec classification binaire et testé sur le dataset NPDI-2K

Étude	Année	Détection par	Technique	Précision
[32]	2023	image	Mobile-Net V2 + DenseNet 169	99,15%
[23]	2022	image	CNN au sein d'une machine de Boltzmann restreinte de type Gaussien-Bernoulli	99,10%
[18]	2023	vidéo	CNN + ResNet-18	97,15%
[13]	2022	vidéo	ResNet101 + DenseNet121 et calcul de similitude (CNN + ViSiL)	96,88%
[19]	2023	vidéo	CNN a deux flux	95,20%
[16]	2020	vidéo (images+son)	AudioVGG et InceptionV3 + LSTM	94,00%
[33]	2020	vidéo (images+son)	VGG-13 + LSTM + CNN sur l'échelle de Mel	92,33%
[4]	2020	image	YOLO3 + ResNet-101 + RF	87,75%

avec un résultat de 85 %.
Une technique qui permet de détecter la peau à travers les pixels existe, cette méthode n'est pas très efficace, son fonctionnement est simple, si le pourcentage de peau dans l'image est élevé alors l'image est classifiée en tant que nudité, si une personne est nue, mais prend peu de place dans l'image (exemple une personne éloigné) l'image ne sera pas classé en tant que contenant de la nudité, l'étude [17] a utilisé cette technique et a obtenu une précision de : 80 %.

Notons que la détection par image de [32] a obtenu le meilleur résultat avec une précision de : 99,15 %, nous remarquons aussi que les études ayant obtenu les meilleures performances ont utilisés et combinés des modèles pré-entraînés.
Nous remarquons aussi que l'augmentation des images grâce à des intelligences artificielles génératives peut être bénéfique afin d'agrandir et diversifier les données.

La diversité des approches n'a pas affecté leurs performances de façon drastiques, mais il faut souligner qu'aucune des études n'a était faite dans le contexte de la modération de contenu pouvant passer à la télévision. En dépit de la performance des modèles, rien ne garantit ou ne peut prédire leurs résultats dans des situations réelles.
Par conséquent, l'utilisation d'un modèle combinant le son et l'image est plus intéressant, cette approche permet de minimiser le risque de rater un contenu NSFW sur un média non pris en charge (à cause de la possibilité de non-synchronisation de l'image et du son).
De plus, l'utilisation de la dimension temporelle devrait en théorie aider le modèle à mieux comprendre l'action effectué, en pratique les modèles ayant cette dimension ont obtenu de bons résultats, mais pas les meilleurs, cette approche reste très intéressante et prometteuse.

Le tableau I.4 est un résumé comportant toutes les études citées ci-dessus, le tableau I.5 présente les études qui ont été testés sur le même dataset, en l'occurrence NPDI-2K, le tableau I.6 présente les deux études qui ont utilisé le dataset LSPD.

TABLE I.6: Comparaison des études utilisant le dataset LSPD

Étude	Année	Classification	Technique	Précision
[2]	2023	Porno / normal/ sexy/ hentai/ dessin	ConvNexT(tiny) ré-entraîné	94,90%
[14]	2022	Porno/non-porno	CNN entraîné sur un autre dataset	79,02%

I.3.5 Datasets

Le dataset NPDI-2K [29] est le plus utilisé, il contient 2000 vidéos(1000 pornographiques et 1000 non pornographique), certains auteurs ont extrait les images depuis les vidéos pour entraîner leur modèle, certains ont utilisé les vidéos en tant que tel pour entraîner leur modèle.

Le dataset LSPD [14] a 500000 images [14] en plus de 4000 vidéos (2000 pornographiques et 2000 non pornographique).
Nous remarquons que la catégorie sexy des images est plus constante et intéressante a utilisé. Nous remarquons également la présence de catégorie dessin et Hentai[6] séparé, ce dataset est assez récent et nous avons trouvé peu d'études l'utilisant, cependant sa qualité et quantité d'images et vidéos est remarquable.

Le dataset NudeNet[12] comporte 711324 images, nous notons que la catégorie sexy et pornographique sont par moment très similaire, la catégorie sexy est inconstante, certaines de ses images sont clairement pornographiques.

Le dataset NPDI-800 [7] est le plus vieux et le moins garni d'entre eux, avec 800 vidéo, nous pouvons le considérer comme dépasser par les nouveaux datasets.

Le dataset Xvideos comporte de loin le plus de vidéos pornographiques, cependant ces vidéos ne sont pas convenables pour l'entraînement d'un modèle.

Nous avons trouvé peu d'information sur les datasets APD-VIDEO et APD-2M, à part leur nombre de vidéos et d'images. Le dataset AIC est plutôt petit comparé aux autres.

Les tableaux I.7 et I.8 [15] présente les datasets et leurs nombres de vidéos et photos et les comparent.
Par conséquent, le dataset LSPD et NPDI-2K sont les plus intéressants, avec une bonne quantité et qualité.
Nous remarquons un manque d'étude sur le dataset LSPD, il est clair que ce dataset sera la référence dans ce domaine pour les années à venir, nous nous attendons à une croissance de sa popularité.
Dans le contexte de notre étude, trouvé un dataset qui contient des scènes NSFW issues de films ou séries est très intéressant, hélas nous n'avons rien trouvé.

14. Il existe aussi 50212 photos pornographiques avec annotation des organes sexuels
15. Les catégories Hentai[6] et dessin sont assignés à pornographique et non pornographique respectivement

TABLE I.7: Comparaison du nombre des vidéos des datasets

Dataset	Année	Pornographique	Non-pornographique	Total
NPDI-800 [7]	2013	400	400	800
NPDI-2K [29]	2016	1000	1000	2000
LSPD [14]	2022	2000	2000	4000
APD-VIDEO [7]	—-	—-	—-	6765
Xvideos [9]	—-	7M	0	7M

TABLE I.8: Comparaison du nombre des images des datasets

Dataset	Année	Pornographique	Non-pornographique	Sexy	Total
AIC[35]	2018	50000	50000	50000	150000
NudeNet[12]	2019	438390	230560	42374	711324
LSPD [14][15]	2022	250000	200000	50000	500000[14]
APD-2M[4]	—-	1070035	1150295	0	2220330

I.4 Conclusion

Dans ce chapitre, nous avons défini l'objectif précis de ce mémoire à travers des définitions. Nous avons effectué une revue de la littérature en présentant et comparant les travaux connexes, discutant des techniques utilisés et leurs performances, ainsi que des datasets, leurs contenus et tailles, en nous aidant de tableaux pour mieux visualiser les informations.

Dans le chapitre suivant, nous présenterons notre approche proposée.

Chapitre II

Conception

II.1 Introduction

Dans ce chapitre, nous allons expliquer notre solution en présentant l'architecture générale de notre projet. Nous allons aussi présenter les datasets créés et utilisés, que ce soit lors des entraînements ou les évaluations.
Nous présenterons aussi l'architecture du modèle image et son, ainsi que l'architecture du modèle vidéo. Tout au long de notre travail, nous avons utilisé la méthode Extreme Programming (XP) [1], qui est une méthodologie de développement de logiciels agile.

II.2 Solution proposée

Dans cette partie, nous allons détailler le processus de création des modèles, les datasets collectés et utilisés ainsi que l'entrainement des modèles.

II.2.1 Architecture générale

Comme le démontre la figure II.1, nous avons commencé par la recherche et la collecte des datasets adaptés à notre projet, nous avons ensuite procédé à la création, l'entrainement et l'évaluation du modèle visuel puis du modèle sonore en collaboration avec notre client afin de créer des modèles efficaces et adapté à ses besoins. Ensuite, nous avons créé un modèle vidéo en proposant une solution permettant de combiner les deux modèles précédents.

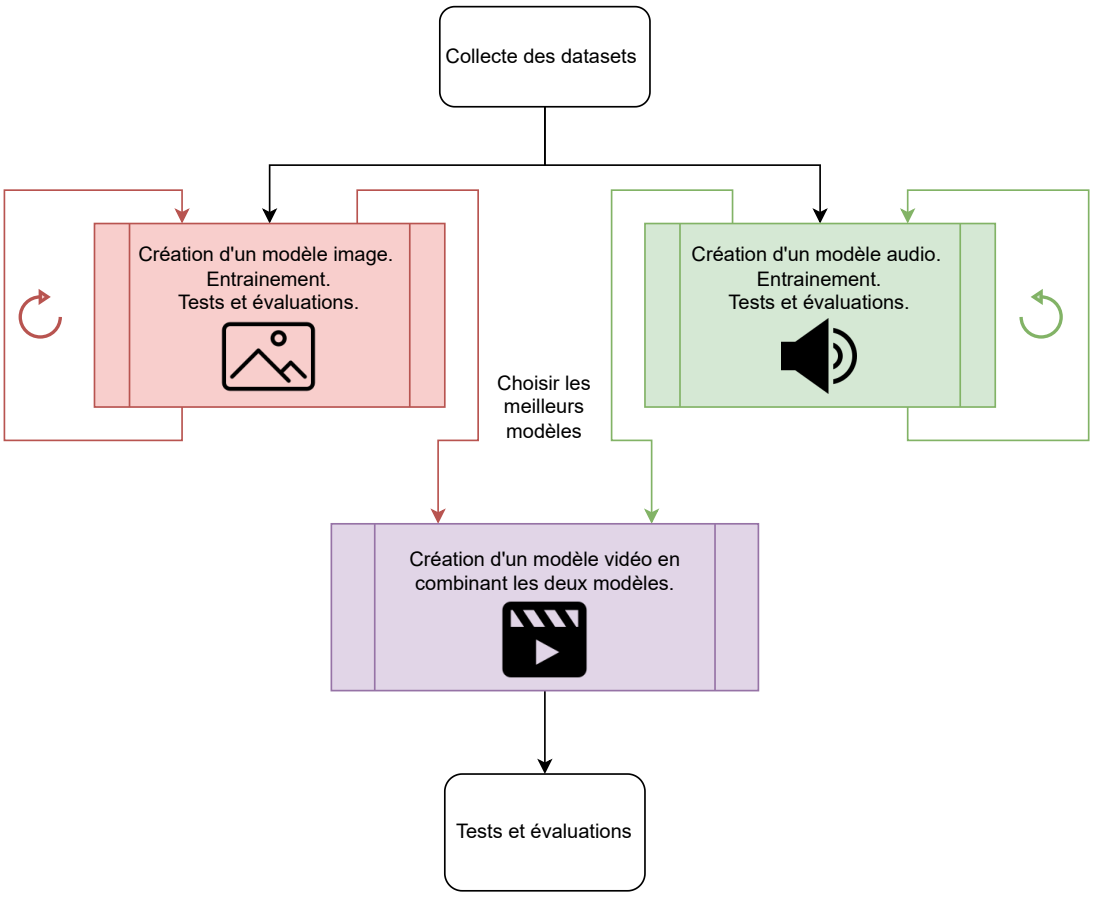


FIGURE II.1 – Architecture générale

II.2.2 Collecte des datasets

Après avoir effectué des recherches sur les travaux précédents, nous avons conclu sur l'utilisation des datasets LSPD [2] et NPDI-2K [3], bien que nous ayons créés d'autres datasets, ils sont tous issus des datasets LSPD et NPDI-2K.

Le dataset NPDI-2K est le plus utilisé dans les travaux précédents, l'utiliser pour effectuer des comparaisons est un choix judicieux. Comme son nom l'indique, ce dataset à 2000 vidéos divisé en deux classes (pornographique et normal).

Le dataset LSPD est le plus récents et le plus complet des datasets, il contient des vidéos classées en deux catégories, et des images classées en cinq catégories, en plus d'un dataset pour Sensitive Object Detection (SOD).

Nous avons nommé le dataset LSPD contenant les images : LSPD images 5 classes, ce dataset a cinq classes (Normal, Hentai, Sexy, Porno, Dessin) combinant au total 500000 images.

1. `http://www.extremeprogramming.org/`, consulté le 22/06/2024
2. `https://sites.google.com/uit.edu.vn/LSPD`, consulté le 22/06/2024.
3. Obtenu en contactant l'un de ses créateurs.

TABLE II.1: Comparaison des datasets utilisés

Dataset	Type	Normal	Pornographique	total
LSPD image cinq classes	Images	250000	250000	500000
LSPD image deux classes	Images	152000	200000	352000
LSPD vidéo	Vidéos	2000	2000	4000
LSPD images extraites des vidéos	Images	195213	283540	478753
LSPD audio	Audios	98493	142847	241340
NPDI-2K	Vidéos	1000	1000	2000
LSPD vidéo test	Vidéos	89	132	221
NPDI-2K test	Vidéos	305	468	773
Notre dataset	Vidéos	165	187	352

Nous avons nommé le dataset LSPD contenant les vidéos : LSPD vidéo, ce dataset a deux classes (pornographique et normal), il contient 4000 vidéos de duré variable.

Nous avons créé les datasets mentionné ci-dessous à partir des datasets LSPD et NPDI-2K, le tableau II.1 résume les datasets utilisés dans notre travail ainsi que leurs tailles.

— LSPD image deux classes : ce dataset contient les deux classes (pornographique et normal) du dataset LSPD pour un total de 352000 images.
L'utilisation de deux classes seulement permet de réduire la complexité et la ressemblance des données, les classes dessin, hentai et sexy ne sont pas aussi pertinentes que les deux autres dans notre contexte. De ce fait, ce dataset permet d'entrainer un modèle solide pour la détection de la nudité et la pornographie.

— LSPD images extraites des vidéos : ce dataset a été construit à partir du dataset LSPD vidéo, nous avons sélectionné manuellement les vidéos pornographiques et aléatoirement les vidéos normales, puis nous avons extrait chaque second une image depuis les vidéos sélectionnées, ensuite, nous avons supprimé les images ayant une luminosité extrême, c'est-à-dire sous-exposé ou surexposé, nous avons aussi supprimé une image sur deux afin de réduire la taille du dataset et de limiter la ressemblance entre images.
Ce dataset a au total 478753 images. Ce dataset a été divisé en entraînement, validation et test en sélectionnant 90%, 5% et 5% respectivement aléatoirement.
Le but de notre projet est la détection sur vidéo, ce dataset permet d'adapter notre modèle aux images présente dans les vidéos, en effet, nous remarque une différence significative entre les images et les images des vidéos, avoir un dataset construit seulement d'images ex-

traites depuis les vidéos est important et permet d'avoir à la fin un modèle plus performant et adapté.

— LSPD audio : ce dataset a été créé en sélectionnant des vidéos pornographiques manuellement et des vidéos normales aléatoirement depuis le dataset LSPD vidéo, ensuite, nous avons vérifié l'existence de l'audio dans les vidéos sélectionnées, nous avons ensuite extrait l'audio et nous l'avons transformé en un seul canal (mono) en faisant une moyenne des canaux, puis nous avons changé le taux d'échantillonnage à 16 kHz, nous avons coupé l'audio en portions d'une seconde, et on a supprimé les portions silencieuses.
  Ce dataset a au total 241340 fichiers audios d'une durée d'une seconde. Ce dataset a été divisé en entraînement, validation et test en sélectionnant 90%, 5% et 5% respectivement aléatoirement.
  Le manque de dataset dans ce contexte nous a poussé à le créer, malgré certaine inconsistance dans l'étiquetage des fichiers, nous l'avons utilisé pour l'entrainement et l'évaluation du modèle sonore.

— LSPD vidéo test : ce dataset a été créé à partir du dataset LSPD vidéo en sélectionnant aléatoirement des vidéos pornographiques et normales, puis nous les avons découpés en vidéos d'une minute, le but est d'augmenter le nombre de vidéos afin d'obtenir des résultats plus pertinent.
  ce dataset a au total 221 vidéos d'une durée d'une minute au maximum, ce dataset a été seulement utilisé pour les tests.

— NPDI-2K test : ce dataset a été créé à partir du dataset NPDI-2K en sélectionnant aléatoirement des vidéos pornographiques et normales, puis nous les avons découpés en vidéos d'une minute, le but est d'augmenter le nombre de vidéos afin d'obtenir des résultats plus pertinent.
  ce dataset a au total 773 vidéos d'une durée d'une minute au maximum, ce dataset a été seulement utilisé pour les tests.

— Notre dataset : pour construire ce dataset, nous avons sélectionné aléatoirement des vidéos diffusées sur l'EPTV trouvé sur YouTube [4], nous les avons téléchargées avec différentes résolutions puis nous les avons coupés en vidéos d'une minute au maximum. Pour la partie pornographique, nous avons sélectionné aléatoirement des vidéos depuis la classe pornographie des datasets LSPD vidéo test et NPDI-2K test.
  Ce dataset a au total 352 vidéos et a été utilisé seulement pour les tests.

II.2.3 Création du modèle visuel

Dans cette sous-section, nous allons présenter le modèle visuel, son architecture ainsi que les différents paramètres de son entrainement.

4. `https://youtube.com/@chaineeptv`, consulté le 16/06/2024

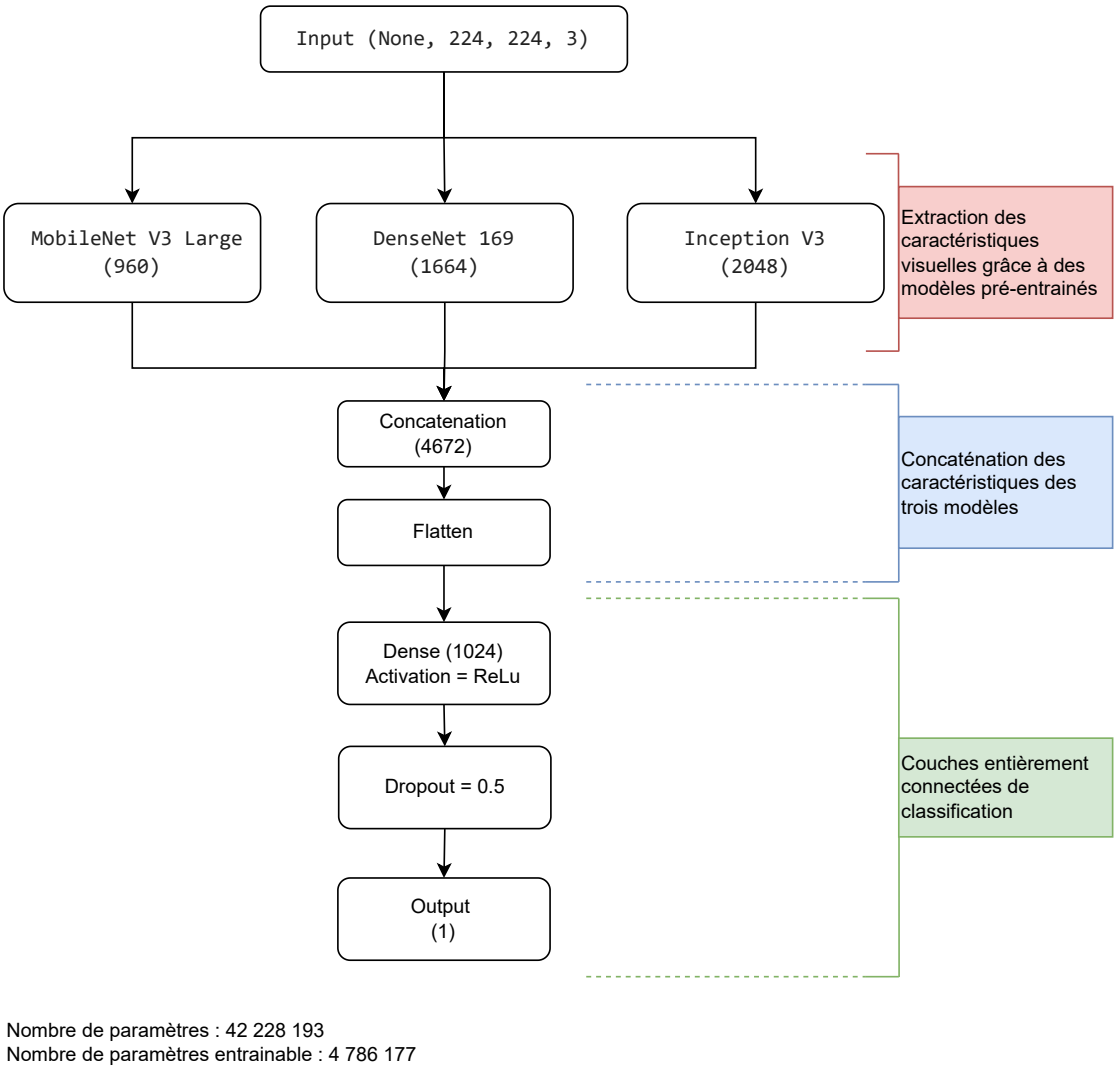


FIGURE II.2 – Architecture du modèle image

II.2.3.1 Architecture du modèle visuel

Pour créer notre modèle de détection de la pornographie dans les images, nous avons combiné des modèles pré-entrainés pour l'extraction de caractéristiques, auxquelles, nous avons ajouté des couches de classifications, comme le représente la figure II.2.

— **Entrée :** Notre modèle accepte une image Rouge Vert Bleu (RVB) avec une résolution de 224 par 224 pixels, les valeurs doivent être normalisées, c'est-à-dire comprises entre 0 et 1, cette étape permet d'accélérer l'entrainement et d'avoir une entrée consistante quelle que soit la profondeur de couleur de l'image.

— **Extraction de caractéristiques visuelles :** nous avons combiné trois modèles CNN pré-entrainés : MobileNetV3-Large, DenseNet169 et InceptionV3 (tous figés et avec les poids

d'ImageNet), nous avons choisi ces modèles en se basant sur les résultats obtenus par les travaux précédents tout en choisissant la version la plus récente (Ex : MobileNetV3-Large a la place de MobileNet-V2).
Nous avons aussi testé le modèle ConvNext(tiny) qui contrairement a l'étude le testant sur le dataset LSPD n'a pas performé aussi bien que InceptionV3, ce qui explique la raison pour laquelle, nous avons préféré l'employé à la place de ConvNext(tiny).
Le choix de MobileNetV3-Large et DenseNet169 est dû à leurs performances élevées dans les études précédentes.
Nous avons combiné ces trois modèles afin de maximiser les performances de notre modèle. En effet, combiner plusieurs modèles permet d'extraire davantage de caractéristiques visuelles, et qui en conséquence, accroît les performances de notre modèle final.

— **Concaténation et aplatissement :** le résultat des modèles pré-entrainés sont concaténés et aplatit afin de pouvoir les connecter avec les couches suivantes.

— **Couches de classifications :** nous avons créé un réseau constitué d'une couche Dense de 1024 neurone avec une activation Rectificateur Linéaire Unitaire (ReLU), connecté à une couche Dropout avec un taux de 50 %, cette dernière permet d'éviter le sur-apprentissage. Pour finir, cette couche est connectée à une sortie binaire, avec une activation sigmoïde.

Le modèle a 42 millions de paramètres dont 4788177 entrainables, nous pouvons le considérer comme un modèle intermédiaire en se basant sur sa taille et complexité.

II.2.3.2 Entrainement du modèle visuel

La figure II.3 résume l'architecture de l'entrainement, qui s'est déroulé en deux étapes :

1. **Entrainement sur le dataset LSPD image deux classes :** l'entrainement sur les images pornographiques et normales permet de créer un modèle robuste le préparant ainsi à son prochain entrainement.
   Voici comment s'est déroulé cet entraînement :
   — **Augmentation des données :** l'augmentation des données ce fait en temps réel, chaque image reçoit aléatoirement une ou plusieurs des modifications suivantes :
   — Rotation : jusqu'à 20°.
   — Translation horizontale : jusqu'à 20%.
   — Translation verticale : jusqu'à 20%.
   — Cisaillement : jusqu'à 20%.
   — Zoom : jusqu'à 20% en avant ou en arrière.
   — Retournement horizontal.
   — Ajustement de la luminosité : luminosité ajustée aléatoirement dans une plage de 50% à 150% de la luminosité originale.

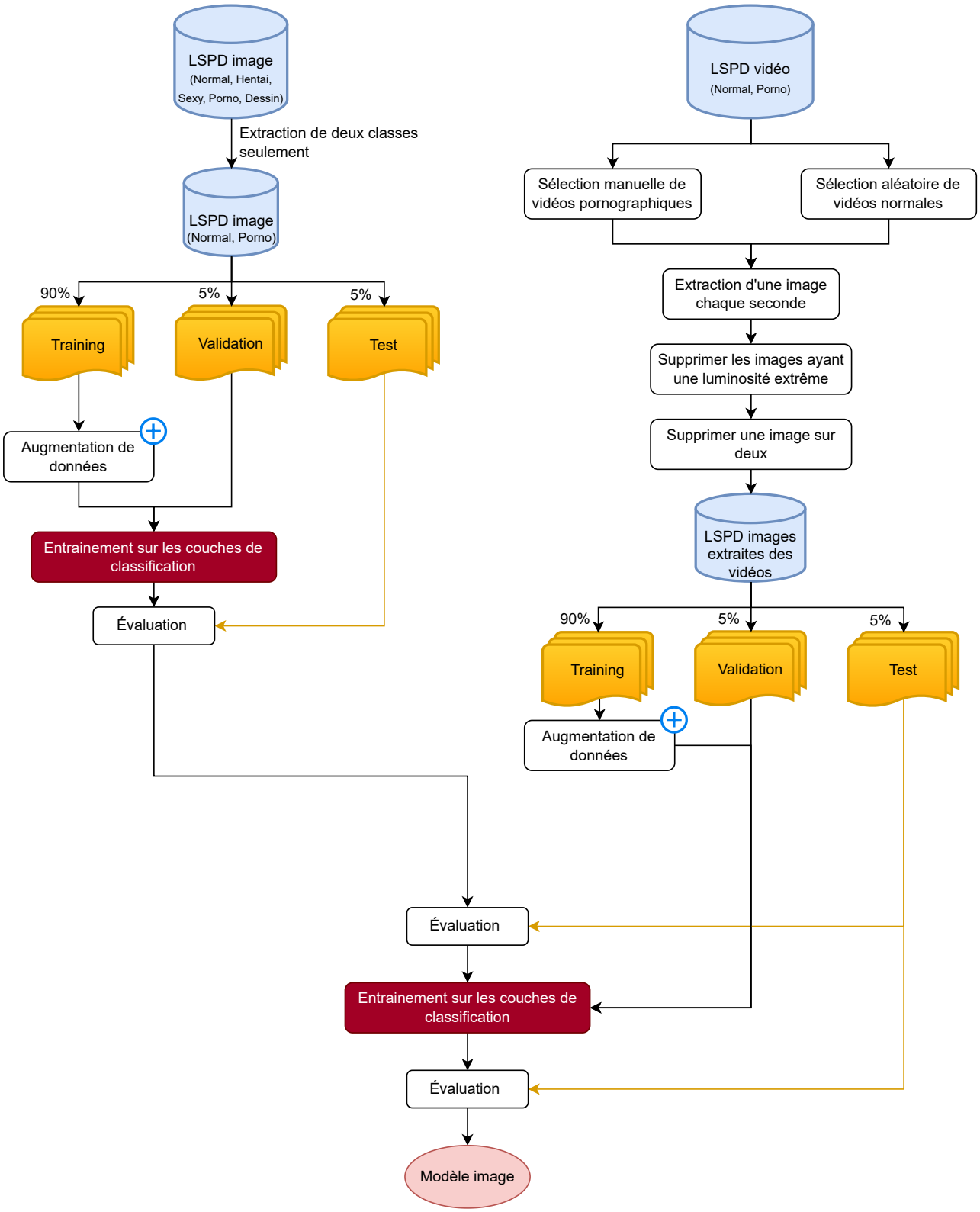


FIGURE II.3 – Architecture de l'entraînement du modèle visuel

TABLE II.2: Variation du taux d'apprentissage par epochs

Epoch	Taux d'apprentissage
$[1, 2]$	$1e{-3}$
$[3, 4]$	$5e{-4}$
$[5, 10]$	$1e{-4}$
$[11, 12]$	$1e{-5}$

Ces modifications permettent d'augmenter les performances du modèle et de le rendre plus robuste dans les cas réels, et cela, en proposant des images légèrement modifiées à chaque epoch, ce qui contribue à un meilleur entrainement.

— **Entrainement :** les classes ont été équilibrées par poids, permettant de mieux gérer les différences du nombre d'images entre les classes pendant l'entrainement et d'obtenir des résultats plus équilibrés.
L'entrainement s'est déroulé sur 12 epochs avec un batch size de 128 images, le modèle a été compilé avec un optimiseur Adam et une fonction de perte de type entropie croisée binaire, le taux d'apprentissage est variable, cette variation est présentée dans le tableau II.2, elle permet au début d'accélérer l'entrainement, et a la fin d'obtenir des meilleurs résultats en évitant des fluctuations trop importante dans les performances.
Un arrêt précoce, ou Early Stopping en anglais, a aussi été utilisé avec une patience de 5 epochs, ce mécanisme permet d'éviter le sur-apprentissage.

— **Évaluation :** l'évaluation ce fait sur des données différentes de celle l'entrainement et de la validation, afin de tester le modèle sur des nouvelles données et de ce fait avoir des résultats plus réaliste et pertinent.
Effectuer une évaluation a ce stade permet déjà de pouvoir comparer le modèle image avec d'autres modèles plus aisément, mais aussi de voir la différence entre les performances de ce modèle sur les images des dataset LSPD images deux classes et LSPD images extraite des vidéos.

2. **Entrainement sur le dataset LSPD images extraites des vidéos :**

— **Évaluation :** faire une évaluation avant le ré-entrainement du modèle afin de connaitre les progrès fait.

— **Augmentation des données :** même technique utilisé précédemment.

— **Entrainement :** l'entrainement s'est déroulé sur 15 epochs avec un batch size de 256 images, le modèle a été compilé avec un optimiseur Adam avec un taux d'apprentissage fixe de : $1e{-4}$ et une fonction de perte de type entropie croisée binaire.

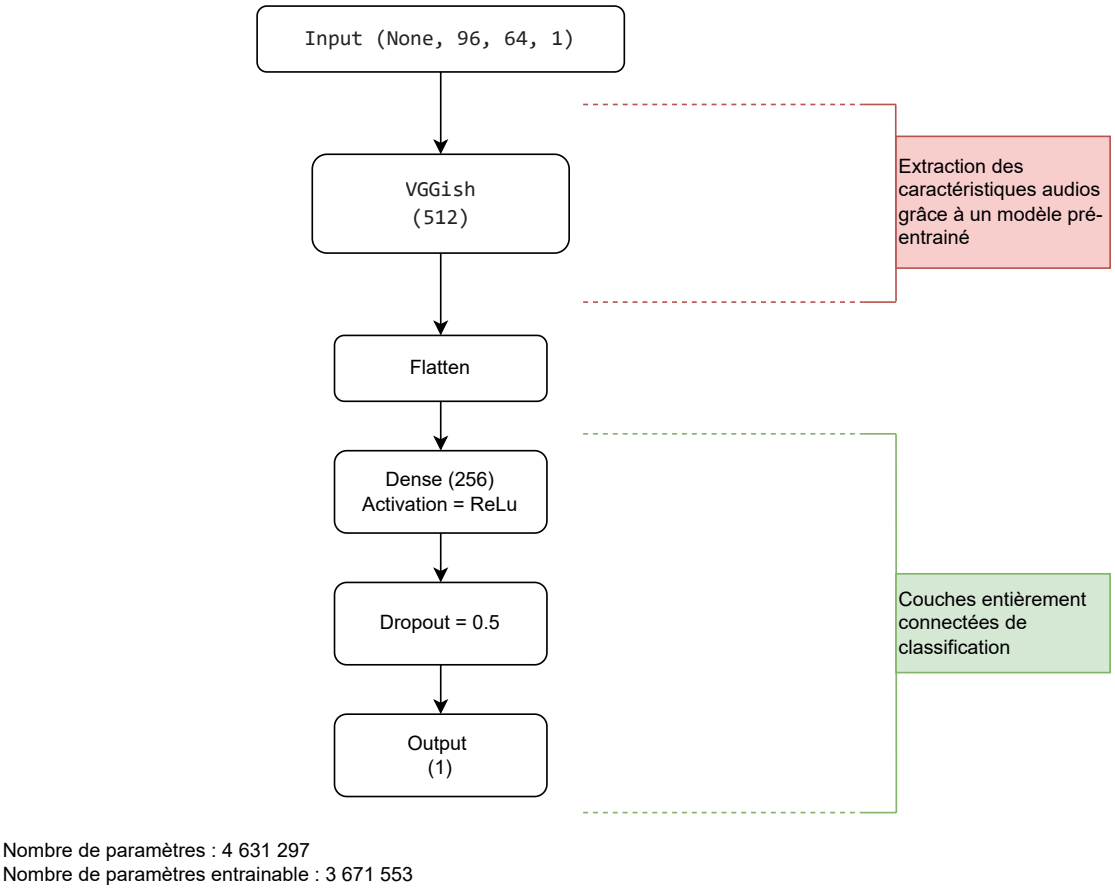


FIGURE II.4 – Architecture du modèle audio

Un arrêt précoce, ou Early Stopping en anglais, a aussi été utilisé avec une patience de 3 epochs, ce mécanisme permet d'éviter le sur-apprentissage.

Le modèle étant déjà entrainé, le but de ce second entrainement est de l'adapter à son utilisation finale qui est la détection d'images extraite de vidéo. En effet, les images extraites des vidéos sont différentes de celle prise en tant que photo, la différence est : la qualité globale de l'image et la présence de flou de mouvement important dans les images extraite à partir de vidéo, cela est dû à la différence de réglage lors de la prise de vue.

— **Évaluation :** une évaluation finale permet de tester notre modèle final sur ce dataset et de comparer les résultats avec ceux de la première évaluation.

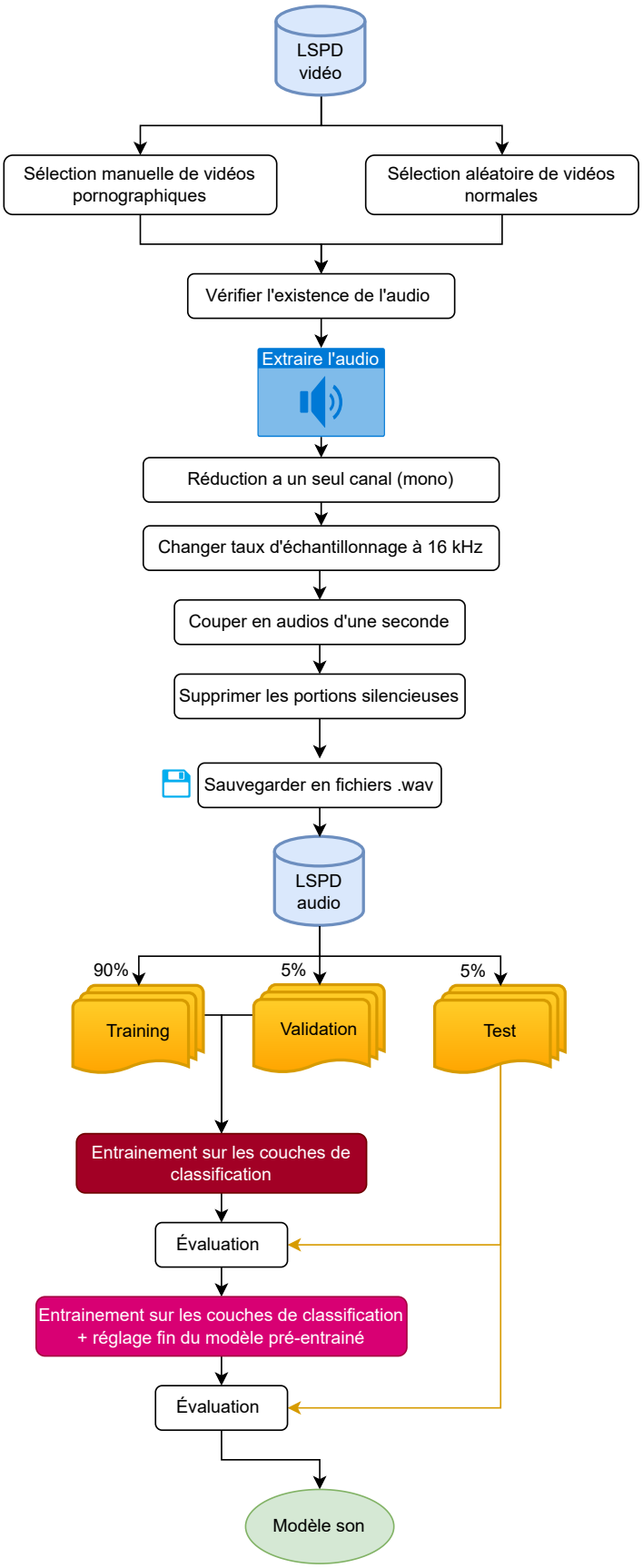


FIGURE II.5 – Architecture de l'entrainement du modèle son

II.2.4 Création du modèle sonore

Dans cette sous-section, nous allons présenter le modèle sonore, son architecture ainsi que les différents paramètres de son entrainement.

II.2.4.1 Architecture du modèle sonore

Pour créer notre modèle de détection de la pornographie dans l'audio, nous avons utilisé un modèle pré-entrainé pour l'extraction de caractéristiques audio, auxquelles, nous avons ajouté des couches de classifications, comme le présente la figure II.4.

— **Entrée :** notre modèle accepte une entrée au format (None, 96, 64, 1), qui permet de représenter le spectre Log Mel d'un audio mono (un seul canal) de 0,96 second avec un taux d'échantillonnage de 16 kHz, normalisé entre 0 et 1.
Ce traitement permet d'avoir une entrée consistante quels que soient le type et le format de l'audio. L'utilisation du spectre Log Mel est une approche courante et efficace utilisé pour la classification de son, elle permet d'obtenir des bonnes performances tout en réduisant la dimensionnalité et la complexité du son.

— **Extraction de caractéristiques audios :** nous avons utilisé un modèle pré-entrainé nommé VGGish avec les poids d'AudioSet. Il s'agit d'un modèle efficace pour l'extraction de caractéristiques audios.

— **Couches de classifications :** nous avons créé un réseau constitué d'une couche Dense de 256 neurones avec une activation ReLU, connecté à une couche Dropout avec un taux de 50 %, permettant d'éviter le sur-apprentissage.
Pour finir, cette couche est connectée à une sortie binaire, avec une activation sigmoïde.

Le modèle final à 4631297 paramètres dont 3671553 entrainables, nous pouvons le considérer comme un modèle non-complexe et rapide.

II.2.4.2 Entrainement du modèle sonore

Comme la figure II.5 l'indique, l'entrainement de ce modèle est composé de plusieurs étapes :

— **Premier entrainement :** cet entrainement concerne seulement sur les couches de classifications, le modèle pré-entrainé est entièrement gelé et le nombre de paramètres entrainable est de : 131585.
Cet entrainement est composé de 20 epochs, avec un batch size de 32 fichiers audios, le modèle a été compilé avec un optimiseur Adam avec un taux d'apprentissage fixe de : $1e{-}4$ et une fonction de perte de type entropie croisée binaire.

— **Évaluation :** une évaluation permettant de connaitre les performances du modèle après le premier entrainement.

— **Second entrainement :** pour cet entrainement, le modèle pré-entrainé a été partialement dégelé, les quatre dernières couches du modèle ont été entrainés pour un total de 3671553

paramètres entrainable. Cette étape de Fine Tuning permet de mieux adapter le modèle pré-entrainé à nos données.
Cet entrainement est composé de 20 epochs, avec un batch size de 32 fichiers audios, le modèle a été compilé avec un optimiseur Adam avec un taux d'apprentissage fixe de : $1e{-}4$ et une fonction de perte de type entropie croisée binaire.

— **Évaluation :** une évaluation finale permettant de connaitre les performances du modèle finale, et de le comparer avec le modèle pré Fine Tuning.

II.2.5 Création du modèle vidéo

Dans cette sous-section, nous allons présenter le modèle vidéo, son architecture et son processus de fonctionnement.

II.2.5.1 Architecture du modèle vidéo

Nous avons créé un modèle permettant de détecter la pornographie dans une vidéo en utilisant les deux modèles présentés précédemment (modèle de détection d'image et modèle de détection de son), la figure II.6 résume son processus de fonctionnement.
Utiliser cette approche nous permet d'améliorer ou de changer les modèles image et son utilisés facilement.

— **Entrée :** le modèle accepte le chemin d'une vidéo et un argument appelé option qui permet de choisir le type et nombre de données retourné.
Le modèle nécessite aussi de passer le modèle image et son qu'on souhaite utiliser.

— **Vérification de la vidéo :** le modèle vérifie si le chemin du fichier donné existe vraiment, puis il vérifie si le fichier est bel est bien une vidéo grâce à son extension (mp4 et avi).
Si la vidéo existe, le modèle la charge et commence à la lire.
Le cas échéant, le modèle retourne None indiquant une erreur dans l'exécution.

— **Coté audio :**

  — **Vérification du son :** le modèle vérifie l'existence d'un contenu audio dans la vidéo, cette étape permet de gérer les vidéos n'ayant pas de son en ignorant cette partie.

  — **Enregistrement de fichiers audios :** enregistre chaque cinq minutes dans un fichier audio (wav) temporaire, cette étape permet de minimiser l'utilisation de la mémoire de la machine en divisant les gros fichiers en segments plus petits.

  — **Vérification de l'audio :** vérifier que l'audio en question n'est pas silencieux grâce à un threshold son de -50 dB, si le volume sonore est inférieur à ce threshold, nous ignorons ce fichier audio et nous passons au suivant, cette étape permet de ne pas fausser les résultats en évitant de prédire des portions trop silencieuses.

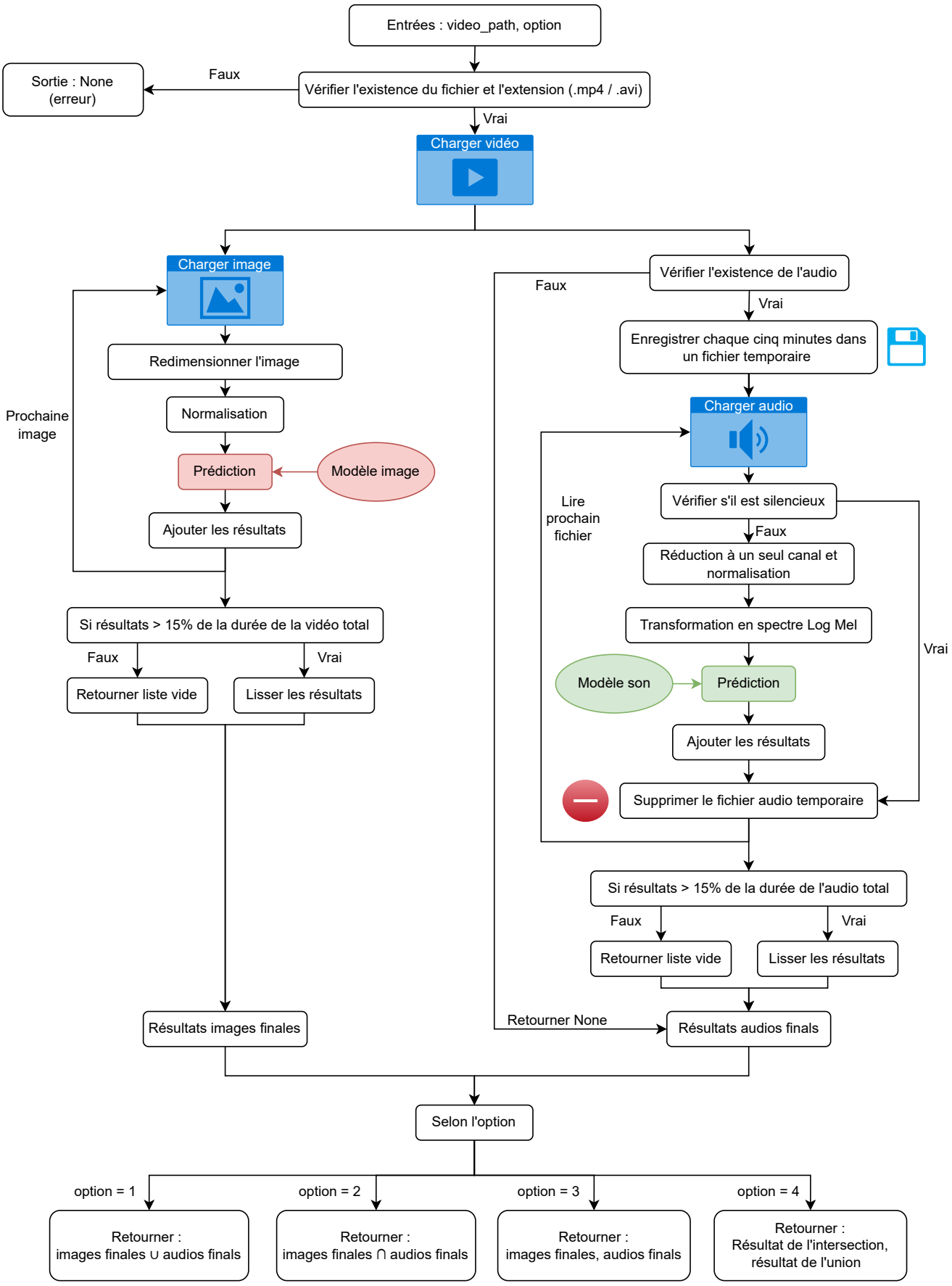


FIGURE II.6 – Architecture du fonctionnement modèle vidéo

— **Préparation du fichier audio :** préparer le fichier audio en le convertissant à un format compatible avec l'entrée du modèle audio, c'est-a-dire le spectre Log Mel d'un fichier audio mono normalisé. Cette étape est essentielle afin de pouvoir exécuter la prédiction.

— **Prédiction :** le modèle son passé comme argument effectue la prédiction et retourne des résultats sous forme de liste de segments d'une longueur de 0,96 seconde où le modèle à détecter de la pornographie, le segment est lui-même une liste ayant deux éléments, le premier est le code temporel du début de la détection et le second celui de la fin de la détection.

L'équation II.1 montre le format des résultats, une liste vide signifie la non-détection de la pornographie.

$$[(\text{code temporel du début de la détection}, \text{code temporel de la fin de la détection}), \ldots] \tag{II.1}$$

La prédiction en elle-même retourne un numéro compris entre 0 et 1 (1 étant pornographique et 0 normal), un threshold de sensibilité de 0,98 permet de classifier l'audio en tant que normal si elle est inférieure à ce threshold, sinon pornographique. Le choix de ce threshold peut totalement changer le comportement du modèle, après quelques expérimentations manuelles, nous avons choisi 0,98, cependant nous n'avons pas effectué des tests plus approfondis.

— **Suppression du fichier temporaire :** dans tous les cas, le fichier temporaire est supprimé et le modèle passe au fichier audio suivant.

— **Résultat final du côté son :** un threshold général sur la durée de détection de la pornographie par rapport à la durée totale de la vidéo est fixé à 15%, si le modèle détecte moins de 15% de la durée totale de vidéo comme étant pornographique, il ignore ce résultat, sinon il retourne le résultat après l'avoir lissé.

Cette étape permet de minimiser l'impact des erreurs sur la prédiction totale de la vidéo.

— **Lissage des résultats :** le but de cette étape est de combiner les résultats ayant une seconde ou moins entre eux, ça permet de simplifier le résultat et d'améliorer la lisibilité.

Le tableau II.3 donne un exemple qui montre le fonctionnement de cette étape.

TABLE II.3: Exemple de la fonction de lissage avec un threshold d'une seconde

Résultat pré-lissage	Résultat post-lissage
$[(2,3),(3,4),(4,5),(6,7),(9,10)]$	$[(2,7),(9,10)]$

— **Coté image :** pour accélérer le temps d'exécution, nous avons créé trois threads, un pour la lecture des images en utilisant une file d'attente de 10 éléments au maximum, un autre

thread pour le traitement des images avec une autre file d'attente d'une taille de 10 éléments au maximum, et finalement un thread pour la classification afin d'exécuter la prédiction. Ces threads peuvent être exécutés en simultané à part celui de la classification qui est exécuté seul, car il a besoin de plus de ressource afin de performer.
Le principe est que le processus prépare jusqu'à dix images en les lisant et traitant, et le GPU exécute les images prêtes qui sont dans la file d'attente. Ce système de relai entre CPU et GPU exploite au mieux les ressources disponibles. La taille optimale des files d'attentes dépend des caractéristiques matérielles du système utilisé.

- **Chargement et préparation de l'image :** le modèle lit une image chaque seconde, la redimensionne et normalise ses valeurs afin de la préparer et de la rendre compatible avec le modèle de prédiction d'image.
  L'intervalle entre les images est passé comme un argument et peut être changé, l'augmenter permet d'avoir plus de précision en dépit d'un temps d'exécution plus conséquent.
- **Prédiction :** le modèle image passé comme argument effectue la prédiction et retourne des résultats sous forme de liste de segments d'une longueur d'une seconde où le modèle à détecter de la pornographie, le résultat a le même format que celui du côté audio, l'équation II.1 montre le format du résultat.
  La prédiction en elle-même retourne un numéro compris entre 0 et 1 (1 étant pornographique et 0 normal), un threshold de sensibilité de 0,9 permet de classifier l'image en tant que normal si elle est inférieure à ce threshold, sinon pornographique.
  Le choix de ce threshold peut totalement changer le comportement du modèle, après quelques expérimentations manuelles, nous avons choisi 0,9, cependant nous n'avons pas effectué des tests plus approfondis.
- **Résultat final du côté image :** cette étape est exactement la même que celle du côté du son.
- **Lissage des résultats :** cette étape est exactement la même que celle du côté du son, le tableau II.3 montre le fonctionnement du lissage.

II.3 Conclusion

Dans ce chapitre, nous avons présenté l'architecture générale de notre projet, nous avons aussi expliqué les datasets créés et utilisés en expliquant la façon dont nous les avons créés ainsi que leur rôle dans notre travail. Nous avons aussi montré l'architecture et l'entrainement de notre modèle visuel et sonore. Pour finir, nous avons expliqué le fonctionnement de notre modèle vidéo.
Tout au long de ce chapitre, nous nous sommes aidés de figures qui permettent de mieux expliquer et visualiser chaque étape, tout en donnant les détails pour chacun des choix que nous avons faits.

Chapitre III

Implémentation et tests

III.1 Introduction

Dans ce chapitre, nous allons détailler l'implémentation de notre projet, en commençant par une description des spécifications matérielles et de l'environnement logiciel utilisé. Ensuite, nous allons examiner les performances du modèle en utilisant des mesures classiques telles que l'Accuracy, la précision, le Recall, le F1-score et la matrice de confusion. Nous présenterons également les expériences menées pour évaluer la capacité du modèle à détecter la pornographie dans les images et le son. Chaque expérience sera décrite en détail, mettant en évidence les résultats obtenus et les défis rencontrés. Enfin, une comparaison des différentes approches sera effectuée avant de conclure sur les performances globales du modèle final. Nous avons créé un Notebook [1] sur Google Colab permettant de tester notre modèle vidéo en ligne aisément.

III.2 Spécifications matérielles

Pour la réalisation de notre travail, nous avons utilisé deux machines, un laptop et un ordinateur, leurs performances sont expliquées dans le suivant :

- **Laptop :**
  - **processeur :** Intel(R) Core(TM) i5-1135G7 @ 2.40GHz 2.42 GHz
  - **RAM :** 4,00 Go
  - **OS :** Système d'exploitation 64 bits, processeur x64
- **Ordinateur :**
  - **processeur :** Intel(R) Core(TM) i5-12400, 2500 MHz, 6 coeurs, 12 threads
  - **RAM :** 32,00 Go
  - **GPU :** NVIDIA GeForce RTX 3060 Ti 8GB GDDR6
  - **OS :** Windows 11 23H2 : Système d'exploitation 64 bits, processeur x64

1. `https://colab.research.google.com/drive/1APZ3DeBlRRzZiNcPadAswbGyxw5rFflC`

Tout les entraînements et expériences ont été réalisés sur l'ordinateur.

III.3 Environnement logiciel

Nous avons utilisé le langage de programmation Python, la version *3.10.14* et nous avons utilisé la distribution Anaconda, nous allons mieux expliquer cet environnement de développement et nous parlerons aussi des bibliothèques de python et leurs rôles.

III.3.1 Anaconda

Il s'agit d'une distribution de logiciels open-source des langages de programmation Python et R [2]. Elle est utilisée pour la science des données et inclut des composés comme Conda, un gestionnaire de paquets et d'environnements virtuels, et l'Anaconda Navigator permet d'offrir les environnements de développement comme le Jupyter Notebook *7.0.8*. Dans notre ordinateur, nous avons une version *3.10.14*.

III.3.2 Bibliothèques python

Pour la création de notre modèle d'apprentissage profond, nous avons utilisés plusieurs bibliothèques dont les principales sont listées ci-dessous.

— **NumPy :** C'est une bibliothèque Python utilisé pour les calculs spécifiques et numériques, elle permet de créer des structures de données (Tableau, matrice...) et traite ces structures avec des opérations soit pour redimensionner ou indexer ou manipuler les tableaux et permet aussi d'utiliser les fonctions mathématiques.
La version de cette bibliothèque utilisée sur les deux machines est *1.26.4*.

— **Pandas :** C'est une bibliothèque Python construite sur NumPy, Conçu pour simplifier le travail avec les données, elle est utilisée pour la manipulation et l'analyse de données (importation des données de format CSV ou Exel ou JSON..., le filtrage et le prétraitement des données), permet aussi la manipulation flexible de données où nous pouvons sélectionner les données facilement avec les étiquettes et des positions.
La version de cette bibliothèque utilisée sur les deux machines est *2.2.1*.

— **Keras :** C'est une bibliothèque Python conçu pour la formation de réseaux de neurones profonds, elle fonctionne sur plusieurs backends de traitement comme CNTK, Tensorflow, Theano et dans notre travail, elle fonctionne sur le backend Tensorflow version *2.10.0* où nous avons utilisé le GPU dans l'exécution des programmes, grâce à cette bibliothèque, les utilisateurs peuvent ajouter leurs propres couches aux modèles qu'ils ont créés.
La version de cette bibliothèque utilisée sur les deux machines est *2.10.0*.[24]

2. `https://www.anaconda.com/`, consulté le 22/06/2024

— **OpenCV :** C'est une bibliothèque Python utilisé pour le traitement d'images et la vision par ordinateur, elle permet d'analyser ou manipuler des images et des vidéos où nous pouvons extraire les images depuis les vidéos ou afficher les images et faire le prétraitement de ces images (redimensionnement, rotation, conversion de formats de couleur...) ou faire le filtrage d'images (flou...).
La version de cette bibliothèque utilisée sur les deux machines est *4.9.0*.

— **Pydub :** C'est une bibliothèque Python utilisé spécialement pour les audios où elle lit ou écrire un fichier audio ou convertie la forme de l'audio (mp3,wav...) ou faire une manipulation sur les audios tels que découpe l'audio ou modification du volume.

— **Scipy :** C'est une bibliothèque Python construite d'après la bibliothèque Numpy utilisé pour les calculs scientifiques et techniques et pour plusieurs choses telles que le traitement du signal et l'analyse statistique.
La version de cette bibliothèque utilisée sur les deux machines est *1.13.1*.

— **MoviePy :** C'est une bibliothèque Python utilisé pour manipuler des vidéos telles que lecture et l'écriture, et découpé la vidéo et traiter les images de la vidéo individuellement et l'exportation de la vidéo sur d'autre format.
La version de cette bibliothèque utilisée sur les deux machines est *2.0.0.dev2*.

— **Scikit-learn :** C'est une bibliothèque Python utilisé pour le Machine Learning et l'analyse de données, elle permet de calculer et d'afficher les performances de notre modèle comme la précision, le recall-score, le F1-score et la matrice de confusion.
La version de cette bibliothèque utilisée sur les deux machines est *1.5.0*.[31]

Nous avons utilisé plusieurs autres bibliothèques comme Splitfolders utilisé pour diviser la dataset sur des parties d'entraînement, de validation et de test. Os effectue des opérations courantes comme la gestion de fichiers et de répertoires, Matplotlib affiche les résultats sous forme de graphes, Seaborn est une bibliothèque de visualisation de données basée sur la bibliothèque Matplotlib et conçue pour la création de graphiques statistiques comme la matrice de confusion.

Dans notre travail, nous avons créé un modèle visuel où nous avons combiné des modèles pré-entrainés et que nous avons ré-entrainé sur de nouvelles dataset.

III.4 Évaluation des performances

Pour déterminer les performances de nos modèles, nous avons utilisé les taux de :

— **Vrais Positifs (TP) :** prédiction positive correcte (en réalité positif).

— **Vrais Négatifs (TN) :** prédiction négative correcte (en réalité négatif).

— **Faux Positifs (FP) :** prédiction positive incorrecte (en réalité négatif).

— **Faux Négatifs (FN) :** prédiction négative incorrecte (en réalité positif).

Dans notre contexte, cela se traduit par :

— **Vrais Positifs (TP) :** contenu pornographique détecté correctement.

— **Vrais Négatifs (TN) :** contenu normal détecté correctement.

— **Faux Positifs (FP) :** contenu normal détecté comme étant pornographique.

— **Faux Négatifs (FN) :** contenu pornographique détecté comme étant normal.

Sur la base de cette répartition, nous avons calculé les métriques suivantes.

III.4.1 Accuracy

Mesure la proportion de prédictions exactes parmi toutes les instances et est calculé comme suit :

$$\frac{TP + TN}{TP + FP + TN + FN}$$

III.4.2 Précision

Mesure la proportion de prédictions positives parmi toutes les instances prédites comme positive et est calculé comme suit :

$$\frac{TP}{TP + FP}$$

III.4.3 Recall

Mesure la capacité du modèle à identifier correctement les instances positives et est calculé comme suit :

$$\frac{TP}{TP + FN}$$

III.4.4 F1-score

Il est calculé en fonction de la précision et du rappel comme suit :

$$\frac{2 * Prcision * Recall}{Prcision + Recall}$$

III.4.5 Matrice de confusion

La matrice de confusion est utilisée pour décrire les performances d'un modèle de classification, le tableau III.1 présente une matrice de confusion dans le cas d'une classification binaire.

TABLE III.1: Matrice de Confusion pour une classification binaire

		Prédiction	
		Négative : 0	**Positive : 1**
Réalité	**Négative : 0**	**Vrai Négatif : TN**	**Faux Positif : FP**
	Positif : 1	**Faux Négatif : FN**	**Vrai Positif : TP**

III.5 Expériences

Pour créer un modèle qui détecte le contenu pornographique dans une vidéo, nous avons entrainé plusieurs modèles visuels et un modèle sonore. Par la suite, nous avons combiné le meilleur modèle de chaque catégorie pour créer notre modèle vidéo sur lequel nous avons effectué plusieurs évaluations.

III.5.1 Partie 1 : Modèle de détection des images

Les modèles pré-entraînés sont très utilisés pour l'extraction des caractéristiques visuelle. Nous avons tenté les expériences suivantes.

III.5.1.1 Expérience 1

Dans la première expérience, nous avons essayé de créer deux modèles InceptionV3 et ConvNeXtTiny :

Premier modèle : Dans ce modèle, nous avons utilisé le modèle pré-entraîné InceptionV3 et supprimé les couches de la classification de ce modèle avec l'attribut 'include-top = False'. Nous avons gelé les couches d'extraction visuelle avec l'attribut 'layer.trainable = False'. Par la suite, nous avons complété notre modèle avec les couches de la classification et utilisé une activation 'softmax' sur la couche de sortie (classification de 5 classes). Nous sommes passés à la compilation du modèle où nous avons utilisé un optimizer Adam avec un taux d'apprentissage descend sur chaque epoch, ensuite, nous avons divisé l'ensemble d'images de la dataset LSPD sur la partie d'entraînement et de test et de validation avec un code spécial et utilisé pour entraîner notre modèle et sur chaque epochs de l'entraînement, nous avons utilisé la fonction 'ImageDataGenerator' pour augmenter la dataset et obtenir un modèle plus robuste.
Nous avons utilisé un batch-size égal à 256 le nombre d'epochs égal à 6, taux d'apprentissage égal à 1e-1 dans le premier epoch, taux d'apprentissage égal à 8e-2 dans le deuxième epoch, taux d'apprentissage égal à 6e-2 dans le $3^{ème}$ epoch, taux d'apprentissage égal à 4e-2 dans le $4^{ème}$ epoch, taux d'apprentissage égal à 2e-2 dans le $5^{ème}$ epoch, taux d'apprentissage égal à 5e-3 dans le $6^{ème}$ epoch.
Après la fin d'entraînement, nous avons testé notre modèle sur la partie test de l'ensemble des images de la dataset LSPD et nous obtenions la matrice de confusion suivant III.1 et les performances suivant III.2.

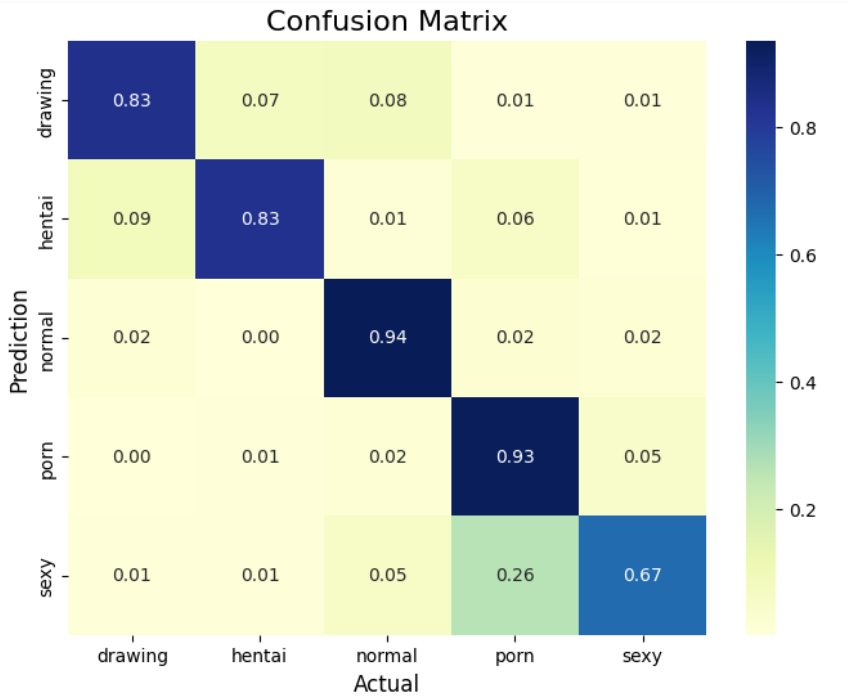


FIGURE III.1 – Matrice de confusion du premier modèle de l'expérience 1

TABLE III.2: Performances du premier modèle de l'expérience 1

performances	Accuracy	Precision	Recall	F1-Score
pourcentage	88.74	86	84	85

TABLE III.3: Performances du premier modèle de l'expérience 1 après l'entraînement des 20 derniers layers

performances	Accuracy	Precision	Recall	F1-Score
pourcentage	90.23	87	87	87

Dans un premier temps, nous avions obtenu une Accuracy faible, notre modèle confondait entre la classe sexy et la classe porno. Pour améliorer notre modèle, nous avons entraîné les 20 dernières couches sur l'ensemble d'images de la dataset LSPD avec l'attribut 'layer.trainable = True' et fixé le taux d'apprentissage à 1e-5. Ensuite, nous avons répété le premier test et nous avons obtenu les performances présentées dans le tableau III.3 et la matrice de confusion dans la Figure III.2.

Après l'entraînement des 20 derniers layers, nous avons observé une augmentation des performances et une diminution des erreurs entre la classe porno et sexy, témoignant d'une amélioration de l'accuracy.

Deuxième modèle : Pour ce deuxième modèle, nous avons utilisé les mêmes attributs que le premier modèle. Nous avons utilisé le modèle ConvNeXtTiny et supprimé les couches de la classification et gelé les couches de ce modèle sur les nouvelles données. Par la suite, nous avons ajouté les couches de la classification avec une sortie à 5 classes et ensuite, nous avons

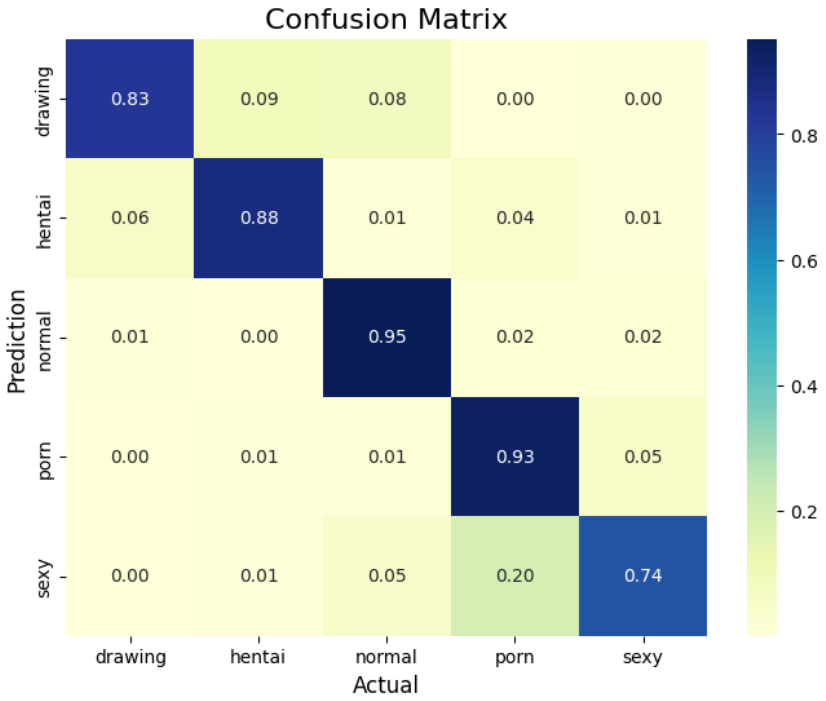


FIGURE III.2 – Matrice de confusion du premier modèle de l'expérience 1 après l'entraînement des 20 derniers layers

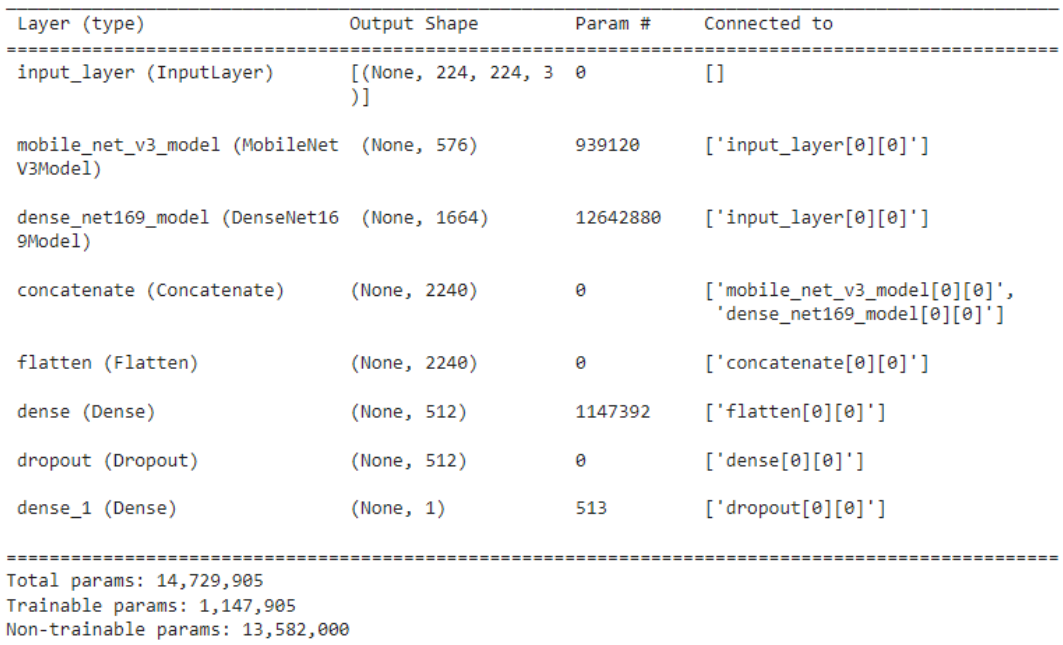
TABLE III.4: Performances du deuxième modèle de l'expérience 1

performances	Accuracy	Precision	Recall	F1-Score
pourcentage	74.42	73	64	66

utilisé un optimizer Adam et avec un taux d'apprentissage comme le premier modèle sur chaque epoch pour la compilation et entraîner ce modèle sur les mêmes données et utilisé la fonction 'ImageDataGenerator' pour augmenter ces données comme le premier modèle.
Nous avons utilisé un batch-size égal à 64 le nombre d'epochs égal à 6.
À la fin de l'entraînement, nous avons testé notre modèle sur la partie test de l'ensemble des images de la dataset LSPD comme le premier modèle et nous avons obtenu les performances présentées dans le tableau III.4.
Nous avons remarqué que l'Accuracy est très faible par rapport au premier modèle et toutes les autres performances sont très faibles. Nous avons donc exclu ce modèle de l'étude.

III.5.1.2 Expérience 2

Les études ont montré que la combinaison des modèles pré-entrainés permet une meilleure extraction des caractéristiques et résulte en de meilleures performances. Nous avons également remarqué que la classification binaire donne de meilleurs résultats que la classification multiclasse.



Layer (type)	Output Shape	Param #	Connected to
input_layer (InputLayer)	[(None, 224, 224, 3)]	0	[]
mobile_net_v3_model (MobileNetV3Model)	(None, 576)	939120	['input_layer[0][0]']
dense_net169_model (DenseNet169Model)	(None, 1664)	12642880	['input_layer[0][0]']
concatenate (Concatenate)	(None, 2240)	0	['mobile_net_v3_model[0][0]', 'dense_net169_model[0][0]']
flatten (Flatten)	(None, 2240)	0	['concatenate[0][0]']
dense (Dense)	(None, 512)	1147392	['flatten[0][0]']
dropout (Dropout)	(None, 512)	0	['dense[0][0]']
dense_1 (Dense)	(None, 1)	513	['dropout[0][0]']

Total params: 14,729,905
Trainable params: 1,147,905
Non-trainable params: 13,582,000

FIGURE III.3 – Détaille du premier modèle de la deuxième expérience

Premier modèle : En amont de la préparation de notre modèle, nous avons fait un petit pré-traitement sur notre ensemble d'images de la dataset LSPD où nous avons combiné les classes pour obtenir 2 classes (la classe drawing et normal dans la première classe, et la classe porno et hentai et sexy dans l'autre classe). Cela permet d'utiliser une classification binaire dans la couche de sortie. Après la préparation des données, nous avons préparé notre modèle en utilisant deux modèles pré-entraînés, le MobileNetV3Small et le DenseNet169, avec la suppression des couches de la classification pour chaque modèle et gel des couches de chaque modèle pour l'entraînement sur les nouvelles données. Ensuite, nous avons connecté la couche d'entrée avec les deux modèles pré-entraînés. Après ça, nous avons combiné les caractéristiques de chaque modèle et ajouté les couches de la classification à notre modèle avec une activation 'sigmoid' sur la couche de sortie de notre modèle. Pour finir, nous avons compilé notre modèle avec un optimizer Adam et un taux d'apprentissage égal à 0.001, la figure III.3 explique bien les détails de notre modèle.

Après la préparation et la compilation de notre modèle, nous l'avons entraîné sur l'ensemble des données que nous avons pré-traitées avec 10 epochs et un batch-size égal à 256. Nos résultats sont résumés dans la figure III.4 où le changement de l'Accuracy et le Loss sur chaque epochs sont expliqués.

Nous avons testé notre modèle sur les données pré-traitées après l'entraînement avec un threshold égal à 0.5 et nous avons obtenu la matrice de confusion présentée en Figure III.5 et les performances affichées sur le tableau III.5.

Après le premier test, nous avons créé un autre modèle aux mêmes valeurs d'attribut comme le premier, mais avec un petit changement sur les couches de la classification et utilisé le modèle

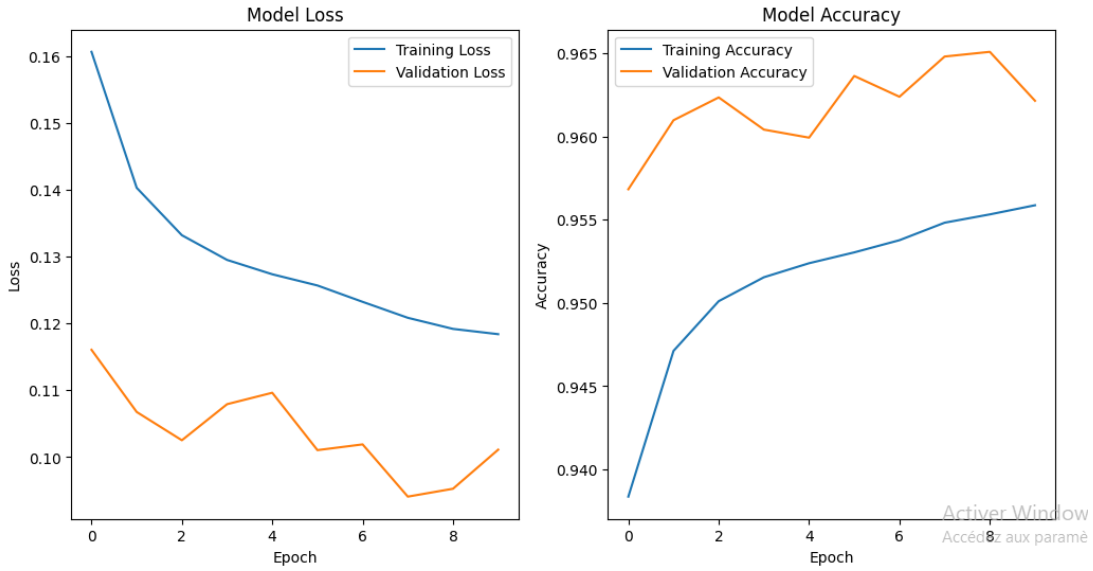


FIGURE III.4 – Résultats de l'entraînement du premier modèle de la deuxième expérience

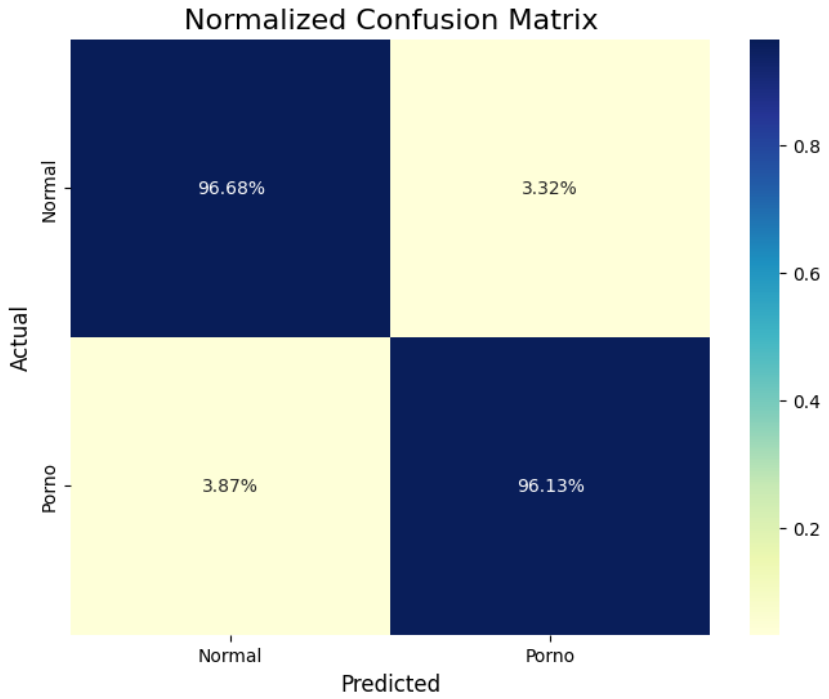


FIGURE III.5 – Matrice de confusion du premier modèle de l'expérience 2

TABLE III.5: Performances du premier modèle de l'expérience 2

performances	Accuracy	Precision	Recall	F1-Score
pourcentage	96	96	96	96

```
Layer (type)                 Output Shape              Param #   Connected to
==================================================================================================
input_layer (InputLayer)     [(None, 224, 224, 3       0         []
                             )]

MobilenetV3large (Functional)  (None, 960)             2996352   ['input_layer[0][0]']

densenet169 (Functional)     (None, 1664)              12642880  ['input_layer[0][0]']

concatenate (Concatenate)    (None, 2624)              0         ['MobilenetV3large[0][0]',
                                                                  'densenet169[0][0]']

flatten (Flatten)            (None, 2624)              0         ['concatenate[0][0]']

dense (Dense)                (None, 1024)              2688000   ['flatten[0][0]']

dropout (Dropout)            (None, 1024)              0         ['dense[0][0]']

dense2 (Dense)               (None, 512)               524800    ['dropout[0][0]']

dropout2 (Dropout)           (None, 512)               0         ['dense2[0][0]']

output (Dense)               (None, 1)                 513       ['dropout2[0][0]']

==================================================================================================
Total params: 18,852,545
Trainable params: 3,213,313
Non-trainable params: 15,639,232
__________________________________________________________________________________________________
```

FIGURE III.6 – Détaille du premier modèle de la deuxième expérience (partie 2)

TABLE III.6: Performances du premier modèle de l'expérience 2 (partie 2)

performances	Accuracy	Precision	Recall	F1-Score
pourcentage	96.33	96.24	96.10	96.17

TABLE III.7: Performances du premier modèle de l'expérience 2 (partie 3)

performances	Accuracy	Precision	Recall	F1-Score
pourcentage	96.47	96.48	96.47	96.47

MobileNetV3Large à la place de MobileNetV3Small comme l'explique la figure III.6. Nous avons également changé le taux d'apprentissage à 0.0001, et nous avons entraîné notre modèle sur les mêmes données que le premier modèle avec 10 epochs et un batch-size de 256. Nous avons testé notre modèle sur les données pré-traitées après l'entraînement avec un threshold égal à 0.5 et nous avons obtenu la matrice de confusion présentée en Figure III.7 et les performances présentées dans le tableau III.6.

Nous avons remarqué une légère amélioration des résultats et avons poursuivi l'entraînement depuis la $10^{ème}$ epoch jusqu'à $30^{ème}$ epoch et avons changé le batch-size à 64 et le threshold à 0.57. Cette valeur de threshold qui a été obtenue suite à un test itératif sur toutes les valeurs 0 à 1 avec un pas de 0.01 nous a permis d'obtenir les meilleurs résultats. Nos résultats sont illustrés par la matrice de confusion en Figure III.8 et les performances sont affichées dans le tableau III.7.

Deuxième modèle : Dans un premier temps, nous avons utilisé le dataset LSPD image 2 classes.

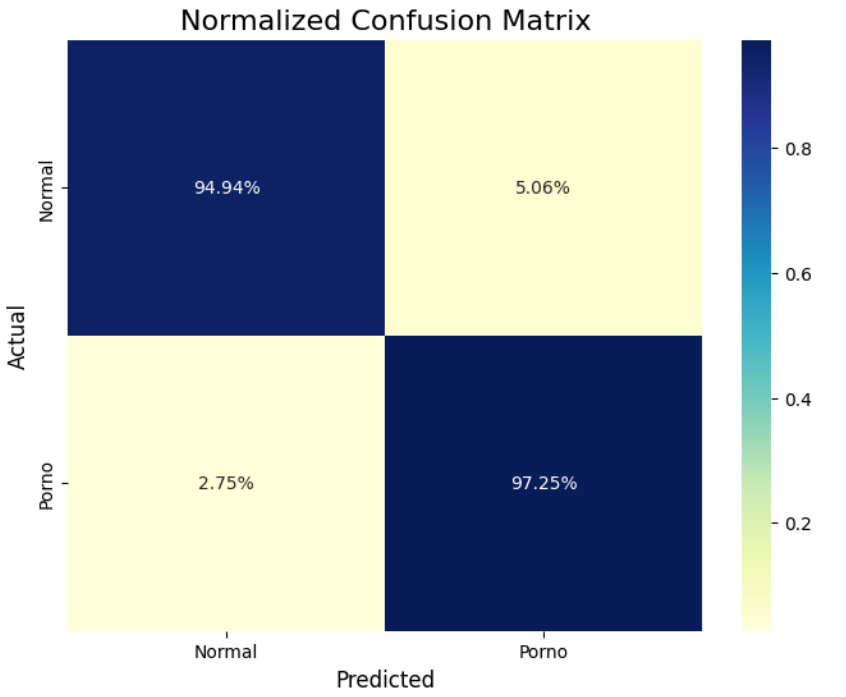


FIGURE III.7 – Matrice de confusion du premier modèle de l'expérience 2 (partie 2)

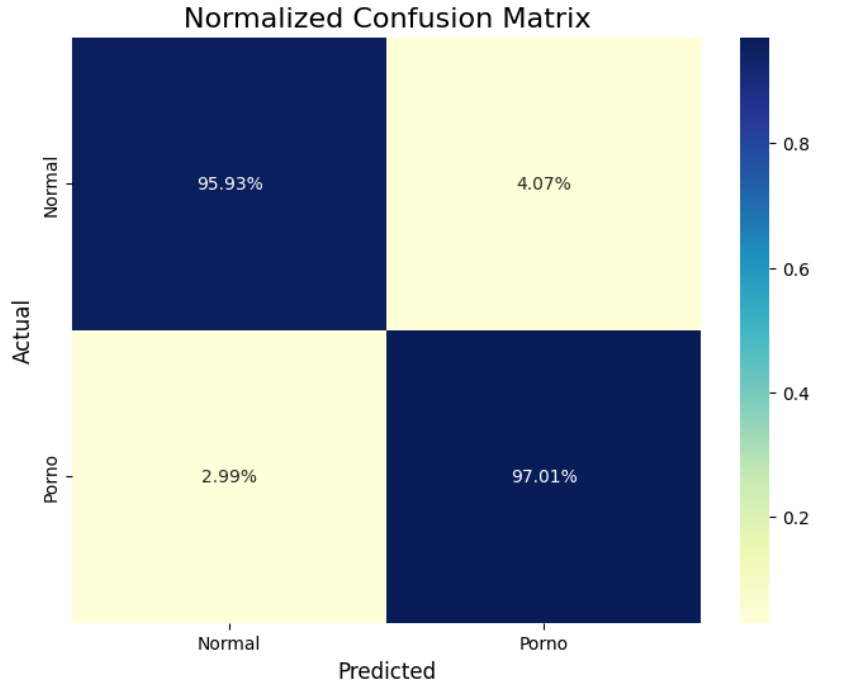


FIGURE III.8 – Matrice de confusion du premier modèle de l'expérience 2 (partie 3)

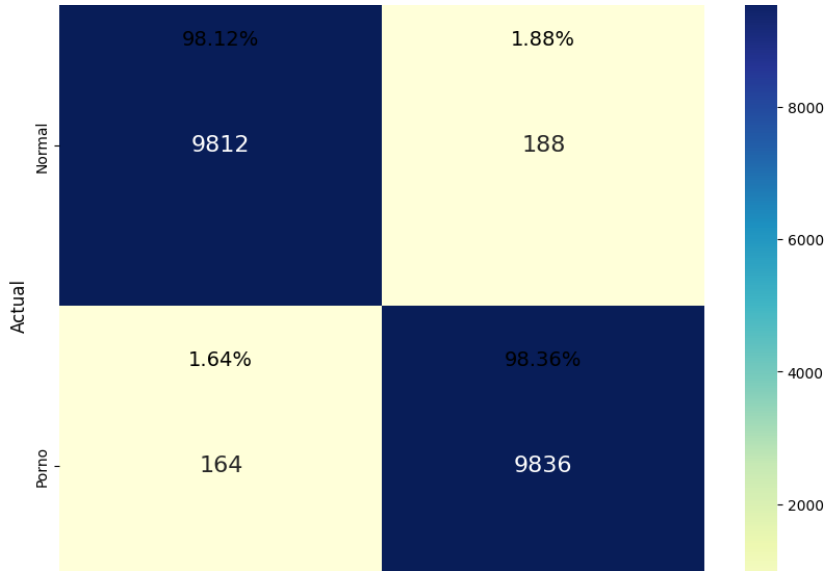


FIGURE III.9 – Matrice de confusion du deuxième modèle de l'expérience 2

TABLE III.8: Performances du deuxième modèle de l'expérience 2

performances	Accuracy	Precision	Recall	F1-Score
pourcentage	98.24	98.24	98.24	98.24

TABLE III.9: Performances du premier test du modèle 2 de l'expérience 2 sur les images extraite depuis les vidéos

performances	Accuracy	Precision	Recall	F1-Score
pourcentage	80.77	80.66	81.73	82.5

Nous avons créé notre modèle en combinant 3 modèles pré-entrainés, le MobileNetV3Large et le DenseNet169 et l'InceptionV3 avec la suppression des couches de la classification pour chaque modèle et gel des couches de chaque modèle. Ensuite, nous avons connecté la couche d'entrée avec les trois modèles pré-entraînés. Après ça, nous avons combiné les caractéristiques de chaque modèle et ajouté les couches de la classification à notre modèle. Les détails de notre modèle sont résumés dans la figure II.2.

Lors de la compilation de notre modèle, nous avons utilisé un optimizer Adam et un taux d'apprentissage variable selon le numéro d'epoch. Notre modèle a été entraîné sur les données que nous avons préparées au début du travail avec 12 epochs et un batch-size égal à 128, taux d'apprentissage égal à 0.001 sur les deux premiers epoch, taux d'apprentissage égal à 5e-4 sur l'epoch 3 et 4, taux d'apprentissage égale à 1e-4 depuis la $5^{ème}$ epoch jusqu'à la $10^{ème}$ epoch, taux d'apprentissage égale à 1e-5 sur l'epoch 11 et 12.

Nous avons testé notre modèle sur les données pré-traitées après l'entraînement avec un threshold égal à 0.5 et nous avons obtenu la matrice de confusion présentée en Figure III.9 et les performances affichées sur le tableau III.8.

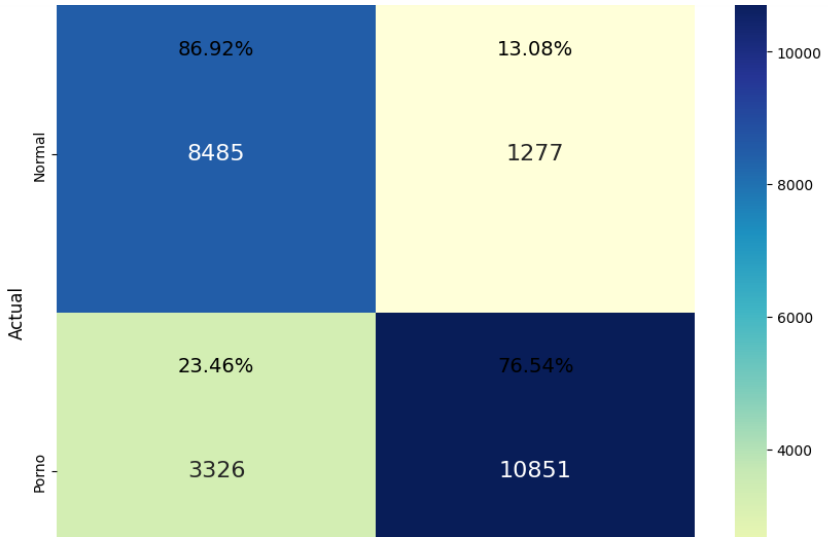


FIGURE III.10 – Matrice de confusion du premier test du modèle 2 de l'expérience 2 sur les images extraire depuis les vidéos

TABLE III.10: Performances du deuxième test du modèle 2 de l'expérience 2 sur les images extraire depuis les vidéos

performances	Accuracy	Precision	Recall	F1-Score
pourcentage	93.5	93.18	93.42	94.48

III.5.1.3 Expérience 3

Après les précédents tests, nous avons voulu tester notre modèle sur les dataset LSPD images extraites des vidéos. Les résultats sont présentés dans la matrice de confusion III.10 et les performances sont affichées dans le tableau III.9.
Nous avons remarqué une différence significative des performances de ce modèle entre l'expérience 2 et 3, cela s'explique par la différence des données. En effet, les images extraites des vidéos sont différentes des images prises en tant qu'images. Ces différences incluent principalement une différence de qualité, un manque de netteté et la présence accrue de flou de mouvements dans les images extraites des vidéos.

Pour mieux adapter notre modèle à la prédiction des images extraites à partir de vidéos, nou avons ré-enrainé le même modèle sur les dataset LSPD images extraites de vidéos sur 15 epochs en utilisant un optimizer Adam et un taux d'apprentissage égal à 1e-4 lors de la compilation et un batch-size égal à 256. À la fin de l'entraînement de notre modèle, nous avons relancé encore le même test avec le même threshold et nous avons obtenu une bonne amélioration par rapport au premier test. Nos résultats sont présentés dans la matrice de confusion III.11 et les performances dans le tableau III.10.

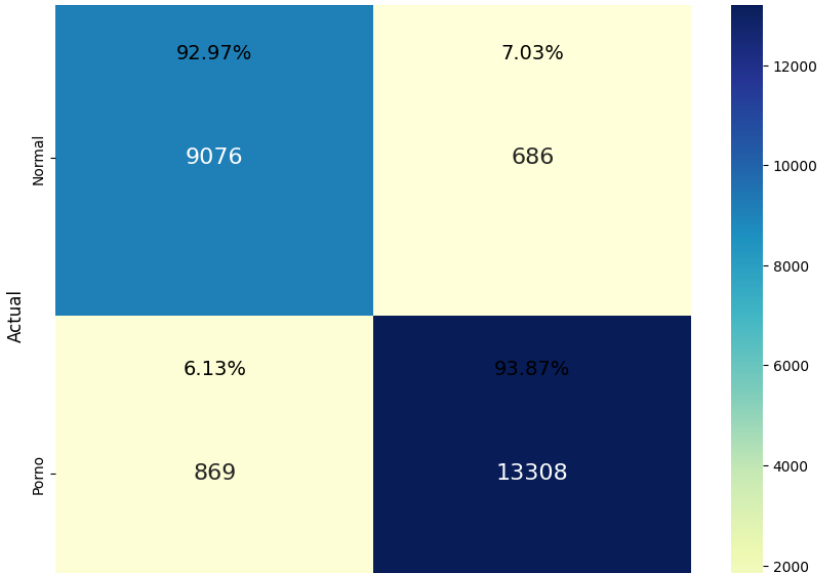


FIGURE III.11 – Matrice de confusion du deuxième test du modèle 2 de l'expérience 2 sur les images extraire depuis les vidéos

TABLE III.11: Performances du premier test du modèle de l'expérience 4

performances	Accuracy	Precision	Recall	F1-Score
pourcentage	88.36	87.86	88.92	89.72

III.5.2 Partie 2 : Modèle de détection de son

Pour créer le modèle sonore, nous avons utilisé un modèle pré-entrainer.

III.5.2.1 Expérience 4

Dans cette expérience, nous avons utilisé un seul modèle pré-entrainer, le VGGish et supprimer les couches de la classification et gelé les couches d'extraction de caractéristiques. Par la suite, nous avons complété notre modèle par les couches de la classification avec une activation 'sigmoid' pour une classification binaire.
la figure II.4 explique l'architecture de notre modèle.
Nous avons entraîné ce modèle pendant 20 epochs sur le dataset LSPD audio, après l'avoir compilé avec un optimizer Adam et un taux d'apprentissage égal à '1e-4' et un batch-size égal à 32 et le threshold à 0.65. Cette valeur de threshold qui a été obtenue suite à un test itératif sur toutes les valeurs 0 à 1 avec un pas de 0.01 nous a permis d'obtenir les meilleurs résultats. Nos résultats sont illustrés par la matrice de confusion en Figure III.12 et les performances sont affichées dans le tableau III.11.
Dans le premier test, nous avons obtenu des résultats faibles donc nous avons continué l'entrainement pour 20 autres epochs sur les mêmes données en changeant le taux d'apprentissage à '1e-5' et en dégelant les quatre dernières couches du modèle pré-entrainé. Après la fin de l'entrainement, nous avons re-testé notre modèle sur les mêmes données avec le même threshold et nous avons

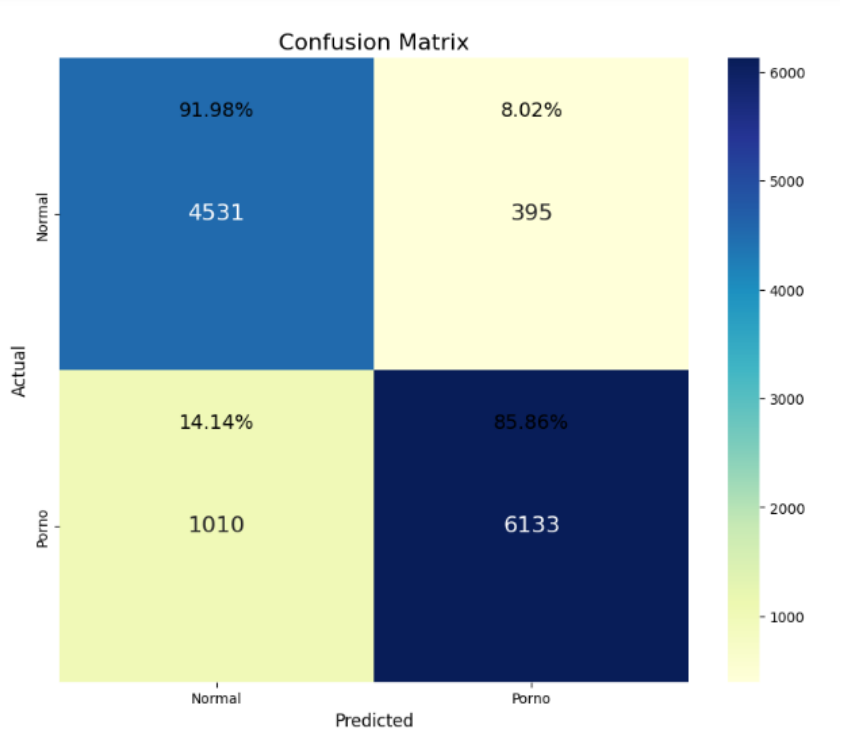


FIGURE III.12 – Matrice de confusion du premier test du modèle de l'expérience 4

TABLE III.12: Performances du deuxième test du modèle de l'expérience 4

performances	Accuracy	Precision	Recall	F1-Score
pourcentage	95.02	94.79	94.93	95.78

remarqué une amélioration sur les résultats comme le montre la matrice de confusion en Figure III.13 et le tableau des performances (Table III.12).

III.5.3 Partie 3 : Modèle de détection de vidéo

Le modèle vidéo a été créé en combinant les deux meilleurs modèles créés dans les parties précédentes (le premier modèle détectant les images porno et le deuxième détectant les sons porno).

III.5.3.1 Expérience 5 : tests sur les vidéos

Pour évaluer notre modèle, nous avons suivi l'architecture synthétisée dans la figure III.14. Nous avons réalisé des évaluations sur trois datasets : LSPD vidéo test, NPDI-2K test et notre dataset.

— **Dataset NPDI-2K test :** l'évaluation sur ce dataset nous permet de comparer notre modèle avec ceux des travaux précédents. En effet, ce dataset est le plus utilisé et nous permet de mieux situé notre modèle par rapport aux autres.

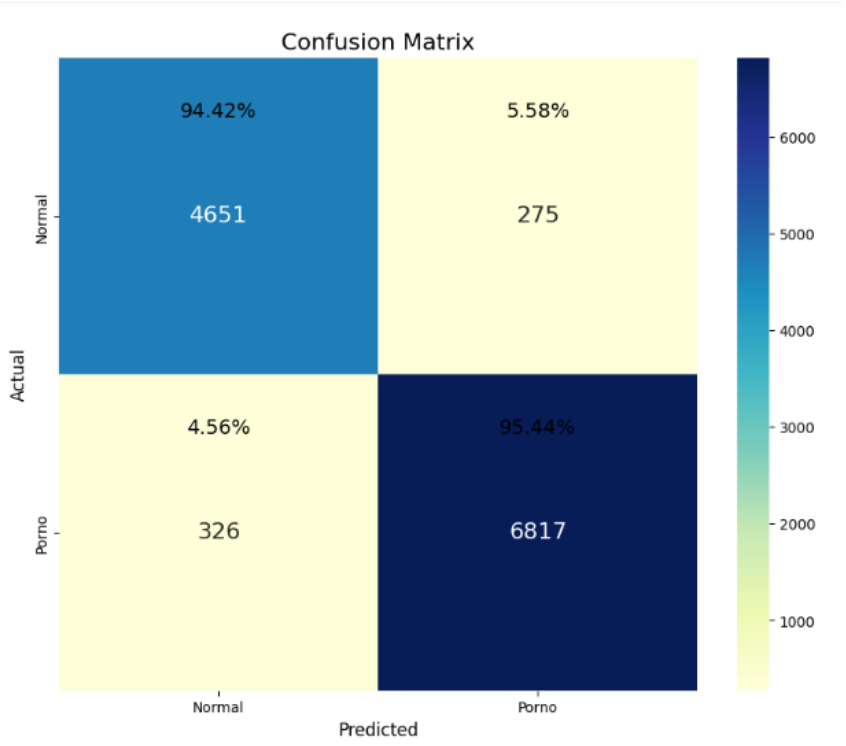


FIGURE III.13 – Matrice de confusion du deuxième test du modèle de l'expérience 4

Ce dataset contient certaines vidéos dont le son et l'image ne sont pas synchronisés, ce qui permet de mieux tester le modèle.

— **Dataset LSPD vidéo test :** l'évaluation sur ce dataset permettra aux futures études de se comparer à ce modèle. Ce dataset est le plus récent et sera sûrement le plus utilisé dans les années à venir, il est judicieux d'effectuer des tests sur ce dataset.

— **Notre dataset :** l'évaluation sur ce dataset permet de connaitre les performances de notre approche dans un cas réel. Ce modèle a été créé dans le but d'automatiser la modération de contenu chez l'EPTV, tester ce modèle sur le contenu de cette chaine est une étape clé de notre projet et reflète les performances de notre approche dans un cas réel.

Finalement, une évaluation générale est faite en combinant les résultats des précédentes évaluations, ça permet d'avoir une meilleure compréhension des performances du modèle grâce aux tests réalisé sur des données variées.

Le modèle a été exécuté avec un threshold de l'image égal à 0.9 et un threshold du son égal à 0.98. Ces valeurs ont été choisies en effectuant des tests manuels. La réalisation des mêmes tests itératifs que ceux des parties précédentes auraient été trop chronophages.
Les performances du test des deux modèles sur les trois dataset sont présentées dans le tableau III.13.

Les matrices de confusion des tests du modèle visuel sont présentées dans les figures III.15, III.16, III.17 dans les dataset LSPD, NPDI-2K, notre dataset respectivement.

Les matrices de confusion des tests du modèle sonore sont présentées dans les figures III.18,

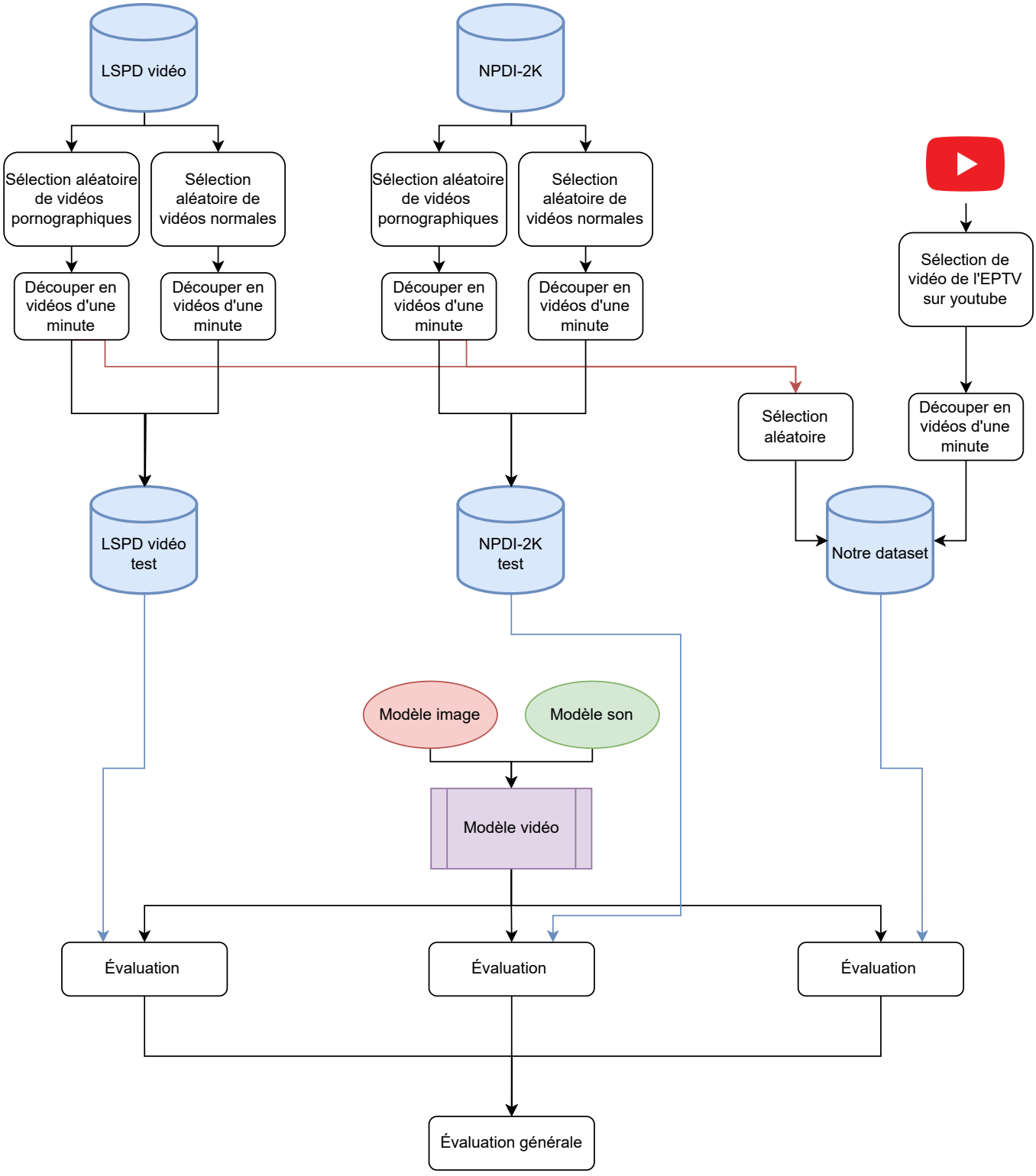


FIGURE III.14 – Architecture des tests du modèle vidéo

TABLE III.13: Performances du test des deux modèles (images et son) sur les trois dataset

Dataset	Type	Accuracy	Précision	Recall	F1-Score
LSPD	Images	99.1	99.25	99.88	99.25
	Son	82.35	84.77	85.23	82.67
	Union	99.1	99.25	99.88	99.25
	Intersection	87.33	88.03	89.39	88.14
NPDI-2K	Images	98.32	98.38	98.1	98.62
	Son	88.23	88.17	87.02	90.51
	Union	91.46	93.68	89.24	93.4
	Intersection	95.08	94.46	95.88	95.78
Notre dataset	Images	99.43	99.47	99.39	99.47
	Son	93.18	93.46	93.51	93.22
	Union	98.86	98.95	98.79	98.94
	Intersection	96.02	96.09	96.26	96.11
Total	Total Union	94.65	95.76	93.59	95.62
	Total Intersection	94.06	93.73	94.89	94.65

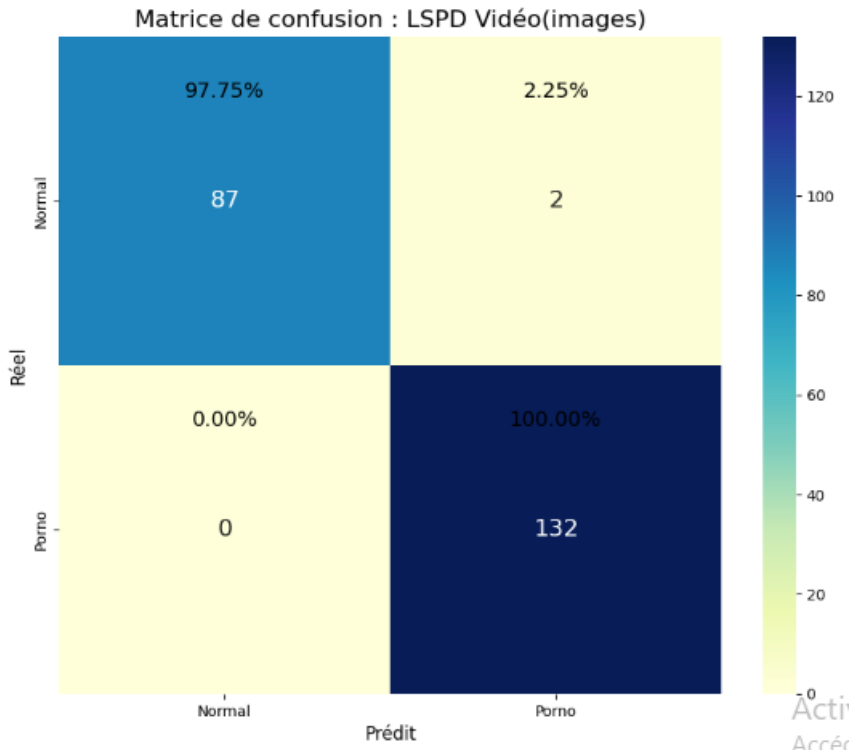


FIGURE III.15 – Matrice de confusion du test du modèle visuel sur les vidéos de LSPD

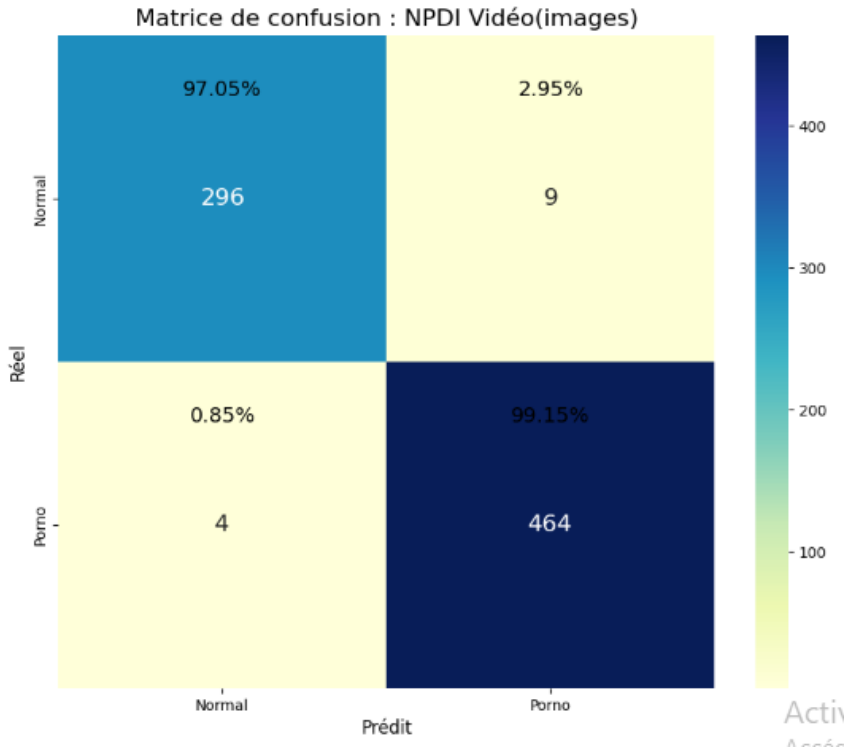


FIGURE III.16 – Matrice de confusion du test du modèle visuel sur les vidéos de NPDI-2K

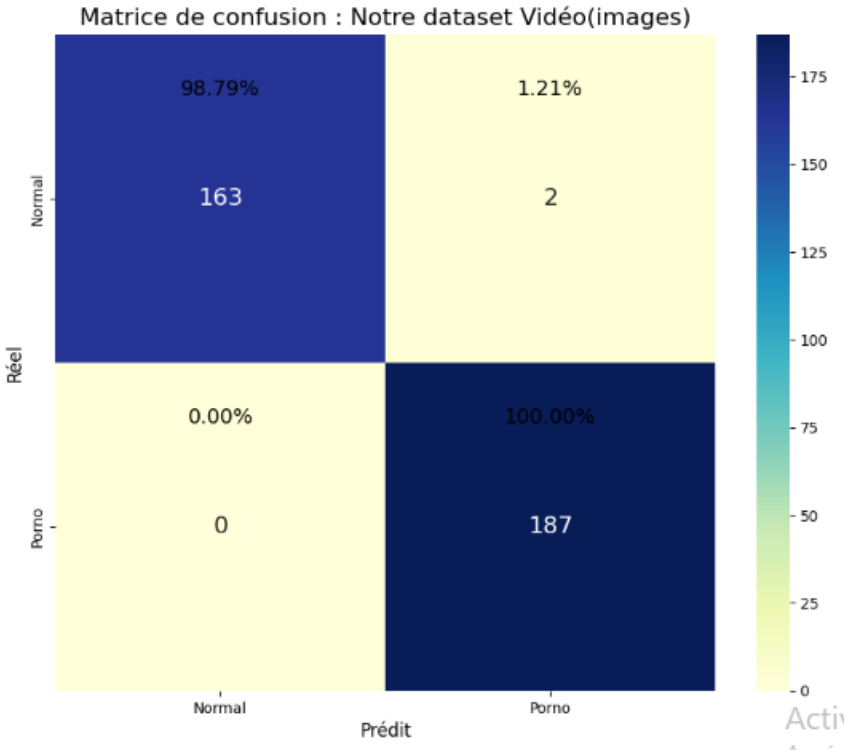


FIGURE III.17 – Matrice de confusion du test du modèle visuel sur les vidéos de notre dataset

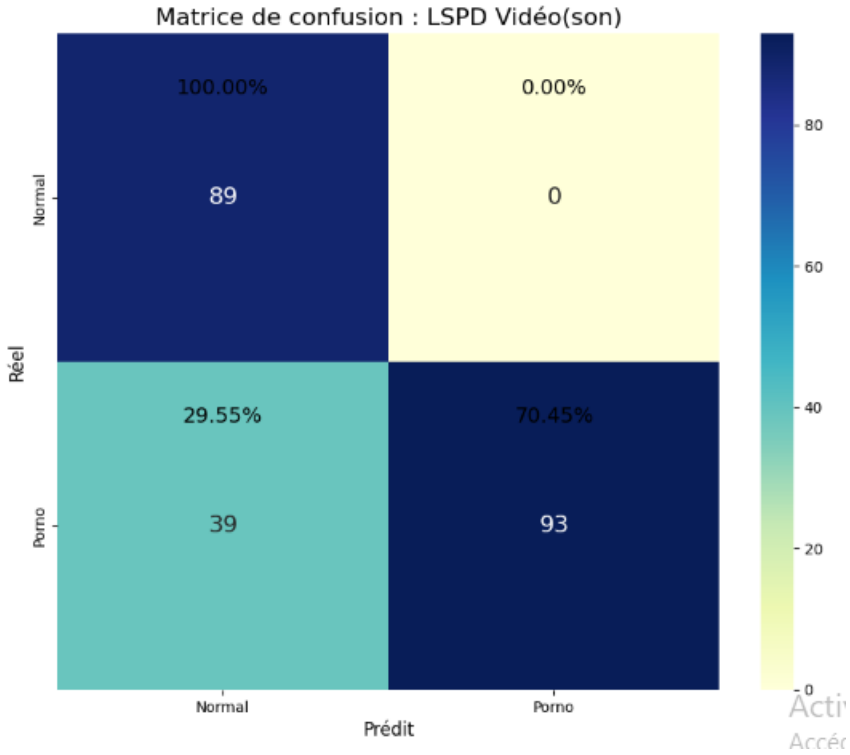


FIGURE III.18 – Matrice de confusion du test du modèle visuel sur les vidéos de LSPD

III.19, III.20 dans les dataset LSPD, NPDI-2K, notre dataset respectivement.

Les matrices de confusion des tests du modèle union entre le modèle visuel et le modèle sonore sont présentées dans les figures III.21, III.22, III.23 dans les dataset LSPD, NPDI-2K, notre dataset respectivement.

Les matrices de confusion des tests du modèle intersection entre le modèle visuel et le modèle sonore sont présentées dans les figures III.24, III.25, III.26 dans les dataset LSPD, NPDI-2K, notre dataset respectivement.

Si le modèle visuel a présenté des accuracy moyennant les 98.95% pour les 3 dataset, le modèle sonore quant à lui s'est montré moins performant pour la détection de contenu pornographique avec une moyenne d'accuracy de 87.92%. Le manque d'efficacité du modèle sonore sur le dataset LSPD s'explique par le manque d'audios dans les vidéos de ce dataset. Par ailleurs, le manque d'efficacité du modèle sonore dans le dataset NPDI-2K s'explique par la présence de certaines vidéos trompeuses dont le son et l'image ne sont pas synchronisés. Le modèle vidéo union s'est montré plus performant que le modèle vidéo intersection dans les dataset LSPD et légèrement plus performant dans notre dataset, tandis que dans la dataset NPDI-2K, l'intersection a mieux performé que l'union. Cela signifie que les performances du modèle vidéo union/intersection dépend du contenu. Nous remarquons que le son limite les performances de l'union et de l'intersection en raison de son efficacité moindre.

Les matrices de confusion suivantes III.27, III.28 et le tableau des performances (Table III.13) détaillent nos résultats.

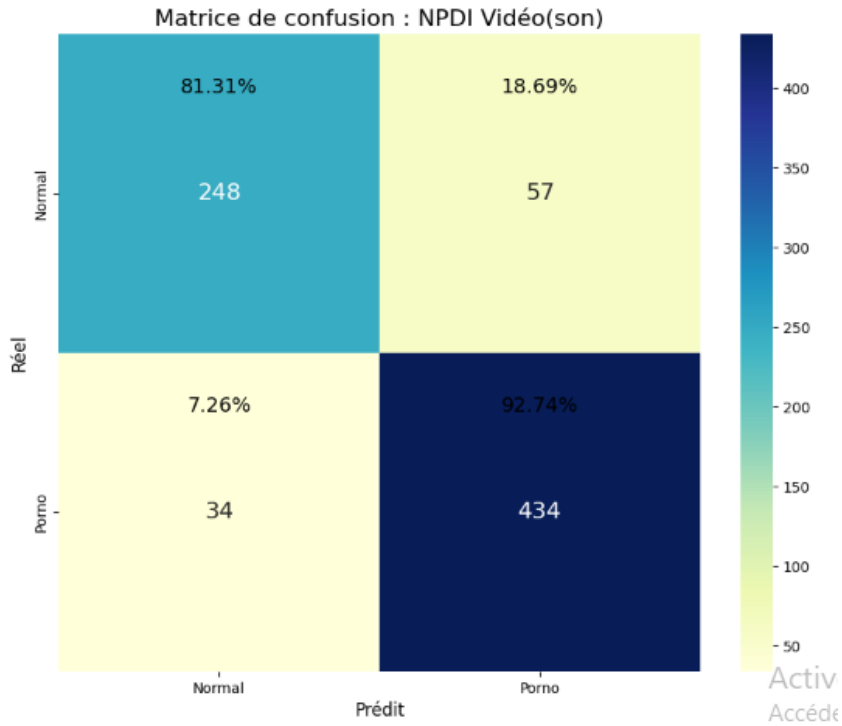


FIGURE III.19 – Matrice de confusion du test du modèle visuel sur les vidéos de NPDI-2K

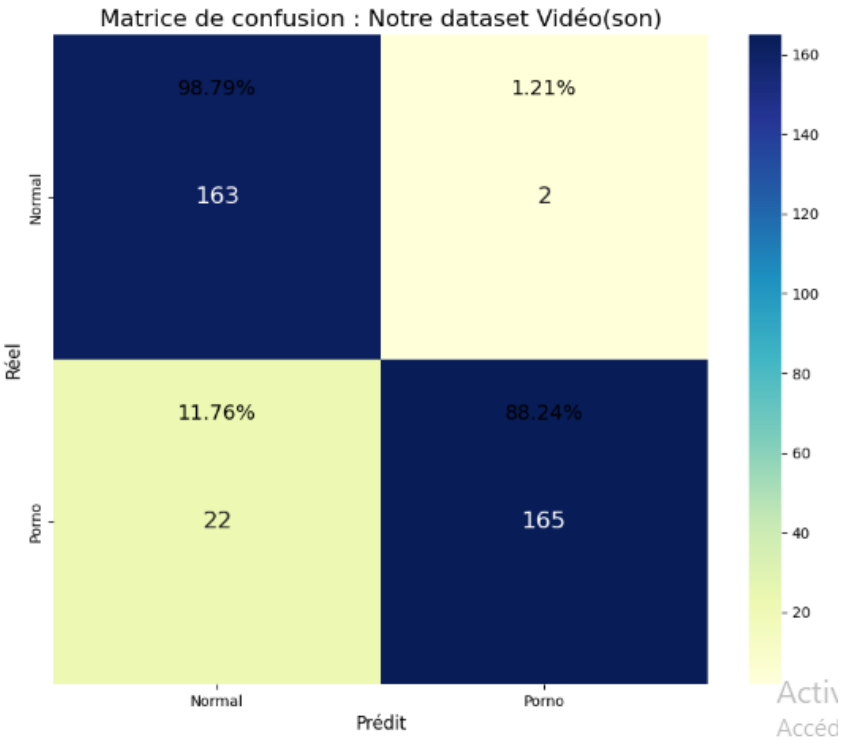


FIGURE III.20 – Matrice de confusion du test du modèle visuel sur les vidéos de notre dataset

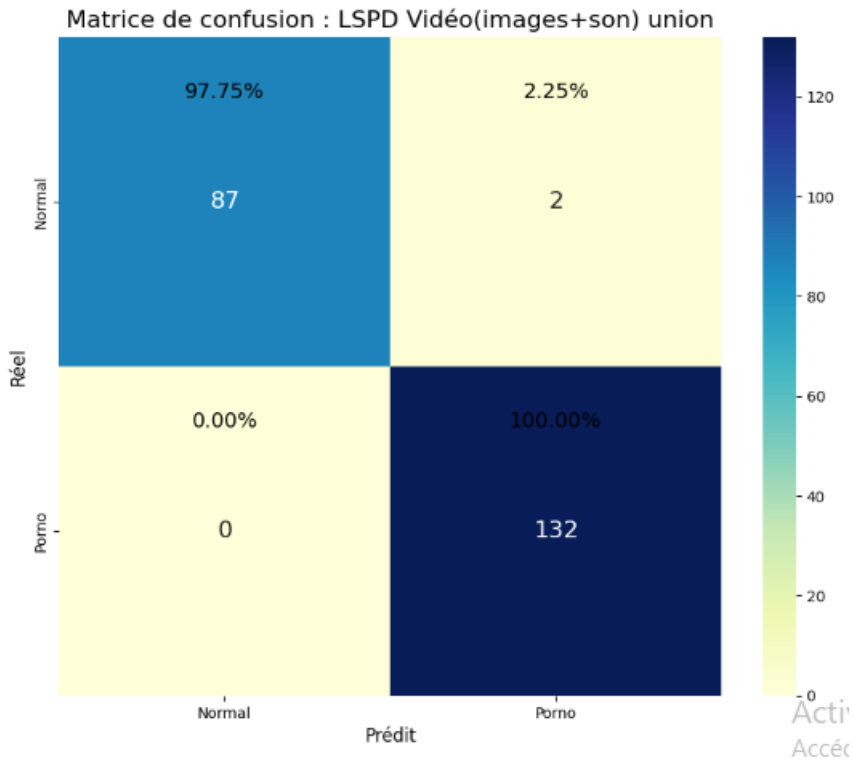


FIGURE III.21 – Matrice de confusion du test du modèle Union entre modèle visuel et sonore sur les vidéos de LSPD

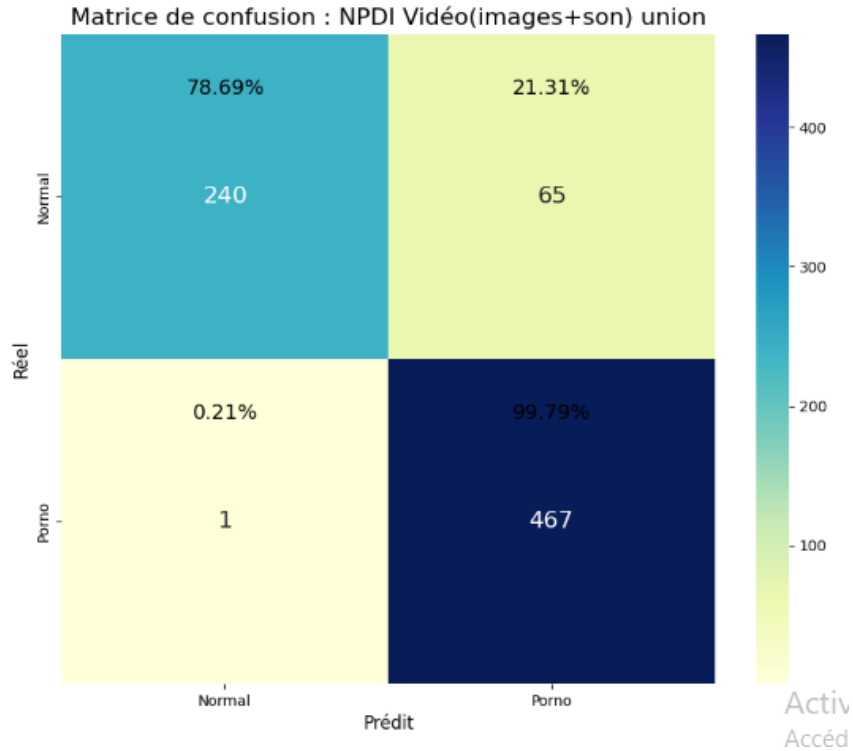


FIGURE III.22 – Matrice de confusion du test du modèle Union entre modèle visuel et sonore sur les vidéos de NPDI-2K

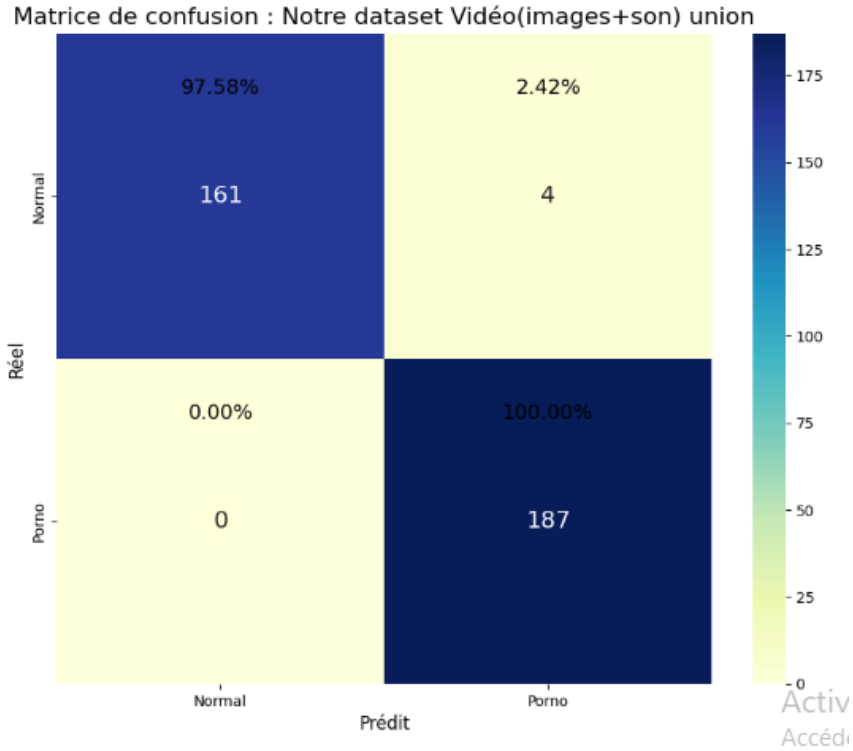


FIGURE III.23 – Matrice de confusion du test du modèle Union entre modèle visuel et sonore sur les vidéos de notre dataset

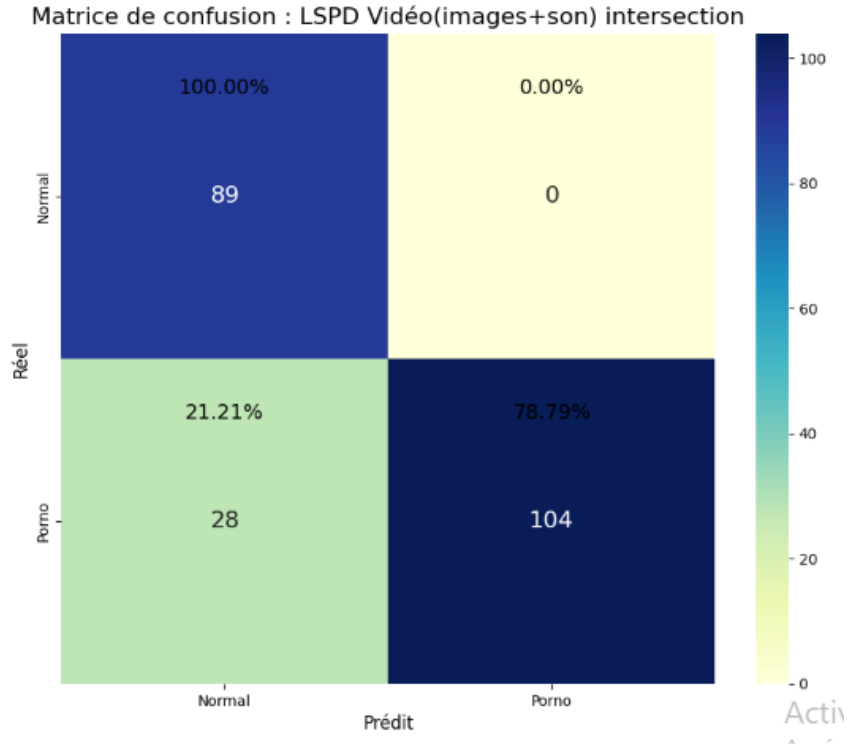


FIGURE III.24 – Matrice de confusion du test du modèle Intersection entre modèle visuel et sonore sur les vidéos de LSPD

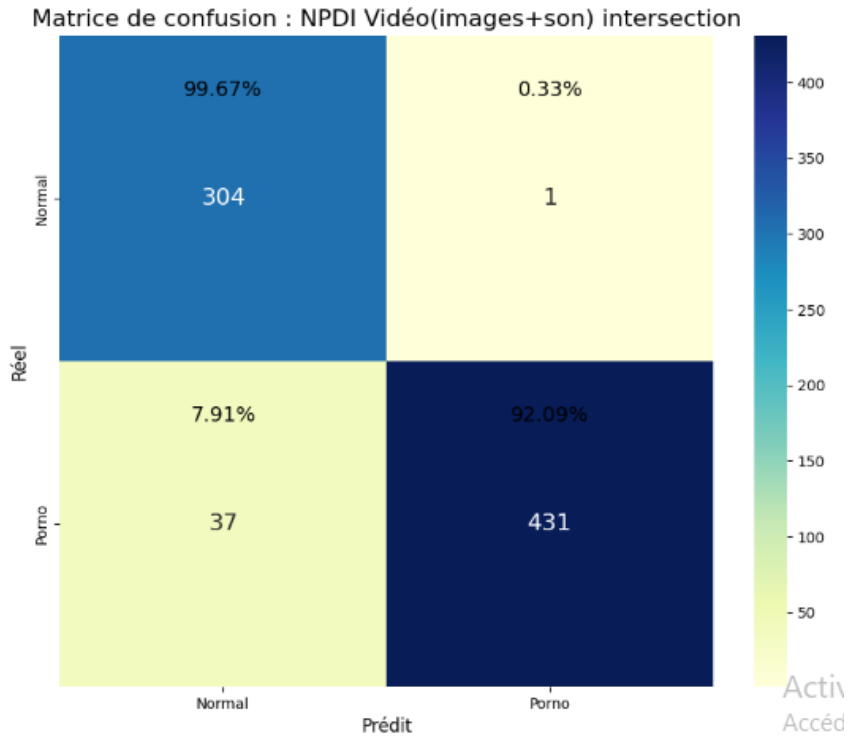


FIGURE III.25 – Matrice de confusion du test du modèle Intersection entre modèle visuel et sonore sur les vidéos de NPDI-2K

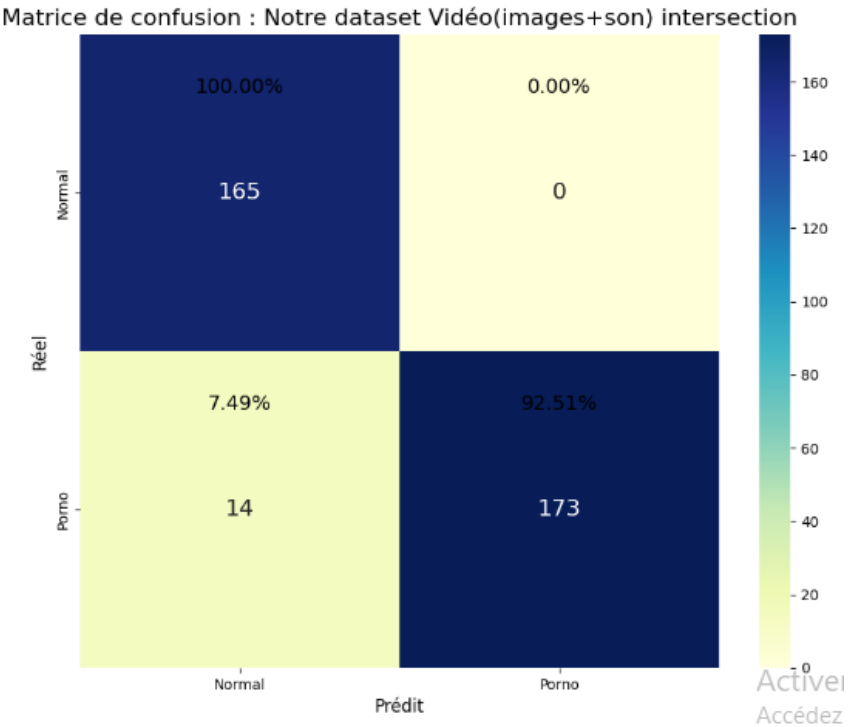


FIGURE III.26 – Matrice de confusion du test du modèle Intersection entre modèle visuel et sonore sur les vidéos de notre dataset

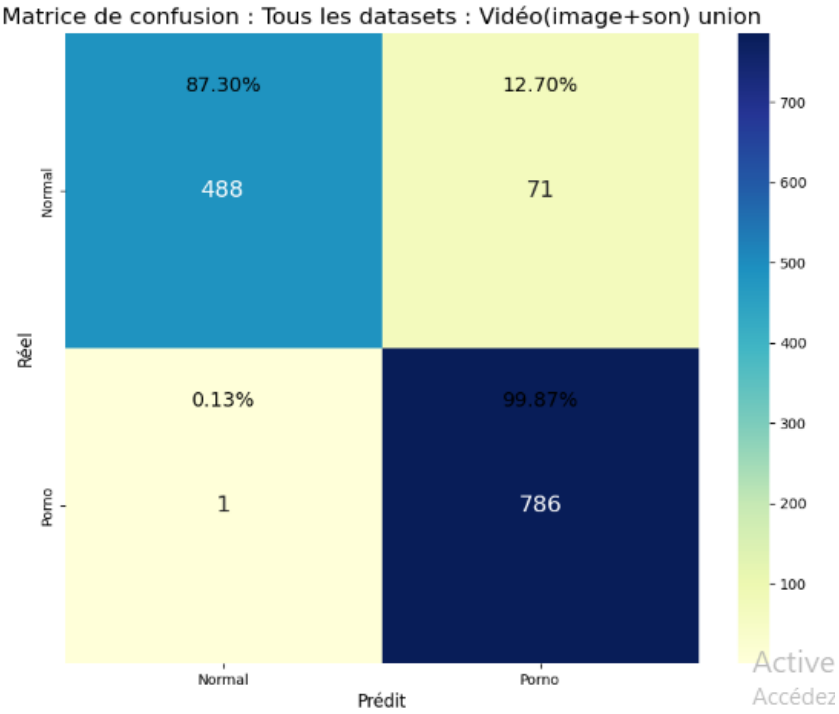


FIGURE III.27 – Matrice de confusion du test du total Union sur total des dataset

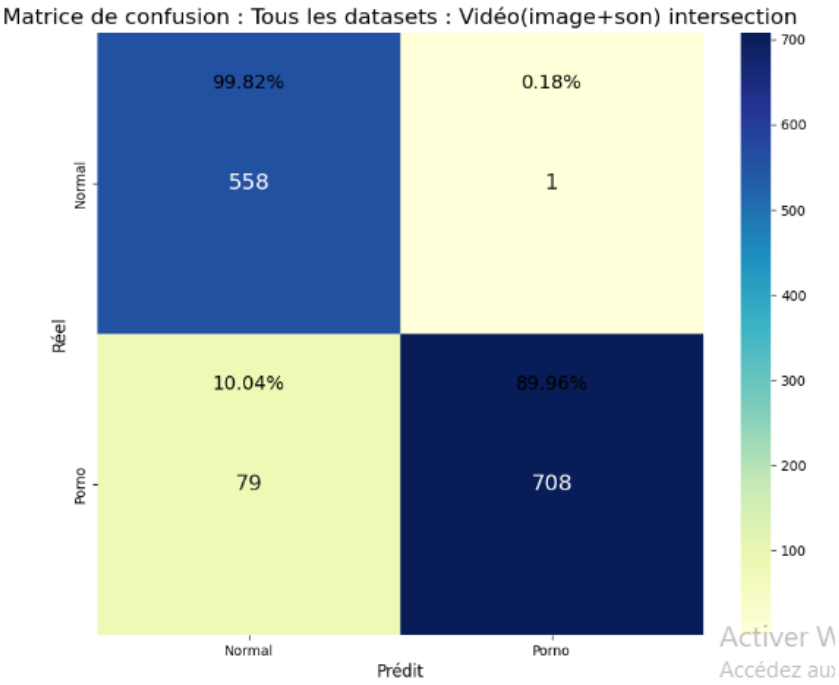


FIGURE III.28 – Matrice de confusion du test du total Intersection sur total des dataset

TABLE III.14: Comparaison de nos méthodes avec les travaux précédents sur le dataset NPDI-2K

Étude	Année	Détection par	Technique	Accuracy
Notre modèle (images)	2024	vidéo	DenseNet169 + MobileNetV3Large + InceptionV3 ré-entrainé	98,32%
[18]	2023	vidéo	CNN + ResNet-18	97,15%
[13]	2022	vidéo	ResNet101 + DenseNet121 et calcul de similitude (CNN + ViSiL)	96,88%
[19]	2023	vidéo	CNN a deux flux	95,20%
Notre modèle (Intersection)	2024	vidéo (images + son)	DenseNet169 + MobileNetV3Large + InceptionV3 ré-entrainé / VGGish ré-entrainé	95,08%
[16]	2020	vidéo (images+son)	AudioVGG et InceptionV3 + LSTM	94,00%
[33]	2020	vidéo (images+son)	VGG-13 + LSTM + CNN sur l'échelle de Mel	92,33%
Notre modèle (Union)	2024	vidéo (images + son)	DenseNet169 + MobileNetV3Large + InceptionV3 ré-entrainé / VGGish ré-entrainé	91,46%
Notre modèle (Son)	2024	vidéo (son)	VGGish ré-entrainé	88,23%

III.6 Comparaison

Le tableau III.14 compare les différentes méthodes et modèles que nous avons utilisés avec les travaux précédents, les tests ont été effectués sur le dataset NPDI-2k.

Nous remarquons que notre modèle vidéo (images) a obtenu de meilleurs résultats par rapport aux travaux utilisant une détection par vidéo (images seulement) avec une Accuracy de 98,32%. Notre modèle (intersection) qui utilise les images et le son a obtenu une Accuracy de 95,08%, c'est le meilleur résultat dans ce contexte. Notre modèle (union) a obtenu une Accuracy de 91,46%, cette différence s'explique par le manque d'efficacité du modèle son sur ce dataset qui a obtenu 88,26%. Ce manque peut être dû à la complexité du son dans ce dataset, en effet, nous retrouvons des vidéos pornographiques ayant un son non pornographique, cela est fait exprès sur certaines vidéos afin de mieux tester les performances des modèles.

Nous avons aussi effectué des tests sur notre modèle image sur le dataset LSPD (images), les tests ont été effectués sur les classes normales et pornographiques seulement. Nous avons obtenu une Accuracy de 98,24%. Ce test a été effectué sur notre modèle image avant son ré-entrainement sur les images extraites des vidéos du dataset LSPD. Nous remarquons une nette progression par rapport aux travaux précédents, cela est dû au manque de travaux sur ce dataset. Le tableau III.15 montre les résultats de la comparaison.

Le tableau III.16 compare les études sur la détection par son avec notre travail, en absence de test sur le même dataset, nous ne pouvons pas faire de comparaison équitable.

TABLE III.15: Comparaison avec les études utilisant le dataset LSPD image

Étude	Année	Classification	Technique	Précision
Notre modèle (image)	2024	Porno/non-porno	DenseNet169 + MobileNetV3Large + InceptionV3 ré-entrainé	98,24%
[2]	2023	Porno / normal/ sexy/ hentai/ dessin	ConvNexT(tiny) ré-entraîné	94,90%
[14]	2022	Porno/non-porno	CNN entraîné sur un autre dataset	79,02%

TABLE III.16: Comparaison avec les études sur la détection par son

Étude	Année	Technique	Classification	Précision	Dataset
[36]	2022	ResNet18 + Log Filter Banks	Porno/Non-porno	97,19% sur dataset privé	Dataset privé
[26]	2022	CNN entraîner sur un spectre Log Mel	Porno/Non-porno	94,89% sur NPDI-800	NPDI-800[7]
Notre modèle (son)	2024	VGGish ré-entrainé	Porno/Non-porno	93,18% sur notre dataset	LSPD audio

III.7 Conclusion

En conclusion, ce chapitre a permis de mettre en lumière les différentes étapes de l'implémentation et de l'évaluation de notre modèle.
Les diverses expériences menées ont montré l'efficacité des modèles à détecter la pornographie et la nudité sur diverse supports multimédias. Nous avons créé un Notebook [3] sur Google Colab permettant de tester notre modèle vidéo aisément.
Bien que certains défis subsistent, les résultats obtenus, mesurés par l'Accuracy, la précision, le Recall et le F1-score, montrent une bonne performance de notre modèle, avec des pistes d'amélioration pour des travaux futurs.
La comparaison finale a permis de situer notre approche par rapport à d'autres méthodes existantes, confirmant ainsi la pertinence et l'efficacité de notre travail.

3. https://colab.research.google.com/drive/1APZ3DeBlRRzZiNcPadAswbGyxw5rFflC

Conclusion et perspectives

Cette recherche répond à un besoin critique de l'EPTV, confronté à la tâche complexe de modérer efficacement un flux constant de contenu multimédia varié et potentiellement inapproprié.

Dans notre étude, nous avons exploré une approche novatrice pour la détection de la pornographie et de la nudité en utilisant une combinaison de modèles CNN pré-entraînés et une analyse multimodale combinant les informations visuelles et sonores grâce à l'utilisation de plusieurs modèles CNN pré-entrainés tels que InceptionV3, MobileNetV3-Large, DenseNet169 et VGGish.

Nous avons utilisé le dataset LSPD, qui est le dataset le plus récent et celui ayant le plus d'images et de vidéos. Nous avons également créé des datasets pour l'entrainement, l'évaluation et les tests. À travers les expériences et les tests réalisés sur les différents datasets, nous avons obtenu une Accuracy de 98,32% sur le dataset NPDI-2K, ce qui constitue le meilleur résultat dans son contexte. Nous avons aussi obtenu une Accuracy de : 95,08% en combinant l'image et le son, ce résultat est aussi le meilleur dans son contexte.

Pour l'avenir, plusieurs pistes d'amélioration peuvent être envisagées :

— **Définition claire des contenus à modérer :** La définition du contenu NSFW et du contenu à modérer reste assez floue, surtout en Algérie. Une définition précise et officielle permettrait de mieux cadrer les efforts de modération. Collaborer avec des organismes de régulation et des experts en éthique pour établir des critères clairs et acceptés est essentiel.

— **Enrichissement des datasets :** La qualité et la diversité des datasets sont cruciales pour le développement de modèles robustes. Créer des nouveaux datasets compatible avec les définitions du contenu à modérer sont la base et l'un des éléments les plus importants pour la progression générale de la modération du contenu.

— **Amélioration du modèle sonore :** Bien que nous ayons utilisé VGGish pour l'analyse audio, nous n'avons pas eu l'opportunité de tester et d'explorer d'autres techniques et architectures disponibles pour améliorer les performances. Tester d'autres modèles ou des variantes de Transformer pour l'audio pourraient accroître les performances du modèle.

— **Optimisation des modèles vidéo :** Le modèle vidéo peut être encore optimisé en ajustant ses paramètres ou en modifiant son architecture. L'utilisation de modèles RNN peut aussi

améliorer les performances du modèle, faire des tests est nécessaire.

— **Élargissement du champ de détection :** Notre modèle permet de détecter la nudité et la pornographie, mais ce ne sont pas les seuls contenus NSFW. La détection de l'excès de sang, de la violence extrême, des discours haineux, et d'autres types de contenus nuisibles est également importante. Développer des modèles spécialisés pour ces types de contenus pourrait automatiser et améliorer significativement le processus de modération de contenu.

En conclusion, notre modèle actuel présente une solution efficace pour la détection de la nudité et de la pornographie dans les contenus multimédias. Cependant, l'amélioration continue des modèles, l'enrichissement des datasets, et la clarification des critères de modération sont des étapes cruciales pour développer une solution complète et fiable.

Bibliographie

[1] Akyon, F. C. and Temizel, A. (2022). Deep architectures for content moderation and movie content rating. *ArXiv*, abs/2212.04533.

[2] Akyon, F. C. and Temizel, A. (2023). State-of-the-art in nudity classification : A comparative analysis. International Conference on Acoustics, Speech, and Signal Processing (ICASSP), Workshop on Computational Intelligence for Multimedia Understanding.

[3] Al-rimy, B. A. S., Saeed, F., Al-Sarem, M., Albarrak, A. M., and Qasem, S. N. (2023). An adaptive early stopping technique for densenet169-based knee osteoarthritis detection model. *Diagnostics*, 13(11).

[4] AlDahoul, N., Abdul Karim, H., Lye Abdullah, M. H., Ahmad Fauzi, M. F., Ba Wazir, A. S., Mansor, S., and See, J. (2021). Transfer detection of yolo to focus cnn's attention on nude regions for adult content detection. *Symmetry*, 13(1).

[5] Algerie Presse Service (2023). Suspension des programmes de la chaîne Essalam TV pour une durée de 20 jours. *Algerie Presse Service*. [En ligne : `https://www.aps.dz/algerie/159508` ; consulté le 31/05/2024].

[6] Andriluka, M., Pishchulin, L., Gehler, P., and Schiele, B. (2014). 2d human pose estimation : New benchmark and state of the art analysis. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3686–3693.

[7] Avila, S., Thome, N., Cord, M., Valle, E., and de A. Araújo, A. (2013). Pooling in image representation : The visual codeword point of view. *Computer Vision and Image Understanding*, 117(5) :453–465.

[8] Borg, M., Tabone, A., Bonnici, A., Cristina, S., Farrugia, R. A., and Camilleri, K. P. (2022). Detecting and ranking pornographic content in videos. *Forensic Science International : Digital Investigation*, 42-43 :301436.

[9] Cheng, F., Wang, S.-L., Wang, X.-Z., Liew, A. W.-C., and Liu, G.-S. (2019). A global and local context integration dcnn for adult image classification. *Pattern Recognition*, 96 :106983.

[10] Connie, T., Al-Shabi, M., and Goh, M. (2018). Smart content recognition from images using a mixture of convolutional neural networks. In Kim, K. J., Kim, H., and Baek, N., editors, *IT Convergence and Security 2017*, pages 11–18, Singapore. Springer Singapore.

[11] de la Communication, M. (2022). L'arav décide de fermer la chaîne tv "al adjwaa". [En ligne : `https://www.ministerecommunication.gov.dz/fr/node/10742` ; consulté le 31/05/2024].

[12] Dictionaries, O. L. (2024). Definition of nsfw abbreviation. [En ligne : `https://www.oxfordlearnersdictionaries.com/definition/english/nsfw`; consulté le 17/04/2024].

[13] Duy, P., Nguyen, Q.-H., Nguyen, T.-T., Tran, L., and Vu, L. (2022a). Joint inter-intra representation learning for pornographic video classification. *Indonesian Journal of Electrical Engineering and Computer Science*, 25 :1481.

[14] Duy, P., Nguyen, T., Nguyen, Q., Tran, H., Khac, N.-K., and Vu, L. (2022b). Lspd : A large-scale pornographic dataset for detection and classification. *International Journal of Intelligent Engineering and Systems*, 15 :198.

[15] Fatima, A. and Lobna, B. B. (2021). Détection de la rétinopathie diabétique avec le deep learning, transfer learning, cnn, u-net. Master's thesis, Université SAAD DAHLAB de BLIDA, Faculté de Technologie, Département d'Électronique. Spécialité électronique des systèmes embarqués.

[16] Freitas, P., Busson, A., Álan Guedes, and Colcher, S. (2020). A deep learning approach to detect pornography videos in educational repositories. In *Proceedings of the 31st Brazilian Symposium on Computers in Education*, pages 1253–1262, Porto Alegre, RS, Brasil. SBC.

[17] Garcia, M. B., Revano, T. F., Habal, B. G. M., Contreras, J. O., and Enriquez, J. B. R. (2018). A pornographic image and video filtering application using optimized nudity recognition and detection algorithm. In *2018 IEEE 10th International Conference on Humanoid, Nanotechnology, Information Technology,Communication and Control, Environment and Management (HNICEM)*, pages 1–5.

[18] Gautam, N. and Vishwakarma, D. K. (2023). Obscenity detection in videos through a sequential convnet pipeline classifier. *IEEE Transactions on Cognitive and Developmental Systems*, 15(1) :310–318.

[19] He, C., Huang, Q., and Luo, J. (2023). A two-stream convolutional neural network-based pornography recognition method. *International Journal of Computer Applications Technology and Research*, pages 14–17.

[20] Howard, A., Sandler, M., Chen, B., Wang, W., Chen, L., Tan, M., Chu, G., Vasudevan, V., Zhu, Y., Pang, R., Adam, H., and Le, Q. (2019). Searching for mobilenetv3. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 1314–1324, Los Alamitos, CA, USA. IEEE Computer Society.

[21] K P, A. R. and S Nath, G. (2023). Convnext-based mango leaf disease detection : Differentiating pathogens and pests for improved accuracy. *International Journal of Advanced Computer Science and Applications*, 14.

[22] Kanjanasurat, I., Anuwongpinit, T., and Purahong, B. (2023). Image enhancement and 27 pretrained convolutional neural network models for diabetic retinopathy grading. *Sensors and Materials*, 35 :1433.

[23] Karamizadeh, S., Chaeikar, S., and Jolfaei, A. (2022). Adult content image recognition by boltzmann machine limited and deep learning. *Evolutionary Intelligence*, 16.

[24] Ketkar, N. (2017). *Introduction to Keras*, pages 95–109.

[25] Li, Z., Gu, T., Li, B., Xu, W., He, X., and Hui, X. (2022). Convnext-based fine-grained image classification and bilinear attention mechanism model. *Applied Sciences*, 12(18).

[26] Lovenia, H., Lestari, D. P., and Frieske, R. (2022). What did i just hear ? detecting pornographic sounds in adult videos using neural networks. In *Proceedings of the 17th International Audio Mostly Conference*, AM '22, page 92–95, New York, NY, USA. Association for Computing Machinery.

[27] Margaret Rouse (2024). What is Content Moderation ? Definition from Techopedia. *Techopedia*. [En ligne : `https://www.techopedia.com/definition/content-moderation`; consulté le 31/05/2024].

[28] Mathew, A. T., Thomas, C. M., Krishnan, A. G., U, G., Saibu, S., and Salim, T. M. (2020). Automated censorable content identification in videos using deep learning. In *2020 International Conference on Data Analytics for Business and Industry : Way Towards a Sustainable Economy (ICDABI)*, pages 1–6.

[29] Moreira, D., Avila, S., Perez, M., Moraes, D., Testoni, V., Valle, E., Goldenstein, S., and Rocha, A. (2016). Pornography classification : The hidden clues in video space–time. *Forensic Science International*, 268 :46–61.

[30] Nguyen, Q.-H., Nguyen, K.-N.-K., Tran, H.-L., Nguyen, T.-T., Phan, D.-D., and Vu, D.-L. (2020). Multi-level detector for pornographic content using cnn models. In *2020 RIVF International Conference on Computing and Communication Technologies (RIVF)*, pages 1–5.

[31] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Müller, A., Nothman, J., Louppe, G., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Édouard Duchesnay (2018). Scikit-learn : Machine learning in python.

[32] Samal, S., Nayak, R., Jena, S., and Balabantaray, B. (2023). Obscene image detection using transfer learning and feature fusion. *Multimedia Tools and Applications*, 82 :1–29.

[33] Song, K. and Kim, Y.-S. (2020). An enhanced multimodal stacking scheme for online pornographic content detection. *Applied Sciences*, 10(8).

[34] Tabone, A., Camilleri, K., Bonnici, A., Cristina, S., Farrugia, R., and Borg, M. (2021). Pornographic content classification using deep-learning. In *Proceedings of the 21st ACM Symposium on Document Engineering*, DocEng '21, New York, NY, USA. Association for Computing Machinery.

[35] Wang, X., Cheng, F., Wang, S., Sun, H., Liu, G., and Zhou, C. (2018). Adult image classification by a local-context aware network. In *2018 25th IEEE International Conference on Image Processing (ICIP)*, pages 2989–2993.

[36] Zhou, L., Wei, K., Li, Y., Hao, Y., Yang, W., and Zhu, H. (2022). Acoustic pornography recognition using convolutional neural networks and bag of refinements.

Annexe : Modèles pré-entrainés

Introduction

Tous les modèles visuels présentés dans cette annexe ont été pré-entraînés sur le même dataset : ImageNet. Ce dataset est largement utilisé pour l'entraînement de modèles de reconnaissance d'images en raison de sa grande taille et de sa diversité. Le tableau A.1 est une comparaison des performances et du nombre de paramètres de ces modèles.

TABLE A.1: Comparaison des performances et du nombre de paramètres des modèles visuels

Modèle	**Précision Top-1 (%)**	**Nombre de paramètres (millions)**
InceptionV3	77.9	23.9
DenseNet169	76.2	14.3
MobileNetV3Large	75.2	5.4
MobileNetV3Small	67.4	2.5
ConvNeXtTiny	80.1	28.6

Modèles visuels

InceptionV3

InceptionV3 est un modèle de reconnaissance d'images qui passe par plusieurs étapes selon son utilisation. Ce modèle est construit progressivement, étape par étape. D'abord, il commence par la convolution factorisée, qui réduit le nombre de paramètres et le coût de calcul tout en préservant la capacité expressive du modèle. Ensuite, il utilise la convolution plus petite, remplaçant les premières convolutions par des convolutions plus petites (Max-pooling), ce qui permet au modèle de détecter des caractéristiques à différentes résolutions spatiales. Enfin, il utilise la convolution asymétrique, employant des convolutions avec des noyaux de taille asymétrique (les noyaux ne sont pas carrés), ce qui permet de capturer des motifs dans différentes directions spatiales [15].

Ce modèle comporte d'autres étapes appelées les étapes de classification, que nous n'avons pas abordées ici, car nous nous concentrons uniquement sur les étapes utilisées dans notre modèle.

Pour une explication de toutes les couches de ce modèle, voir la figure A.1.

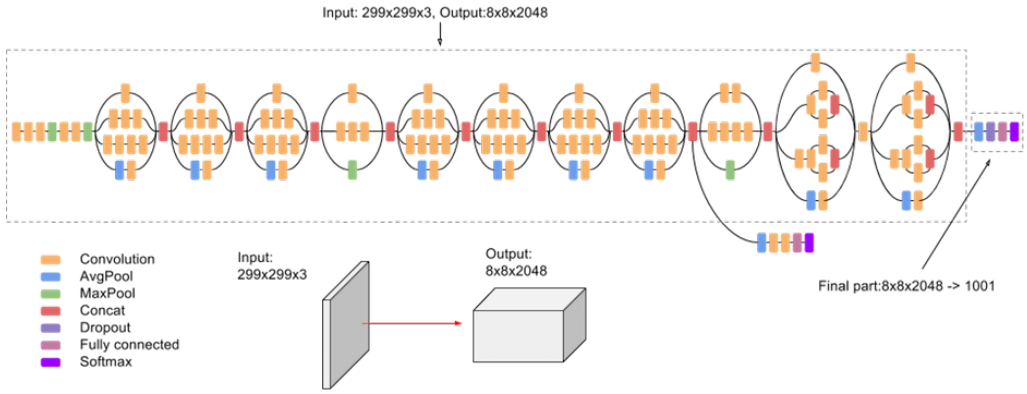


FIGURE A.1 – Architecture en couches d'InceptionV3 [15]

DenseNet169

Layers	Output Size	DenseNet 169
Convolution	112×112	7×7 conv, stride 2
Pooling	56×56	3×3 max pool, stride 2
Dense Block (1)	56×56	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 6$
Transition Layer (1)	56×56	1×1 conv
	28×28	2×2 average pool, stride 2
Dense Block (2)	28×28	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 12$
Transition Layer (2)	28×28	1×1 conv
	14×14	2×2 average pool, stride 2
Dense Block (3)	14×14	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 32$
Transition Layer (3)	14×14	1×1 conv
	7×7	2×2 average pool, stride 2
Dense Block (4)	7×7	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 32$
Classification Layer	1×1	7×7 global average pool
	1000	1000D fully-connected, softmax

FIGURE A.2 – Architecture en couches de DenseNet169 [3]

DenseNet169 est le plus grand modèle de la famille DenseNet, utilisé pour la classification d'images et la reconnaissance d'objets visuels. Il comporte plusieurs couches, et chaque sortie de couche est reliée à la couche suivante, créant un chemin direct entre les couches d'entrée et de sortie. Cela contribue à augmenter la profondeur des CNN profonds tout en évitant la perte d'information. Ce modèle comporte des couches convolutives, de max-pooling, denses et de transition comme expliqué dans la figure A.2. Les couches convolutives permettent d'extraire les caractéristiques des images en appliquant plusieurs filtres. Par exemple, si nous avons une image

de taille $l \times n$ et appliquons un filtre $m \times m$, le résultat de la convolution est :

$$R = (l - m + 1) \times (n - m + 1)$$

où l est la hauteur, n est la largeur et m est la taille du filtre carré $m \times m$. Les couches de max-pooling regroupent les caractéristiques d'une zone en appliquant un filtre. Par exemple, pour une image de taille (hauteur, largeur, canal RGB) ou (H, L, C), la méthode MaxPool regroupe les caractéristiques pour obtenir une carte de caractéristiques de taille plus petite par rapport à l'image d'origine.

$$MaxPool = \frac{C \times (H - f + 1) \times (L - f + 1)}{s \times s}$$

où f est la taille du filtre de pooling $f \times f$ et s est le pas de déplacement du filtre [3].

MobileNetV3

Ce modèle se décline en deux versions : MobileNetV3Large et MobileNetV3Small, utilisées pour la classification d'images et la détection d'objets, ainsi que dans les processus des téléphones mobiles grâce à l'architecture de réseau NAS (Neural Architecture Search) et l'algorithme NetAdapt. Il utilise une combinaison de convolutions séparables en profondeur et de convolutions ponctuelles, réduisant le nombre de paramètres et le coût de calcul par rapport aux convolutions classiques [20, 22]. Les couches utilisées dans le modèle MobileNetV3Large et leurs détails sont expliqués dans la figure A.3.

Input	Operator	Exp Size	#out	SE	NL	s
$224^2 \times 3$	conv2d	-	16	-	HS	2
$112^2 \times 16$	bneck, 3×3	16	16	-	RE	1
$112^2 \times 16$	bneck, 3×3	64	24	-	RE	2
$56^2 \times 24$	bneck, 3×3	72	24	-	RE	1
$56^2 \times 24$	bneck, 3×3	72	40	1	RE	2
$28^2 \times 40$	bneck, 3×3	120	40	1	RE	1
$28^2 \times 40$	bneck, 3×3	120	40	1	RE	1
$28^2 \times 40$	bneck, 3×3	240	80	-	RE	2
$14^2 \times 80$	bneck, 3×3	200	80	-	RE	1
$14^2 \times 80$	bneck, 3×3	200	80	-	RE	1
$14^2 \times 80$	bneck, 3×3	184	80	-	RE	1
$14^2 \times 80$	bneck, 3×3	184	112	1	RE	1
$14^2 \times 112$	bneck, 3×3	480	112	1	RE	1
$14^2 \times 112$	bneck, 3×3	672	160	1	RE	2
$7^2 \times 160$	bneck, 3×3	672	160	1	RE	1
$7^2 \times 160$	bneck, 3×3	960	160	1	RE	1
$7^2 \times 160$	conv2d, 1×1	960	960	-	HS	1
$7^2 \times 960$	avg pool, 7×7	-	-	-	-	1
$1^2 \times 960$	conv2d, 1×1	-	1280	-	HS	1
$1^2 \times 1280$	conv2d, 1×1	-	k	-	-	1

FIGURE A.3 – Architecture en couches de MobileNetV3Large [20]

Les couches utilisées dans le modèle MobileNetV3Small et leurs détails sont expliqués dans la figure A.4.

Input	Operator	Exp Size	Out	SE	NL	S
$224^2 \times 3$	Conv2d, 3×3	-	16	-	HS	2
$112^2 \times 16$	Bneck, 3×3	16	16	$\surd$	RE	2
$56^2 \times 16$	Bneck, 3×3	72	24	-	RE	2
$28^2 \times 24$	Bneck, 3×3	88	24	-	RE	1
$28^2 \times 24$	Bneck, 5×5	96	40	$\surd$	HS	2
$14^2 \times 40$	Bneck, 5×5	240	40	$\surd$	HS	1
$14^2 \times 40$	Bneck, 5×5	240	40	$\surd$	HS	1
$14^2 \times 40$	Bneck, 5×5	120	48	$\surd$	HS	1
$14^2 \times 48$	Bneck, 5×5	144	48	$\surd$	HS	1
$14^2 \times 48$	Bneck, 5×5	288	96	$\surd$	HS	2
$7^2 \times 96$	Bneck, 5×5	576	96	$\surd$	HS	1
$7^2 \times 96$	Bneck, 3×3	576	96	$\surd$	HS	1
$7^2 \times 96$	Conv2d, 1×1	-	576	$\surd$	HS	1
$7^2 \times 576$	Pool, 7×7	-	-	-	-	1
$1^2 \times 576$	Conv2d 1×1, NBN	-	1024	-	HS	1
$1^2 \times 1024$	Conv2d 1×1, NBN	-	1000	-	-	1

FIGURE A.4 – Architecture en couches de MobileNetV3Small [20]

Lorsqu'un filtre de taille $m \times m$ est appliqué à une image de C canaux, produisant une sortie de N canaux, le nombre de paramètres change selon la méthode de convolution, comme expliqué ci-dessous :

Convolutions séparables en profondeur

$$paramtres = m \times m \times C$$

Convolutions ponctuelles

$$paramtres = 1 \times 1 \times C \times N$$

En combinant les convolutions séparables en profondeur et les convolutions ponctuelles, comme dans le modèle MobileNetV3Large, le nombre de paramètres est réduit par rapport aux convolutions classiques.

$$paramtres = (m \times m \times C) + (1 \times 1 \times C \times N)$$

Convolutions classiques

$$paramtres = m \times m \times C \times N$$

ConvNeXtTiny

Ce modèle a été développé comme une extension de l'architecture de type Transformer, ajoutant des couches convolutives pour apprendre les caractéristiques avec le mécanisme d'attention. Il utilise une structure convolutive parallèle, contrairement aux réseaux de neurones convolutifs traditionnels qui utilisent une série de couches de convolution suivies de couches entièrement connectées [21]. Pour plus de détails sur l'architecture des couches de ce modèle, voir les figures A.5 et A.6.

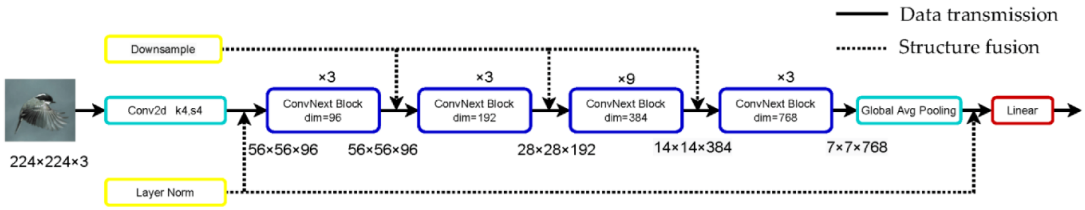


FIGURE A.5 – Structure du modèle ConvNeXtTiny [25]

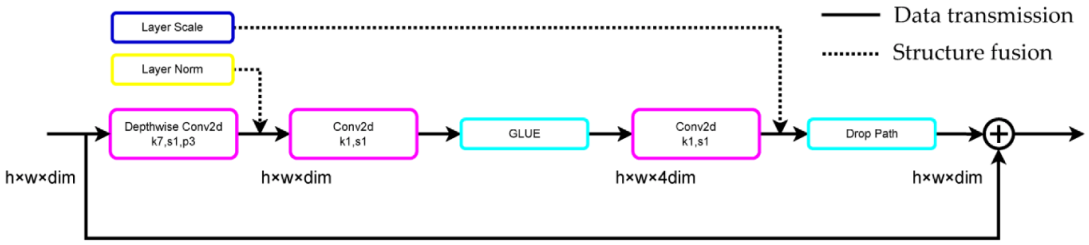


FIGURE A.6 – Structure du bloc du modèle ConvNeXtTiny [25]

Modèle sonore

VGGish

VGGish est un modèle basé sur l'architecture VGG, spécialement conçu pour le traitement des données audios. Il est utilisé pour l'extraction de caractéristiques audios dans des tâches telles que la classification des sons et la reconnaissance vocale. VGGish utilise des couches convolutives pour apprendre les caractéristiques des signaux audio, ce qui permet de capturer des informations importantes dans le domaine temporel et fréquentiel.

VGGish [1] est implémenté en Keras avec TensorFlow en tant que modèle de classification audio de type VGG. Ce modèle a été développé à partir du modèle utilisé pour AudioSet, un ensemble de données d'événements audio annotés par des humains comprenant plus de 2 millions de bandes sonores de vidéos YouTube de 10 secondes.

1. `https://github.com/DTaoo/VGGish`, consulté le 18/06/2024