Détection des défauts dans les vieux films par apprentissage profond à partir d'une restauration semi-manuelle

Arthur Renaudeau, Travis Seng, Axel Carlier, Fabien Pierre, François Lauze, Jean-François Aujol, Jean-Denis Durou

Détection des défauts dans les vieux films par apprentissage profond à partir d'une restauration semi-manuelle

Arthur RENAUDEAU[1] Travis SENG[1] Axel CARLIER[1] Fabien PIERRE[2]
François LAUZE[3] Jean-François AUJOL[4] Jean-Denis DUROU[1]

[1] IRIT, UMR CNRS 5505, Université de Toulouse
[2] Université de Lorraine, CNRS, Inria, LORIA, F-54000 Nancy, France
[3] DIKU, Université de Copenhague, Danemark
[4] Université de Bordeaux, Bordeaux INP, CNRS, IMB, UMR 5251, F-33400 Talence, France

arthur.renaudeau@irit.fr

Résumé

La détection des défauts constitue la première étape de la restauration de vieux films. La spécificité de notre travail est d'apprendre l'expertise d'un restaurateur de films au travers d'une paire de séquences constituée d'un film comportant des défauts, et de ce même film après restauration semi-manuelle aidée d'un logiciel spécialisé. Pour détecter les défauts avec un minimum d'interaction humaine, et réduire le temps consacré à la restauration, nous alimentons un réseau de neurones de type U-Net avec une séquence d'images défectueuses en entrée, afin de détecter les variations spatio-temporelles anormalement élevées de l'intensité des pixels. La sortie du réseau étant le masque des défauts, nous créons des masques comparatifs en utilisant les différences entre les versions défectueuse et restaurée du film (ceux utilisés lors de la restauration n'étant pas directement accessibles). Notre réseau réussit à détecter automatiquement les défauts réels plus précisément que les sélections manuelles, voire certains défauts oubliés par l'expert en restauration.

Mots Clef

Détection de défauts, vieux films, apprentissage profond.

Abstract

Detection of defects is the first step in the restoration of old movies. The specificity of our work is to learn the expertise of a film restorer through a pair of sequences consisting of a film with defects, and this same film after semi-manual restoration with the help of a specialized software. In order to detect defects with a minimum of human interaction, and to reduce the time spent on restoration, we feed a U-Net neural network with a sequence of defective images as input, in order to detect abnormally high spatio-temporal variations in pixel intensity. The output of the network being the defect mask, we create comparative masks using the differences between the defective and restored versions of the film (those used during restoration not being directly accessible). Our network manages to automatically detect real defects more accurately than manual selections, or even some defects missed by the restoration expert.

Keywords

Defect detection, old movies, deep learning.

1 Introduction

Grâce à la puissance de calcul accrue des ordinateurs modernes, capables d'exploiter les GPU et la rapidité des multiprocesseurs, des algorithmes sophistiqués, notamment ceux qui se fondent sur des réseaux de neurones, sont développés pour restaurer les vieux films. Une intervention humaine fastidieuse et coûteuse reste cependant nécessaire, notamment pour la détection des défauts. De nombreux types de défauts dans les vieux films ont été inventoriés [11], parmi lesquels les plus étudiés, vis-à-vis de la restauration, sont les taches et les rayures verticales [12].

Les taches sont des ensembles de pixels sombres ou clairs interconnectés qui apparaissent à des endroits spatio-temporels aléatoires dans les séquences vidéo. Elles sont causées par la présence de particules ou par une mauvaise manipulation de produits chimiques sur la bande de film. Elles ont des formes aléatoires, une forte corrélation spatiale et une faible corrélation temporelle, car il est très improbable de trouver deux taches de même forme dans deux images consécutives. En revanche, les rayures présentent des caractéristiques opposées, avec une faible épaisseur spatiale mais une forte corrélation temporelle. Elles sont produites par le frottement d'une particule sur la bande de film pendant la projection ou la duplication. Vu le sens de déroulement du film, les rayures apparaissent le plus souvent verticalement et prennent la forme de lignes verticales persistantes sur plusieurs images consécutives. Les rayures sont difficiles à détecter à cause

de leur similitude avec des éléments naturels présents dans une séquence vidéo, tels que des lampadaires.

Nous proposons une détection de ces artefacts construite à partir de connaissances expertes. Nous partons d'un jeu de données assez particulier, puisqu'il contient les images d'un film détérioré, ainsi que la restauration de ce film par un expert en restauration. À partir de ces données, nous souhaitons, dans un premier temps, trouver le masque des défauts associé à chaque image détériorée, en la comparant avec l'image restaurée. Une fois le jeu de données de masques obtenu, nous entraînons un réseau de neurones avec des séquences d'images en entrée pour prendre en compte la variation temporelle d'intensité due aux défauts. En résumé, nos principales contributions sont les suivantes :

- Nous apprenons à identifier les défauts directement à partir de l'expertise d'un professionnel de la restauration de films combinée aux traitements automatiques d'un logiciel spécialisé, ce qui constitue à notre connaissance une nouvelle approche de l'identification des défauts.
- Dans ce but, nous construisons un pipeline générant un ensemble de données de masques de défauts en comparant soigneusement les images défectueuses et restaurées.
- Nous entraînons un réseau de neurones adapté qui effectuera la détection automatique des défauts.

(a) Trois images consécutives avec des rayures et des taches dans l'image centrale.

(b) Les mêmes images après restauration par un expert.

FIGURE 1 – Images numérisées d'un vieux film et sa restauration semi-automatique. Notre réseau calcule un masque des défauts à partir de la séquence originale seule (a). L'étape d'apprentissage se fonde sur les deux séquences (défectueuse (a) et restaurée (b)) pour récupérer un masque de défauts résultant d'un traitement sur les différences entre ces deux images.

2 État de l'art

Les premiers détecteurs de défauts (heuristiques) dans les vidéos étaient à l'origine utilisés pour détecter le bruit impulsionnel, comme celui mis en œuvre par [24] pour la BBC, qui utilise un seuil sur les différences absolues entre images consécutives pour détecter les défauts. Cependant, le mouvement apparent entre les images n'est pas pris en compte. C'est pourquoi [14] a introduit l'indice de détection des pics (SDI), avec quelques variantes SDIa ou SDIx [11], où les différences absolues entre images sont utilisées mais avec compensation du mouvement cette fois.

2.1 Détection des rayures

Le premier détecteur de rayures a été proposé par [10], les lignes étant modélisées par des sinusoïdes amorties contenues dans l'image. Un sous-échantillonnage vertical et un filtrage combinés à une transformation de Hough sont d'abord effectués, puis un raffinage bayésien permet de conserver les lignes correspondant au modèle. Ce modèle a été généralisé dans [3] pour détecter les rayures en utilisant la transformation en ondelettes et la loi de Weber dans des films en niveaux de gris, puis dans des films en couleur dans [2]. La fermeture morphologique a été utilisée sur une image dans [7] pour détecter les rayures à partir de la différence avec l'original, puis pour les suivre dans les autres images avec un filtre de Kalman. Cette idée de fermeture a été reprise par [21], après avoir découpé l'image en plusieurs bandes horizontales, afin de séparer le premier plan de l'arrière-plan pour une meilleure détection. Dans [8], une dérivée horizontale seuillée est appliquée à l'image, puis la moyenne de chaque colonne est calculée et seuillée à nouveau pour détecter les rayures parmi tous les contours verticaux. La transformation de Hough ainsi que des filtres médians avec une taille de fenêtre variable sont utilisés dans [4]. En plus de la méthode de [10], [17] examine les valeurs des pixels à gauche et à droite de la rayure pour la cohérence, afin de limiter les fausses alarmes dues aux contours. Ensuite, les différentes lignes possibles qui étaient fermées sont regroupées par une méthode a contrario. Une autre idée pour limiter les fausses alarmes a été introduite dans [16], qui consiste à réaligner les différentes images du masque en suivant le mouvement apparent et en éliminant les détections de rayures qui restent purement verticales, donc sont susceptibles de représenter des éléments réels dans la scène.

2.2 Détection des taches

Dans [13], deux modèles de détection sont développés : un modèle à champs aléatoires de Markov, et un modèle autorégressif 3D, associés à leurs heuristiques de détection respectives. Ce modèle MRF est ensuite utilisé dans [26], suivi de deux étapes de raffinage pour éliminer les fausses alarmes. Les deux contraintes consistent, d'une part, à imposer la continuité spatiale avec un MRF et, d'autre part, à imposer une contrainte de corrélation temporelle avec un suiveur de caractéristiques pyramidal de type Lucas-Kanade. Le détecteur de différences ordonnées (ROD) a quant à lui été introduit dans [15]. Il consiste à utiliser les pixels temporellement voisins dans les images avant et arrière avec compensation du mouvement, pour compa-

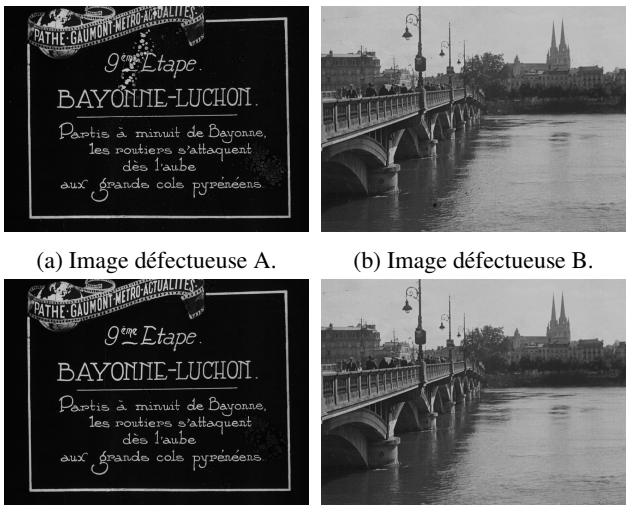
rer leur distance au pixel courant à la valeur moyenne de cette distance. Une version simplifiée du détecteur ROD (SROD) dans [1] ne traite que les valeurs minimales et maximales des pixels voisins. Le détecteur SROD a également été utilisé dans [25], où la détection est combinée avec un détecteur spatial fondé sur la croissance et la fermeture morphologiques des zones. Une autre application du détecteur SROD a été présentée dans [6], où il est appliqué en deux étapes : la première est la version classique, tandis que la seconde est utilisée après compensation du mouvement dans les images. La détection est réalisée dans [18] comme une segmentation d'image par croissance de régions sans graines tout en introduisant une nouvelle mesure de confiance fondée sur les différences temporelles de l'image. D'autres méthodes nécessitent plusieurs étapes pour détecter les taches. Par exemple, dans [27], la première étape consiste à trouver de potentielles taches en fonction de leurs caractéristiques spatiales. Parmi ces candidats, les véritables taches sont détectées par la discontinuité temporelle des intensités. La détection de régions considérées comme des taches est effectuée dans [28] en utilisant la détection de changements soudains dans une région à l'aide du filtrage médian temporel. Ensuite, ces régions sont classées comme étant des taches ou non, en trouvant la similarité de la région candidate dans les images adjacentes dans l'espace de gradient, avec un détecteur d'histogramme de gradients orientés.

2.3 Détection par apprentissage profond

La première application de l'apprentissage profond a été consacrée à la détection des rayures dans [9] en utilisant la séparation de la forme et de la texture. La détection de la forme est réalisée par filtrage, tandis que la texture est classée par un réseau de neurones avec les images de contours en entrée. Une détection de taches en trois étapes a été proposée dans [22] : compensation de mouvement, détection SROD [1] et classification de tous les pixels ayant des valeurs anormales, en utilisant un réseau de neurones convolutif. Les mêmes auteurs ont essayé une approche en trois étapes dans [23]. La première étape consiste à créer un descripteur contenant : la luminosité des trois images consécutives, la même pour les images après compensation du mouvement, l'amplitude du flux optique de Lucas-Kanade, et enfin les modèles binaires issus de l'opérateur de motif binaire local. Ensuite, une détection SDI [14] est effectuée. Son résultat et le descripteur sont enfin passés en entrée d'un CNN. Pour la détection des taches et des rayures, [29] a appliqué une classification avec une architecture CNN encodeur-décodeur, comportant la concaténation de couches dans la partie encodeur. Ensuite, en utilisant la sortie du réseau avant la dernière convolution pour l'image actuelle et l'image précédente, une mise en commun de la moyenne spatiale est effectuée et un seuil sur la distance euclidienne des deux résultats permet de détecter les taches. Les rayures sont quant à elles détectées après fermeture morphologique de la sortie du réseau et, avec des considérations d'analyse de forme, les défauts dont la hauteur est beaucoup plus grande que la largeur sont conservés.

3 Préparation du jeu de données

Notre jeu de données est un film composé de deux ensembles d'environ 3000 images en niveaux de gris de taille 1728×1280 pixels : le premier contient les images originales comportant des défauts, le second est constitué des mêmes images après restauration par un expert (voir FIGURE 2) à l'aide du logiciel de restauration de films DIAMANT-Film [1]. À partir d'une paire d'images défectueuse et restaurée, nous créons un masque de zones défectueuses que nous utilisons comme sortie à prédire par le réseau de neurones.



(a) Image défectueuse A. (b) Image défectueuse B.

(c) Image restaurée A. (d) Image restaurée B.

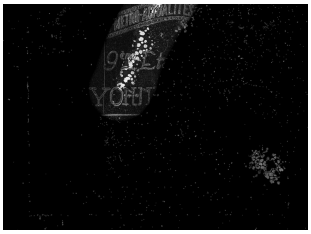
FIGURE 2 – Exemples d'images respectivement défectueuses et restaurées. Les différentes images du film représentent du texte (a) ou des scènes naturelles (b).

3.1 Création de masques par différences d'images et seuillage

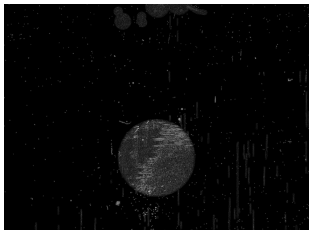
L'idée pour obtenir les images de masque est de calculer la différence absolue pour chaque paire d'images défectueuse et restaurée (voir FIGURE 3), puis de seuiller cette différence, qui est comprise entre 0 et 65535 (pour des images en 16 bits).

Comme cela apparaît sur la FIGURE 3, les plus gros défauts sont détectés et sélectionnés manuellement par l'utilisateur du logiciel, avec des formes géométriques simples. Malheureusement, le logiciel effectue une copie des images voisines sur toute la sélection, même si certaines parties ne doivent pas être restaurées. C'est pourquoi le choix du seuillage est dicté par le compromis suivant : détecter suffisamment de défauts, mais éviter la sur-détection.

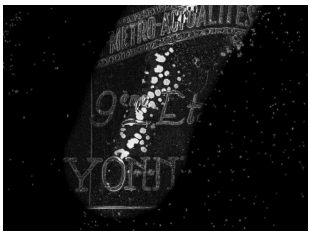
1. https ://www.hs-art.com/index.php/solutions/diamant-film



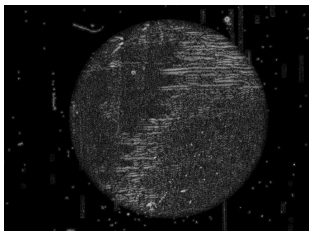
(a) Différence des images A.



(b) Différence des images B.

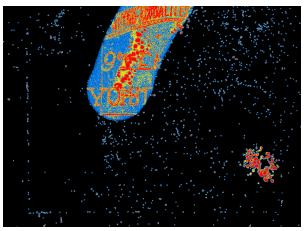


(c) Zoom sur (a).

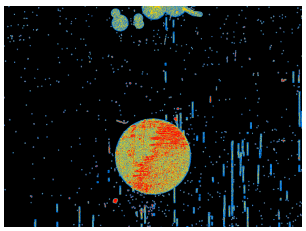


(d) Zoom sur (b).

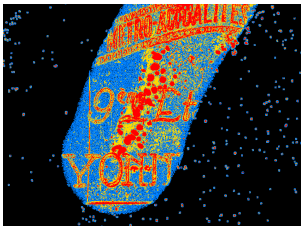
FIGURE 3 – Différences absolues entre les images défectueuses et restaurées de la FIGURE 2. Si les différences les plus importantes sont plus claires, les formes géométriques du logiciel de restauration assistée par ordinateur sont également facilement reconnaissables.



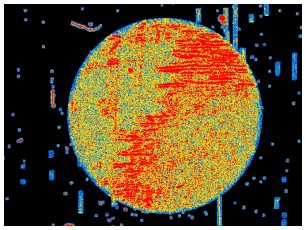
(a) Différence des images A - Classification.



(b) Différence des images B - Classification.



(c) Zoom sur (a).



(d) Zoom sur (b).

FIGURE 4 – Classification des différences présentes sur la FIGURE 3. Les pixels bleus correspondent au seuil minimal et ne doivent pas être sélectionnés. Considérer les lettres à l'intérieur de la sélection manuelle comme de véritables défauts est plus difficile à appréhender, malgré une forte variation d'intensité.

Avec les différents seuils choisis expérimentalement, nous avons estimé la limite de sur-détection à au moins 1% de la valeur maximale (en jaune sur la FIGURE 4), ce qui représente seulement 30% des pixels ayant été restaurés. En particulier, avec un seuil à 5% de la valeur maximale (en rouge sur la FIGURE 4), nous pouvons distinguer les formes des différents défauts, même si la correction effectuée sur les lettres est encore visible. Cependant, avec un seuil plus élevé, les défauts réels ne sont pas détectés avec précision. Même avec un seuil soigneusement choisi, il y a des « trous » dans la détection car l'intensité de certains pixels restaurés peut rester inchangée.

3.2 Remplissage de masques par fermeture morphologique

Pour résoudre le problème des pixels non détectés entourés de pixels défectueux, nous avons décidé de remplir ces zones en utilisant la fermeture morphologique afin de récupérer la connectivité spatiale. Les noyaux des filtres morphologiques utilisés dépendent fortement des types de défauts présents dans les images. Comme expliqué précédemment, il s'agit principalement de taches qui ont des formes arrondies et de rayures verticales. Le paramètre important est la taille de la fermeture à effectuer sur les images du masque I_{masque}. Pour le faire automatiquement, nous choisissons des tailles de fermeture de départ assez petites T_{ligne} et T_{disque}, puis augmentons progressivement chaque taille de fermeture de 1 pixel. La condition d'arrêt pour la taille de fermeture correcte est atteinte lorsque le nombre de nouveaux pixels qui deviennent des pixels de masque augmente plus qu'à l'itération précédente (voir **Algorithme 1**).

Algorithme 1 - Fermeture appliquée aux masques.

$T_{\text{ligne}} \leftarrow 3$, $T_{\text{disque}} \leftarrow 2$, I_{masque}

$\Delta \leftarrow +\infty$, $\Delta^* \leftarrow \#(I_{\text{masque}})$

tant que $\Delta > \Delta^*$ **faire**

  $I^*_{\text{masque}} \leftarrow \text{fermeture}\,(I_{\text{masque}}, T_{\text{ligne}})$

  $I^*_{\text{masque}} \leftarrow \text{fermeture}\,(I^*_{\text{masque}}, S_{\text{disque}})$

  $\Delta \leftarrow \Delta^*$

  $\Delta^* \leftarrow \#(I^*_{\text{masque}} - I_{\text{masque}})$

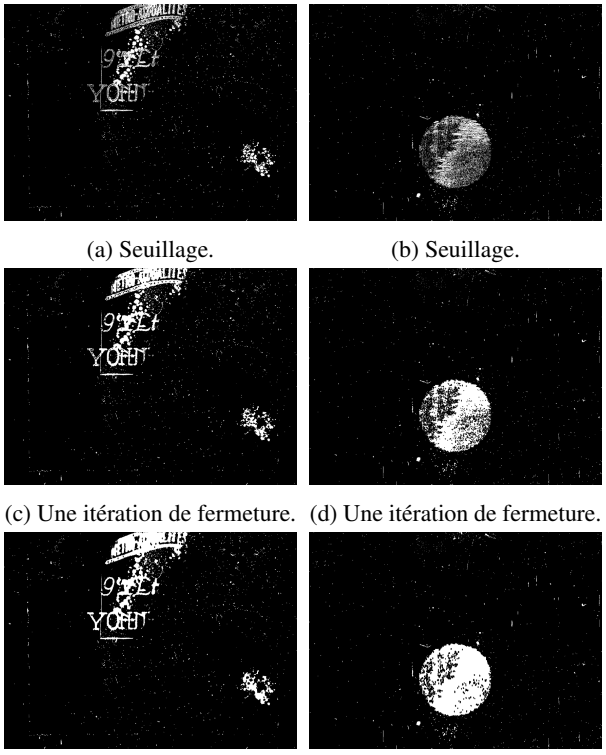
  $I_{\text{masque}} \leftarrow I^*_{\text{masque}}$

  $S_{\text{ligne}} \leftarrow S_{\text{ligne}} + 1$, $S_{\text{disque}} \leftarrow S_{\text{disque}} + 1$

fin tant que

Les résultats de notre algorithme de fermeture morphologique des masques sont présentés sur la FIGURE 5. Il est particulièrement efficace pour combler les gros défauts (voir FIGURE 5) et pour relier les différentes parties éloignées d'une même rayure verticale.

Nous avons ensuite recherché les différentes profondeurs de défaut, afin de connaître la corrélation temporelle entre certains d'entre eux. En particulier, nous avons construit une image avec la profondeur maximale des défauts pour chaque pixel dans le jeu de données, avec son histogramme associé sur la FIGURE 6. À l'exception de quelques gros



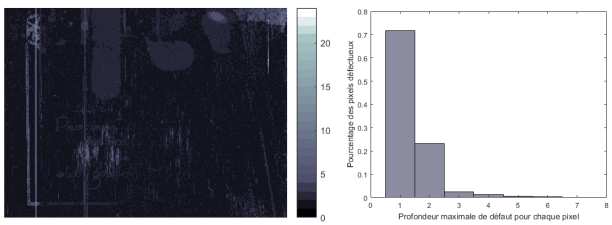
(a) Seuillage. (b) Seuillage.

(c) Une itération de fermeture. (d) Une itération de fermeture.

(e) Dernière itération. (f) Dernière itération.

FIGURE 5 – Fermeture morphologique des images de masque après seuillage. Chaque forme est automatiquement remplie par notre algorithme, qui augmente la taille des noyaux morphologiques à chaque itération.

défauts se chevauchant partiellement sur deux images successives, les seuls défauts profonds sont les rayures. Dans l'ensemble, 72% des pixels ont une profondeur de défaut maximale d'une image et 95% d'entre eux ont une profondeur maximale de deux images. Par conséquent, l'utilisation de trois images consécutives pour l'entraînement doit permettre de détecter la grande majorité des défauts.



(a) Profondeur maximale de défaut pour chaque pixel. (b) Histogramme normalisé associé.

FIGURE 6 – Profondeur temporelle maximale des différents défauts. Les défauts ayant les plus grandes profondeurs dans (a) sont des rayures. On peut également identifier quelques grandes formes sélectionnées par le restaurateur. Pour 72% des pixels, les défauts ont une profondeur maximale d'une image, le score cumulé à une profondeur maximale de deux images étant de 95%.

4 Expériences sur le réseau de neurones

4.1 Partition de l'ensemble de données

Pour rappel, notre jeu de données est composé d'un total de 2974 images de 1728×1280 pixels, qui sont réparties en 23 scènes. La première étape a consisté à diviser ces 23 scènes en trois types de scènes de natures différentes : les scènes contenant uniquement du texte explicatif ou descriptif, les scènes avec un plan fixe et les scènes où la caméra est en mouvement (voir TABLE 1).

TABLE 1 – Répartition des scènes et des images dans les trois types de scènes.

Types de scènes	**Nombre de scènes**	**Nombre d'images**
Texte	7 (30%)	931 (31%)
Plan fixe	9 (40%)	1105 (37%)
Caméra en mouvement	7 (30%)	938 (32%)

La deuxième étape consiste à diviser les scènes en trois ensembles de données différents pour l'apprentissage, la validation et le test. Pour l'ensemble de validation et l'ensemble de test, nous choisissons une scène de chaque type, ce qui signifie trois scènes pour chacun de ces deux ensembles, et 17 scènes pour l'apprentissage (voir TABLE 2). Après cette partition, d'autres manipulations sont nécessaires pour l'analyse statistique et les contraintes sur les tailles d'entrée (*patches* de taille 512×512) et de sortie lors de l'implémentation du réseau de neurones.

TABLE 2 – Répartition des scènes, des images et des triplets de *patches* en fonction de l'ensemble d'apprentissage, de l'ensemble de validation et de l'ensemble de test.

Ensemble	**Nombre de scènes**	**Nombre d'images**	**Nombre de *patches***
Apprentissage	5+7+5	2296 (77%)	27144
Validation	1+1+1	366 (12%)	4320
Test	1+1+1	312 (11%)	3672

4.2 Modèle U-Net avec *patches* spatio-temporels

Nous avons décidé d'utiliser un réseau U-Net, initialement conçu pour la segmentation d'images biométriques [20].

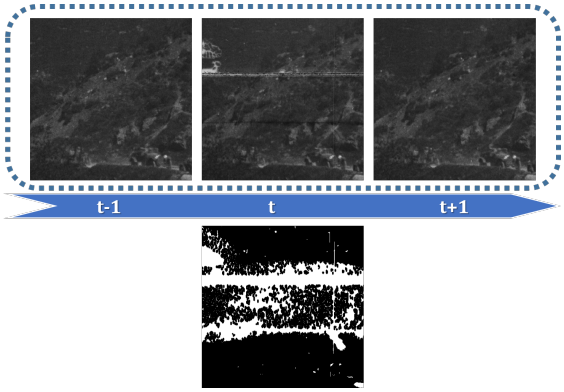


FIGURE 7 – Entrée du réseau U-Net : trois *patches* consécutifs d'images défectueuses, et masque associé du *patch* central pour opérer la comparaison avec la sortie du réseau.

L'un des problèmes les plus récurrents dans l'utilisation des réseaux de neurones est que cela implique de coder les tailles des images d'entrée et de sortie en plus des poids. Par conséquent, le réseau ne peut pas être utilisé avec des images d'une taille différente de celles utilisées lors de la formation. La solution du redimensionnement, qui modifie les proportions de l'image, n'est pas non plus satisfaisante. Par conséquent, nous nous sommes débarrassés de cette contrainte en nous entraînant sur des *patches* et non pas sur les images entières. Ce choix n'est pas un problème dans notre cas d'utilisation particulier, car les défauts peuvent survenir de manière aléatoire n'importe où dans l'image : il n'y a pas de préalable spatial à apprendre. L'étape de prédiction avec l'image a également lieu avec l'image découpée en *patches* qui se chevauchent. Une conséquence de cette opération est l'augmentation de la taille du jeu de données (voir TABLE 2). Par rapport au réseau U-Net original utilisé avec une seule image couleur, nous avons utilisé trois *patches* consécutifs en entrée (voir TABLE 2). Le masque de défauts du *patch* central est utilisé pour la comparaison avec la sortie. dans la fonction de perte du réseau (voir FIGURE 7).

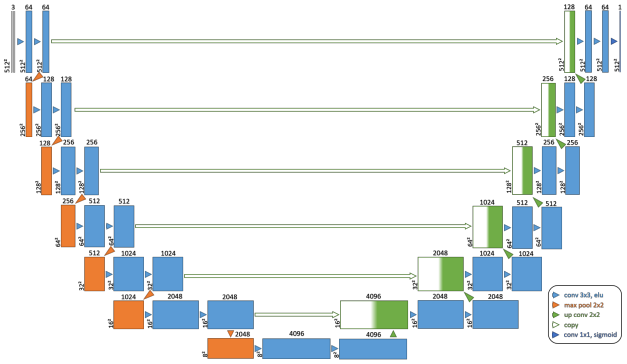


FIGURE 8 – Architecture U-Net utilisée pour la détection des défauts, avec trois *patches* consécutifs de taille 512×512 en entrée, sept couches, et un *patch* de détection des défauts en sortie.

L'architecture de notre réseau U-Net, qui est illustrée sur la FIGURE 8, comporte une partie encodage puis décodage comme dans [20], mais avec sept couches dans notre cas. Le chemin de contraction consiste en l'application répétée de deux étapes de convolutions de taille 3×3, chacune suivie d'une activation ELU (unité linéaire exponentielle) et d'une opération de *max-pooling* sur une fenêtre de taille 2×2 avec un pas de 2 pour le sous-échantillonnage. À chaque étape de sous-échantillonnage, le nombre de canaux de caractéristiques est doublé. Chaque étape du chemin d'expansion consiste en un suréchantillonnage de la carte de caractéristiques suivi d'une étape de convolution de taille 2×2 qui divise par deux le nombre de canaux de caractéristiques, d'une concaténation avec la carte de caractéristiques correspondante du chemin de contraction, et de deux étapes de convolutions de taille 3×3, chacune suivie d'une activation ELU. Dans la dernière couche, on se ramène à une sortie de profondeur 1 à laquelle est appliquée une sigmoïde pour mettre en correspondance chaque vecteur de caractéristiques avec la classe associée. La fonction de perte pour l'apprentissage du réseau est l'inverse de l'approximation linéaire du coefficient de Dice (autre appellation du F1-Score), définie comme suit :

$$\begin{aligned}\text{Loss}\,(y_C, y_U) &= -\frac{2\sum_{i,j} y_C(i,j)y_U(i,j)}{\sum_{i,j} y_C(i,j) + y_U(i,j)} \\ &\approx -\frac{2\times TP}{(TP+FP)+(TP+FN)} \in [-1,0]\end{aligned}$$

où $y_C(i,j) \in \{0,1\}$ et $y_U(i,j) \in [0,1]$ désignent respectivement les pixels du *patch* de masque de défauts de la restauration du film par l'expert, et le *patch* du masque de défauts en sortie du réseau U-Net. Les valeurs TP, FP et FN représentent respectivement les nombres de pixels comptés comme vrais positifs, faux positifs et faux négatifs. Le réseau a été entraîné à l'aide de l'optimiseur Adam, avec un taux d'apprentissage de 5.10^{-5}.

TABLE 3 – Fonction de perte par rapport aux différents ensembles de données et types de scènes calculés après la prédiction.

	Texte	**Plan fixe**	**Mouvement**
Apprentissage	-0,625	-0,456	-0,363
Validation	-0,842	-0,290	-0,201
Test	-0,849	-0,547	-0,248

Les résultats en termes de fonction de perte, après avoir prédit toutes les images possibles dans l'ensemble du jeu de données (voir TABLE 3), montrent que, plus les scènes

sont complexes, plus il est difficile pour le réseau de détecter les défauts avec précision. En effet, les scènes comportant uniquement du texte blanc sur fond noir ont de meilleurs scores que les scènes de plan fixe, qui ont également de meilleurs scores que les scènes comportant un mouvement de caméra. Les scores peuvent ne pas sembler totalement satisfaisants. L'une des raisons est que l'expertise du restaurateur ne constitue pas réellement une vérité terrain, ce qui est dû par exemple à la surdétection liée à de grandes sélections manuelles. Comme le montre la matrice de confusion de la TABLE 4, le pourcentage de TP / FP / FN est vraiment faible par rapport au total des pixels impliqués dans les images. Il est difficile de conclure quant à la réelle bonne détection ou non en fonction de la qualité des masques que nous utilisons pour l'entraînement. Pourtant, les prédictions correctes représentent 99,58% de l'ensemble des pixels.

TABLE 4 – Matrice de confusion entre les masques créés et prédits.

Créé \ Prédit	**0**	**1**
0	99,44%	0,10%
1	0,32%	0,14%

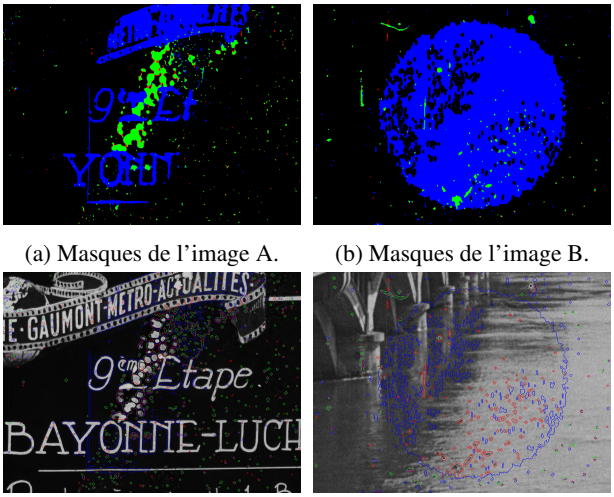
Nous avons également étudié différents ensembles d'entrée afin de déterminer la quantité d'informations temporelles nécessaire à une détection correcte. Comme prévu, il est plus efficace d'utiliser trois images qu'une seule, puisque la plupart des défauts n'apparaissent que sur une seule image. Plus surprenant, l'utilisation de cinq images est moins efficace que trois.

TABLE 5 – Fonction de perte sur l'ensemble de test pour une, trois et cinq images en entrée.

Nombre d'images	1	3	5
F1-Score (test)	-0,352	**-0,419**	-0,374

4.3 Résultats sur l'ensemble de données

L'utilisation de formes géométriques pour sélectionner les différentes zones à restaurer implique que toute la sélection englobant les défauts est remplacée par une copie des images voisines, y compris les pixels non défectueux. Malgré le seuillage et la fermeture morphologique, nous avions indiqué que les lettres des textes ne pouvaient être écartées du masque. Par conséquent, cela conduit à de nombreux pixels considérés comme faux négatifs sur la FIGURE 9. Le même problème se pose pour les autres scènes avec les grandes sélections manuelles qui ne sont pas complètement détectées comme des défauts par le réseau.



(a) Masques de l'image A. (b) Masques de l'image B.

(c) Contours des masques (a). (d) Contours des masques (b).

FIGURE 9 – Comparaison des différents masques pour les images A et B. Les vrais positifs sont signalés en vert, les faux négatifs en bleu, et les faux positifs en rouge. Les contours des formes de défauts sont ajoutés sur les images défectueuses, à l'aide du même code couleur.

D'autre part, certains défauts n'ont semble-t-il pas été détectés par l'expert en restauration, probablement en raison du grand nombre d'images à traiter et du temps limité qu'il a pu y consacrer. Pourtant, le réseau parvient à surpasser la détection manuelle de la restauration dans ce cas et les défauts réels qui n'ont pas été détectés auparavant (voir FIGURE 10), même s'ils sont considérés comme des faux positifs.

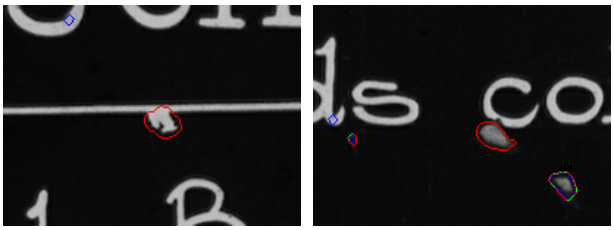


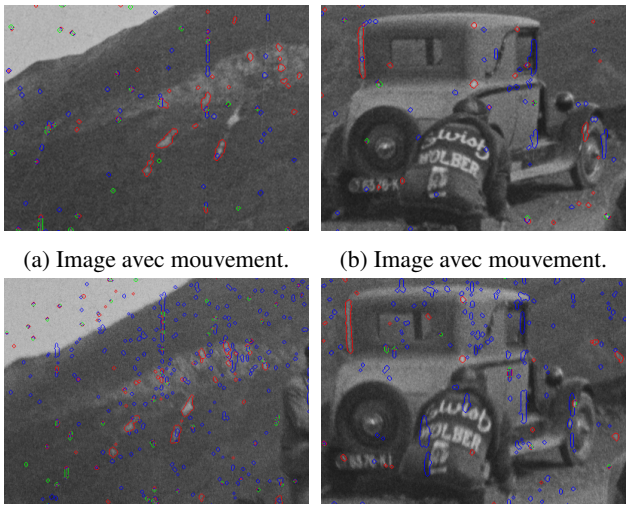
FIGURE 10 – Défauts réels considérés comme des faux positifs en raison de la non détection par l'expert en restauration, sans doute à cause du peu de temps qu'il a pu y consacrer.

Malgré ces bons résultats visuels, qui ne se reflètent pas forcément dans la fonction de perte, notre travail présente encore quelques faiblesses. Par exemple, même si le jeu de données en comporte très peu, la détection des rayures reste limitée par le fait de n'utiliser que trois images consécutives (trois *patches*, pour être précis) afin de détecter une

(a) $I_{(n-1)}$ (b) $I_{(n)}$ (c) $I_{(n+1)}$ (d) $M_{(n)}$

FIGURE 11 – La rayure n'est pas détectée (en bleu) lorsque sa profondeur est supérieure à la profondeur de détection du réseau. Elle est considérée comme faisant partie des pixels non défectueux puisque la rayure est présente au même endroit dans toutes les images.

anomalie temporelle dans l'intensité des pixels. En effet, dans l'exemple de la FIGURE 11, la rayure est présente au même endroit dans les trois images consécutives et le réseau ne peut donc pas détecter d'anomalie temporelle. Une autre limitation du réseau concerne la détection des défauts lorsqu'il y a un mouvement important dans la scène. En effet, dans ce cas, le réseau ne parvient pas à compenser suffisamment le mouvement pour récupérer les pixels correspondants dans les images voisines. Par conséquent, il considère certaines parties des images comme des taches, comme nous pouvons le voir sur la FIGURE 12.



(a) Image avec mouvement. (b) Image avec mouvement.

(c) Image suivante de (a). (d) Image suivante de (b).

FIGURE 12 – Fausses détections (en rouge) qui ne devraient pas être détectées, contrairement à la FIGURE 10. En raison du mouvement important, le réseau les détecte comme étant des taches.

4.4 Comparaison avec une autre séquence

Nous avons utilisé une autre séquence de [17] (voir FIGURE 13) afin de comparer notre détecteur de défauts au leur. Même si les rayures ne sont pas entièrement détectées dans notre cas, ce qui est le but de la méthode de [17], presque tous les autres défauts de type tache sont quant à eux détectés.

(a) Newson et *al.* [17]. (b) Notre méthode.

FIGURE 13 – Comparaison entre la détection des rayures de [17] en rouge (a), et notre méthode de détection générale des défauts avec les contours en bleu (b) sur la séquence « Star ».

5 Conclusion et perspectives

Nous avons introduit une nouvelle approche en termes de détection de défauts dans les vidéos, en ouvrant l'étude pour l'entraînement sans la localisation de ces défauts, mais seulement avec les images défectueuses et restaurées. À partir de ces images, nous avons créé une possibilité de masques de défauts associés, qui se fonde sur le seuillage de la différence absolue entre l'image défectueuse et sa restauration, suivi d'une étape de fermeture morphologique automatique. Après une évaluation temporelle des défauts récupérés, nous avons entraîné un réseau U-Net afin de détecter les discontinuités spatio-temporelles pour une détection automatique des défauts.

Dans certains cas, nous pouvons surpasser la détection manuelle à partir de la restauration avec notre réseau. D'autres cas montrent que des améliorations sont encore possibles, soit dans le raffinage des masques créés à partir des images défectueuses et restaurées, avec l'augmentation des données, soit en prenant en compte la compensation de mouvement dans notre réseau via des modèles tenant compte du flux optique [5]. Pour conclure, notre approche vise à poser les bases de travaux plus avancés, et à s'inscrire dans un cadre plus large de restauration de vieux films, en combinaison avec des techniques d'*inpainting* vidéo [19].

Remerciements

Nous tenons à remercier notre partenaire culturel, la Cinémathèque de Toulouse, et tout particulièrement son expert en restauration, qui nous a fourni les données indispensables à l'élaboration de cet article. Cette étude a été réalisée avec le soutien financier du Ministère de la Culture, dans le cadre du programme « Services Numériques Innovants 2019 » et de l'Agence Française de la Recherche dans le cadre du projet PostProdLEAP (ANR-19-CE23-0027-01).

Références

[1] J. Biemond, P. M. B. van Roosmalen, and R. L. Lagendijk. Improved Blotch Detection by Postprocessing. In *Proceedings of ICASSP*, volume 6, 1999.

[2] V. Bruni, P. Ferrara, and D. Vitulano. Color Scratches Removal using Human Perception. In *Proceedings of ICIAR*, volume 5112, 2008.

[3] V. Bruni, D. Vitulano, and A. C. Kokaram. Fast Removal of Line Scratches in Old Movies. In *Proceedings of ICPR*, volume 4, 2004.

[4] K. Chishima and K. Arakawa. A Method of Scratch Removal from Old Movie Film using Variant Window by Hough Transform. In *Proceedings of ISCIT*, 2009.

[5] A. Dosovitskiy, P. Fischer, E. Ilg, P. Hausser, C. Hazirbas, V. Golkov, P. Van Der Smagt, D. Cremers, and T. Brox. Flownet: Learning Optical Flow with Convolutional Networks. In *Proceedings of ICCV*, 2015.

[6] M. K. Gullu, O. Urhan, and S. Erturk. Blotch Detection and Removal for Archive Film Restoration. *AEU - International Journal of Electronics and Communications*, 62(7), 2008.

[7] L. Joyeux, O. Buisson, B. Besserer, and S. Boukir. Detection and Removal of Line Scratches in Motion Picture Films. In *Proceedings of CVPR*, volume 1, 1999.

[8] L. Khriji, M. Meribout, and M. Gabbouj. Detection and Removal of Video Defects using Rational-Based Techniques. *Advances in Engineering Software*, 36(7), 2005.

[9] K.-T. Kim and E. Y. Kim. Automatic Film Line Scratch Removal System Based on Spatial Information. In *Proceedings of the ISCE*, 2007.

[10] A. C. Kokaram. Detection and Removal of Line Scratches in Degraded Motion Picture Sequences. In *Proceedings of EUSIPCO*, 1996.

[11] A. C. Kokaram. On Missing Data Treatment for Degraded Video and Film Archives: a Survey and a New Bayesian Approach. *Transactions on Image Processing*, 13(3), 2004.

[12] A. C. Kokaram. Ten Years of Digital Visual Restoration Systems. In *Proceedings of ICIP*, volume IV, 2007.

[13] A. C. Kokaram, R. D. Morris, W. J. Fitzgerald, and P. J. Rayner. Detection of Missing Data in Image Sequences. *Transactions on Image Processing*, 4(11), 1995.

[14] A. C. Kokaram and P. J. Rayner. System for the Removal of Impulsive Noise in Image Sequences. In *Proceedings of VCIP*, volume 1818 of *Proceedings of the SPIE*, 1992.

[15] M. J. Nadenau and S. K. Mitra. Blotch and Scratch Detection in Image Sequences Based on Rank Ordered Differences. In *Time-Varying Image Processing and Moving Object Recognition, 4*. Elsevier, 1997.

[16] A. Newson, A. Almansa, Y. Gousseau, and P. Pérez. Temporal Filtering of Line Scratch Detections in Degraded Films. In *Proceedings of ICIP*, 2013.

[17] A. Newson, P. Pérez, A. Almansa, and Y. Gousseau. Adaptive Line Scratch Detection in Degraded Films. In *Proceedings of CVMP*, 2012.

[18] J. Ren and T. Vlachos. Segmentation-Assisted Detection of Dirt Impairments in Archived Film Sequences. *Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 37(2), 2007.

[19] A. Renaudeau, F. Lauze, F. Pierre, J.-F. Aujol, and J.-D. Durou. Alternate Structural-Textural Video Inpainting for Spot Defects Correction in Movies. In *Proceedings of SSVM*, 2019.

[20] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional Networks for Biomedical Image Segmentation. In *Proceedings of MICCAI*, volume 9351 of *Lecture Notes in Computer Science*, 2015.

[21] T. K. Shih, L. H. Lin, and W. Lee. Detection and Removal of Long Scratch Lines in Aged Films. In *Proceedings of ICME*, 2006.

[22] R. Sizyakin, N. Gapon, I. Shraifel, S. Tokareva, and D. Bezuglov. Defect Detection on Videos using Neural Network. In *Proceedings of DTS*, volume 132, 2017.

[23] R. Sizyakin, V. Voronin, N. Gapon, M. Pismenskova, and A. Nadykto. A Blotch Detection Method for Archive Video Restoration using a Neural Network. In *Proceedings of ICMV*, volume 11041, 2019.

[24] R. Storey. Electronic Detection and Concealment of Film Dirt. *SMPTE Journal*, 94(6), 1985.

[25] S. Tilie, I. Bloch, and L. Laborelli. Fusion of Complementary Detectors for Improving Blotch Detection in Digitized Films. *Pattern Recognition Letters*, 28(13), 2007.

[26] X. Wang and M. Mirmehdi. Archive Film Defect Detection and Removal: an Automatic Restoration Framework. *Transactions on Image Processing*, 21(8), 2012.

[27] Z. Xu, H. R. Wu, X. Yu, and B. Qiu. Features-Based Spatial and Temporal Blotch Detection for Archive Video Restoration. *Journal of Signal Processing Systems*, 81(2), 2015.

[28] H. Yous and A. Serir. Blotch Detection in Archived Video Based on Regions Matching. In *Proceedings of ISIVC*, 2016.

[29] H. Yous, A. Serir, and S. Yous. CNN-Based Method for Blotches and Scratches Detection in Archived Videos. *Journal of Visual Communication and Image Representation*, 59, 2019.