Détection d'anomalies en temps réel dans le flux vidéo

Fabien Poirier*, Rakia Jaziri** Camille Srour*** Gilles Bernard****

*fab_16@hotmail.fr, **rjaziri@ai.univ-paris8.fr, ***cs@othello.group,
****gb@ai.univ-paris8.fr,

Résumé. De nos jours, de nombreux lieux profitent de la télésurveillance. Mais lorsqu'un incident survient, celle-ci est utilisée dans le but de constater les événements passés. On peut donc la considérer comme un outil de dissuasion plutôt que de détection. Dans cet article, nous allons proposer une approche d'apprentissage automatique profond (deep learning en anglais) visant à résoudre cette lacune. Cette approche utilise des modèles de convolution (CNN) permettant d'extraire des caractéristiques pertinentes liées aux images analysées, qui formeront par la suite des séries temporelles destinées à être traitées par des modèles récurrents.

1 Introduction

Actuellement, la télésurveillance permet de contrôler à distance plusieurs lieux simultanément grâce à un système de caméras disposé dans un espace public ou privé. Les images obtenues par les caméras sont transmises sur un ensemble d'écrans pour être visionnées et analysées, puis archivées ou détruites. Cette surveillance a pour but de contrôler les conditions de sécurité et de sûreté de ces lieux. Généralement, ces images sont analysées par des personnes physiques, ce qui rend la tâche longue et coûteuse. De plus, l'efficacité d'un tel système dépend de l'attention et de la réactivité du surveillant. Avec l'avancée de l'intelligence artificielle dans de nombreux domaines comme celui du traitement d'image, du traitement audio, de la reconnaissance d'actions, ou encore de la détection d'anomalies, il serait plus efficace d'automatiser au moins partiellement cette analyse dans le but d'aider le surveillant dans sa tâche voire de le remplacer. Pour contribuer à cette tâche, tout modèle se doit d'être capable de détecter d'éventuels problèmes le plus rapidement possible avec un maximum de précision, voir d'être capable de les prédire. Nous abordons dans ce qui suit la question de savoir s'il est possible de réaliser un modèle de détection, en temps réel, d'anomalies dans le flux vidéo.

Dans cet article, nous proposons une approche profonde basée sur les modèles récurrents. De par sa nature complexe, une vidéo peut être analysée de trois manières différentes, suivant qu'on prend en compte l'image, le son, ou les deux à la fois. Étant donné que la majorité des vidéos provenant de caméra de surveillance ne contiennent pas de son, ce qui optimise leur stockage, nous avons focalisé nos travaux sur l'analyse d'image. De plus, nous nous concentrerons principalement sur des anomalies pouvant avoir un impact direct sur la sécurité ou la sûreté des personnes présentes dans les vidéos analysées. Après un bref état de l'art, nous présentons notre architecture, nos expérimentations et résultats, avant de conclure.

2 État de l'art

Les réseaux de neurones récurrents Long Short-Term Memory (LSTM) sont très utilisés pour la détection d'anomalies sur des séries temporelles. Proposé par Hochreiter et Schmidhuber (1997), et amélioré par Gers et al. (2000), grâce à l'introduction d'une porte d'oubli, le LSTM est un modèle de réseau de neurones récurrent (RNNs). Les RNNs ont été mis au point initialement par Elman (1990), puis adaptés par Jordan (1997). Contrairement aux simples réseaux feedforward, ils comportent des liens de rétroaction entre les unités qui leur permettent de mémoriser des séries temporelles dynamiques.

Puis en 2015 Tran et al. (2015) ont développé C3D (Convolution 3 Dimensions), un modèle efficace pour extraire des critères spatio-temporels dans le flux vidéo dans le cadre de la reconnaissance d'actions, et plus précisément, de sports. La convolution 3D se base sur des séquences d'images successives afin d'analyser les différences entre celles-ci. Cette technique a montré de très bons résultats dans de nombreux domaines tels que l'estimation de la pose humaine, la segmentation d'images médicales, la reconnaissance d'actions ou encore dans la vidéo-surveillance.

Comme le montrent les travaux de Sultani et al. (2018) ou encore ceux de Aberkane et Elarbi (2019), grâce à sa 3e dimension, C3D permet d'extraire des caractéristiques pertinentes à partir de mouvements. Il est donc généralement utilisé en tant qu'extracteur de caractéristiques dans les modèles de détection d'anomalie et de violence dans les flux vidéo.

Plus récemment, Majd et Safabakhsh (2019) ont utilisé un réseau appelé ConvLSTM (Convolutional Long Short Term Memory), combinant l'extraction de caractéristiques des réseaux à convolution (CNN), technique très efficace sur les images, et la mémoire du LSTM afin de traiter des séries temporelles d'images. Ce nouveau réseau sensible au mouvement est parfaitement adapté pour la reconnaissance d'actions dans les données vidéo.

Pour finir, en 2019, plusieurs articles dédiés à ces domaines Berenguer et al. (2019) Ghani et al. (2019) Peixoto et al. (2019) ont montré que les deux architectures, à savoir C3D et ConvLSTM, étaient toutes les deux pertinentes. Il est vrai qu'au vu des résultats, les performances de C3D semblent légèrement supérieures à celles de convLSTM, en revanche il est bien plus complexe à mettre en oeuvre. De plus, selon l'article Convertini et al. (2020), les modèles de type convLSTM ont quand même de très bons résultats.

3 L'approche proposée

3.1 Préparation

La plupart des vidéos collectées présentant une anomalie ont été découpées à la main dans le but de ne garder que celle-ci. Pour chaque vidéo, une séquence de x images a été formée avec un pas constant entre chaque image afin que la séquence extraite couvre bien l'intégralité de la vidéo. (FIG . 1) De plus, nous avons augmenté les données avec des zooms, des recadrages, ou des effets miroir (horizontal flip). La taille des données étant trop importante pour être chargée en mémoire, nous avons utilisé un générateur pour réaliser cette tâche.

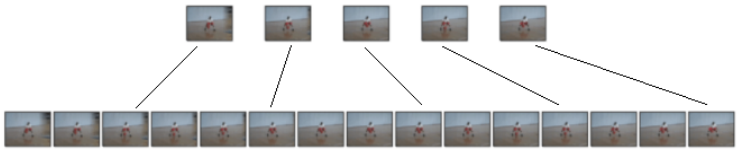

FIG. 1 – *Constitution de la séquence medium.com/smileinnovation/training-neural-network-with-image-sequence-an-example-with-video-as-input-c3407f7a0b0f*

3.2 Architecture

Le modèle proposé en figure 2 fonctionne aussi bien sur des vidéos provenant de jeux de données ou en flux continu comme avec une webcam. Avant chaque analyse, il est possible de paramétrer la séquence de prédiction voulue (séquence exprimée en secondes), le modèle va alors calculer automatiquement le pas à appliquer entre les images en fonction du nombre de fps de la vidéo. Une fois ce pas calculé l'extraction des images peut commencer. Pour la phase d'extraction, nous avons opté pour un modèle VGG, plus particulièrement VGG19. VGG est un modèle proposé par Simonyan et Zisserman (2014) reconnu pour ces performances dans le domaine de la vision par ordinateur ; il a notamment gagné la compétition ILSVRC (ImageNet Large Scale Visual Recognition Challenge) en 2014 en atteignant une précision de 92.7% sur le jeu de données imageNet.

Ce modèle est composé de 23 couches groupées en 5 principaux blocs composés chacun de 2-3 couches de convolution suivies d'une couche de pooling (FIG 3). Dans notre architecture, nous l'avons partiellement ré-entrainé à partir de la 12e couche représentant le milieu du modèle et le 4e bloc. Étant donné que cette convolution est 2D elle n'est pas du tout adaptée pour des séquences d'images. Chaque image extraite va donc être redimensionnée à une taille de 112*112 puis donnée une par une au réseau dans le but d'y extraire des caractéristiques pertinentes. Concernant notre batch nous l'avons fixé à 16.

Une fois ces caractéristiques extraites, ces données seront passées à notre couche Time Distributed. Contrairement à une convolution 2D présentant une perte d'information à cause d'une fusion de toutes nos images pour les faire tenir sur un format 2D. La couche Time Distributed va permettre d'accumuler les caractéristiques venant d'images différentes dans le but de former notre séquence de prédiction. Pour chaque séquence, nous avons fixé une taille de 30 images en nous basant sur la moyenne actuelle (30 fps), la taille des données d'entrée de la couche Time Distribued est donc de 30*112*112*3. L'introduction d'un modèle GRU à notre architecture nous permet de mettre en place un historique nous permettant de mieux traiter nos séquences. Ce type de modèle est composé de 2 portes : une porte d'oubli et une porte de mise à jour (FIG 4). Pour commencer, la donnée va passer par la porte d'oubli, porte qui va permettre de contrôler le nombre d'informations qui doit être oublié puis les informations gardées, jugées pertinentes sont données à la porte de mise à jour pour être apprise. Porte qui a pour rôle de concaténer les nouvelles données avec celle de l'état précédent puis de les passer par une fonction sigmoïde pour détecter les composantes importantes. Pour finir afin d'effectuer notre classification nous avons ajouté à notre modèle quelques couches totalement connectées et

quelques couches d'oubli. Concernant la fonction loss étant donné notre nombre de classes nous avons choisi la fonction binary cross-entropy, mais à terme vu que certaines anomalies peuvent survenir en parallèle, nous opterons pour une categorical cross-entropy. Concernant l'optimizer nous en avons testé une grande variété et celui avec lequel nous avons obtenu les meilleurs résultats a été SGD, mais il nous a fallu réduire le coefficient d'apprentissage.

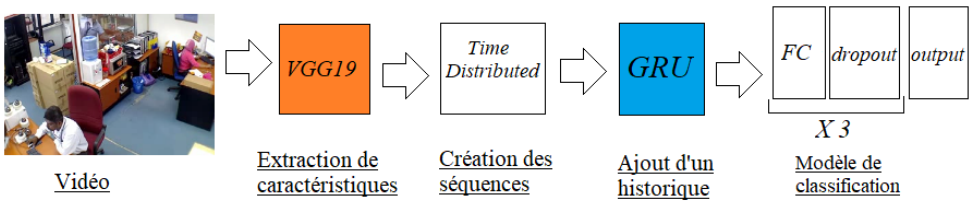

FIG. 2 – *Architecture de l'approche proposée*

FIG. 3 – *VGG19 architecture*
www.quora.com/What-is-the-VGG-neural-network

4 Expérimentations

4.1 Jeux de données

Notre jeu de données contient +4000 vidéos regroupées en 13 classes parmi lesquelles on retrouve : bagarres, incendies, accidents de voiture ou encore vidéos sans anomalie. Nous avons donc des scènes très variées : vidéo de jour comme de nuit, provenant de différentes sources telles qu'un téléphone, une caméra de surveillance, une caméra de télévision, ainsi que différents angles de vue.

4.2 Démarche

Nous avons réalisé de nombreuses expérimentations suivant 3 axes principaux : définir un jeu de données correctes, trouver une préparation adéquate pour nos données et définir le

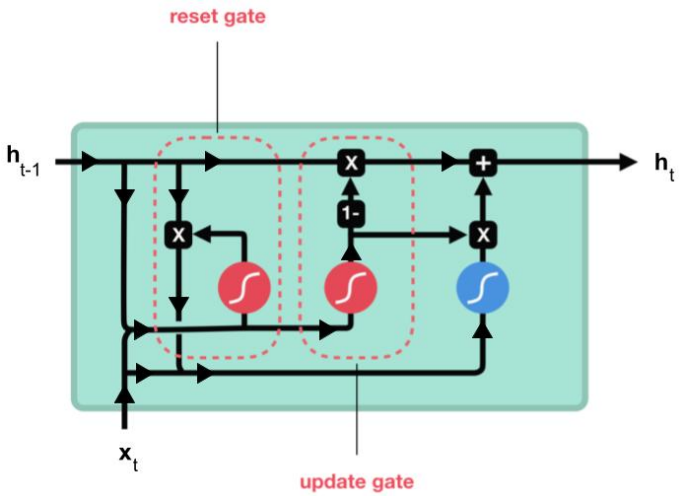

FIG. 4 – *GRU architecture*
penseeartificielle.fr/comprendre-lstm-gru-fonctionnement-schema/

modèle à utiliser. Nous avons commencé par entraîner un modèle sur l'ensemble de nos classes, mais ces données étant massives, le temps d'apprentissage s'en est retrouvé impacté. De plus, les données étant assez différentes les unes des autres, les résultats obtenus à l'heure actuelle ne se sont pas montrés à la hauteur de nos attentes. Nous avons donc préféré travailler sur un sous-échantillon de notre jeu de données à savoir 2 classes : Bagarre / Normal, que nous avons séparé aléatoirement selon une répartition 60/40.

Le 2e axe abordé a été de savoir si nous devions réaliser une seule séquence résumant l'intégralité de la vidéo, ou plusieurs séquences formées à partir d'images successives. Nous avons donc confronté ces deux approches sur des vidéos où seules les anomalies ont été conservées et des vidéos non modifiées.

Le dernier axe de recherche a été de trouver le modèle optimal. Pour cela, nous avons testé 5 réseaux à convolutions différents, dont 3 convolutions de notre conception ainsi que vgg19 et mobilnet. Pour la partie classification, nous avons aussi comparé les performances d'un LSTM et d'un GRU.

4.3 Résultats

Dans cette section, nous commencerons par présenter l'ensemble des résultats obtenus sur notre jeu de test sous forme de tableaux. Étant donné que nous n'avons pas accès à de véritables caméras de surveillance, chacun de nos tests a été réalisé sur un ensemble de vidéos fini et une seule prédiction a été effectuée. Puis, nous montrerons les performances de notre approche sur des vidéos représentant des manifestations anti-pass sanitaire (vidéos non incluses dans le jeu d'apprentissage et de test) pour lesquels nous effectuerons une prédiction toutes les 30 images.

Modèle	Accuracy	Précision	Recall	F1-Score
Conv (3 blocs) + GRU + pred modèle	78.8%	80%	75.9%	78.1%
Conv (5 blocs) + GRU + pred modèle	85.1%	85.7%	84.2%	85%
Conv (8 blocs) + GRU + pred modèle	80.1%	78%	83.7%	80.7%
Mobilnet + GRU + pred modèle	85.9%	86.9%	84.4%	85.7%
VGG19 + GRU + pred modèle	86.5%	87.1%	85.7%	86.4%
VGG19 + LSTM + pred modèle	85.1%	84.3%	86.3%	85.3%
VGG19 + GRU	85%	86.1%	83.4%	84.7%

TAB. 1 – *Résultats des différents modèles sur le jeu de test*
Un bloc représente : une ou deux couches de convolution + une couche de pooling

FIG. 5 – *Exemple de résultat : détection de la classe fight / normal*

FIG. 6 – *Travaux en cours*

D'une manière générale, nous sommes satisfaits des résultats obtenus. Notre modèle est capable de détecter plusieurs formes de "violence" telles que des bagarres impliquant une ou plusieurs personne, armé ou non, ainsi que des jets de projectiles. De plus malgré la latence lié à la première prédiction du fait que notre modèle soit un modèle de détection et non de prédiction, celui-ci fournit quand même des résultats rapides comparables à du temps réel. Par contre étant donné qu'une prédiction attribut une seule étiquette à l'ensemble des images d'une séquence, cela cause par fois des erreurs visuelles, car il est possible dans une même séquence

d'avoir plusieurs actions opposées : neutre ou anomalie.

5 Conclusion et travaux futurs

Dans cet article, nous avons montré qu'il était possible d'utiliser l'apprentissage profond pour détecter en temps réel avec une assez grande précision des anomalies provenant d'un flux vidéo, telle que des bagarres. Le système proposé a été mis en œuvre en utilisant des réseaux de neurones convolutifs entraînés sur des images extraites de vidéos, combinés à des réseaux permettant d'analyser des séries temporelles. L'idée initiale étant atteinte, la perspective principale est maintenant de faire en sorte que le modèle présenté soit capable de traiter plusieurs types d'anomalies.

Par la suite, il serait intéressant de comparer notre approche à une approche utilisant un modèle C3D. Il serait aussi possible d'ajouter de la détection d'objet ou d'analyse le son présent dans les vidéos afin d'améliorer la détection de certaines anomalies. Où encore élaborer un modèle capable de prédire les anomalies quelques secondes avant que celles-ci n'aient lieu, ce qui facilitera l'intervention des personnes concernées.

Références

Aberkane, S. et M. Elarbi (2019). Deep reinforcement learning for real-world anomaly detection in surveillance videos. In *2019 6th International Conference on Image and Signal Processing and their Applications (ISPA)*, pp. 1–5. IEEE.

Berenguer, A. D., M. C. Oveneke, M. Alioscha-Perez, et H. Sahli (2019). Paired supervised learning and unsupervised pretraining of cnn-architecture for violence detection in videos. In *BNAIC/BENELEARN*.

Convertini, N., V. Dentamaro, D. Impedovo, G. Pirlo, et L. Sarcinella (2020). A controlled benchmark of video violence detection techniques. *Information 11*(6), 321.

Elman, J. L. (1990). Finding structure in time. *Cognitive science 14*(2), 179–211.

Gers, F. A., J. Schmidhuber, et F. Cummins (2000). Learning to forget : Continual prediction with lstm. *Neural computation 12*(10), 2451–2471.

Ghani, R. F. et al. (2019). Robust real-time fire detector using cnn and lstm. In *2019 IEEE Student Conference on Research and Development (SCOReD)*, pp. 204–207. IEEE.

Hochreiter, S. et J. Schmidhuber (1997). Long short-term memory. *Neural computation 9*(8), 1735–1780.

Jordan, M. I. (1997). Serial order : A parallel distributed processing approach. In *Advances in psychology*, Volume 121, pp. 471–495. Elsevier.

Majd, M. et R. Safabakhsh (2019). A motion-aware convlstm network for action recognition. *Applied Intelligence 49*(7), 2515–2521.

Peixoto, B., B. Lavi, J. P. P. Martin, S. Avila, Z. Dias, et A. Rocha (2019). Toward subjective violence detection in videos. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 8276–8280. IEEE.

Simonyan, K. et A. Zisserman (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv :1409.1556*.

Sultani, W., C. Chen, et M. Shah (2018). Real-world anomaly detection in surveillance videos. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 6479–6488.

Tran, D., L. Bourdev, R. Fergus, L. Torresani, et M. Paluri (2015). Learning spatiotemporal features with 3d convolutional networks. In *Proceedings of the IEEE international conference on computer vision*, pp. 4489–4497.

Summary

Nowadays, many places use security cameras. When an incident occurs, this technology is only used a posteriori. So it can be considered more as a deterrence tool than as a detection tool. In this article, we will propose a deep learning approach in order to solve this issue. Our approach uses convolutional models (CNN) to extract relevant characteristics linked to the video images; these characteristics will form times series to be analyzed by LSTM / GRU models.