Détection temporelle de saillance dynamique dans des vidéos par apprentissage profond

Léo Maczyta, Patrick Bouthemy, Olivier Le Meur

Détection temporelle de saillance dynamique dans des vidéos par apprentissage profond

Léo Maczyta[1] Patrick Bouthemy[1] Olivier Le Meur[2]

[1] Inria, Centre Rennes - Bretagne Atlantique
[2] Univ Rennes, CNRS, IRISA Rennes

leo.maczyta@inria.fr

Résumé

Le problème étudié concerne l'analyse de la saillance dynamique dans des vidéos. Plus précisément, nous cherchons à chaque instant à déterminer si une image peut être classée comme saillante ou non selon des informations liées au mouvement dans l'image. Une image sera détectée comme saillante si elle contient des éléments dont le mouvement se démarque de son contexte spatio-temporel. L'approche proposée traite les cas où la caméra est en mouvement et utilise l'apprentissage profond. Plusieurs variantes en sont proposées et comparées. La détection temporelle est pertinente pour de nombreuses applications où une alerte ou une attention particulière doit être déclenchée en rapport avec le contenu dynamique des vidéos. Des expérimentations sur une base de vidéos réelles variées montrent la très bonne précision de classification obtenue.

Mots Clef

Saillance dynamique, Estimation du mouvement, Apprentissage profond, Classification d'image

Abstract

We address the problem of motion saliency in videos. More precisely, we aim to determine at each time step if an image can be classified as salient or not according to its motion content. An image will be detected as salient if it contains objects whose motion departs from its spatio-temporal context. The proposed approach handles situations with a mobile camera and involves a deep learning stage. Several variants are proposed and compared. Temporal saliency detection is relevant for applications that require to trigger alerts or to monitor dynamic behaviours from videos. Experiments on real videos demonstrate that the proposed methods can provide accurate classification.

Keywords

Dynamic saliency, Motion estimation, Deep learning, Image classification

1 Introduction

Nous abordons le problème de la saillance dynamique dans des vidéos définie comme la présence d'un contenu spatio-temporel se démarquant de son contexte et ce principalement par son mouvement propre. L'information de saillance dynamique permet de repérer la présence d'objets au mouvement singulier, inattendu ou de comportement rare. La saillance dynamique est une information pertinente pour des tâches qui requièrent d'appréhender un environnement en mouvement, la caméra pouvant être elle-même en mouvement.

Dans cet article, nous nous intéresserons plus précisément à un objectif de classification. Nous voulons déterminer pour chaque image d'une séquence vidéo si elle doit être classée comme saillante ou non, au sens où elle contient ou non des éléments dynamiquement saillants. Nous ne chercherons pas à ce stade à estimer des cartes de saillance dynamique. Une telle estimation est certes plus élaborée et peut permettre une localisation spatiale, mais, d'un autre côté, la classification image par image apporte une décision explicite et souvent suffisante. Ce type de problème n'a pas encore été abordé à notre connaissance, mais il est pourtant crucial pour de nombreuses applications, comme l'irruption d'obstacles en robotique mobile ou pour des véhicules autonomes, la détermination d'alertes en vidéosurveillance, le déclenchement d'attention pour un diagnostic sur vidéos, ou le pointage sur des informations pertinentes pour l'élaboration de résumés de vidéos. Nous proposerons une approche originale à ce problème de classification (que l'on peut nommer aussi détection temporelle) de la saillance dynamique dans des vidéos, approche qui s'appuiera sur un apprentissage profond.

L'article est organisé comme suit. Dans la section 2, un rapide état de l'art des méthodes de saillance dynamique est fourni. Des méthodes d'apprentissage profond pouvant être appliquées à ce problème sont également discutées. Dans la section 3, les bases de vidéos, synthétiques et réelles, constituées pour l'évaluation de notre approche de classification sont présentées. Dans la section 4, nous décrivons notre approche et les différentes méthodes que nous avons développées. Les résultats expérimentaux sont rassemblés

et commentés dans la section 5. Enfin, la section 6 contient la conclusion.

2 État de l'art sur la saillance dynamique

L'étude de la saillance spatio-temporelle ou saillance dynamique s'est tout d'abord concentrée sur la mise en évidence d'objets en mouvement dans des scènes vues par une caméra statique [19]. La scène peut néanmoins inclure des artefacts (ou textures) dynamiques [4], telles que de la végétation ou des drapeaux oscillant dans le vent, des ondulations à la surface de l'eau, qu'il ne s'agit pas de considérer comme saillants [30]. De façon plus générale, la scène peut être observée par une caméra en mouvement. Une première catégorie de méthodes cherche à compenser la composante du mouvement dans l'image due à la caméra mobile [10]. Un second type d'approches consiste à combiner des informations spatiales et temporelles [6, 15, 20, 31]. Karimi et al. [14] utilisent des indices spatio-temporels et représentent les vidéos comme des graphes spatio-temporels afin de minimiser une fonction globale. Un problème voisin est la détection d'anomalies dans des foules [24].

L'apprentissage profond a permis d'obtenir d'importantes avancées dans le domaine du traitement d'image, notamment en classification d'image [9, 17, 29]. Des applications en traitement vidéo, telles que le calcul du flot optique [11], la segmentation d'objets mobiles [30], la description automatique de vidéos [34], ou encore la reconnaissance d'actions dans des vidéos [18, 26, 27] ont montré la pertinence de l'application des réseaux de neurones convolutionnels aux vidéos également. Les architectures existantes sont variées, avec par exemple [27] qui introduit une architecture « deux flux » pour la reconnaissance d'actions.

L'étude de la saillance dans des vidéos à l'aide de méthodes d'apprentissage profond n'a été entreprise que récemment. Dans [3], Chaabouni *et al.* cherchent à estimer la saillance définie par des cartes de fixation du regard. Cette définition n'est pas équivalente à celle qui nous intéresse, puisque le regard peut être attiré par des éléments statiques de la scène, tels que des personnes. Des travaux proches concernent la détection d'anomalies dans des scènes ou des foules [7, 33, 35]. Dans [32], les auteurs s'intéressent au cas plus général de l'obtention de cartes de saillance spatio-temporelle en exploitant mouvement et apparence.

La méthode proposée dans cet article cherche à déterminer si chaque image issue d'une vidéo peut être déclarée saillante ou non, c'est-à-dire comportant ou non une information saillante au sens du mouvement, indépendamment de l'apparence et de la nature des objets de la scène.

3 Description des bases de vidéos

Dans cette section, nous introduisons les bases de vidéos synthétique et réelle que nous avons employées dans nos expérimentations.

3.1 Base d'exemples synthétiques

Les méthodes d'apprentissage en général, et en particulier celles basées sur l'apprentissage profond, requièrent un jeu de données d'apprentissage suffisamment important [8, 28]. Il a été d'autre part montré dans de nombreuses applications de vision par ordinateur que le recours à des données synthétiques s'avère judicieux. Aussi, nous avons construit une base de vidéos synthétiques pour une première phase d'entraînement des réseaux.

Chaque élément de la base consiste en une paire d'images formée d'une image originale et d'une seconde image obtenue par application à la première d'un modèle de mouvement paramétrique, en l'occurrence affine. Des éléments saillants formés de portions d'autres images subissant un mouvement affine différent de ce mouvement principal sont ensuite inclus. Les images de PascalVOC 2012 [5] ont servi pour la génération de cette base. Des exemples sont donnés en figure 1. Les exemples sont générés avec une probabilité de 0,5 pour l'absence de saillance, 0,25 pour la présence d'un élément saillant et 0,25 pour la présence de deux éléments saillants. Les éléments saillants ont une dimension limitée, de l'ordre de 0,5% à 1,5% de la surface de l'image. Nous avons délibérément inclus des éléments saillants de petite taille pour constituer des exemples de saillance difficiles à détecter.

En générant les exemples de cette façon, le risque serait que la méthode proposée distingue les éléments saillants de par leur aspect généralement différent du reste de l'image, et non du fait de leur véritable saillance dynamique. Pour éviter ce problème, les exemples non saillants peuvent contenir jusqu'à deux portions issues d'autres images, qui suivent cette fois le mouvement principal. La forme des portions d'images incluses a été générée aléatoirement.

Afin de maximiser la variabilité des exemples servant à entraîner les réseaux, les paires d'images sont générées à la volée pendant l'apprentissage. Pour la phase d'apprentissage, nous sommes amenés à générer à la volée environ 4 millions d'éléments. Les ensembles de validation et de test contiennent 2000 éléments chacun.

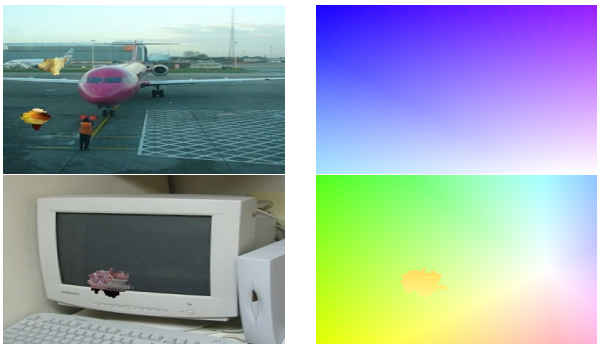

FIGURE 1 – Exemples tirés de la base de vidéos synthétiques. Les champs de vitesse associés sont représentés en code couleur, teinte pour la direction du vecteur, saturation pour son amplitude. L'image du haut est non saillante et celle du bas est saillante.

3.2 Base de vidéos réelles

Pour pouvoir évaluer notre méthode de classification de la saillance dynamique, il est nécessaire de disposer en outre d'une base de vidéos réelles. En particulier, puisque la méthode proposée s'applique à des situations générales comportant des caméras mobiles, il faut que la caméra soit effectivement en mouvement dans une part conséquente des vidéos traitées.

La vérité-terrain qui nous intéresse doit porter sur la présence de saillance liée au mouvement dans chaque image. La base de vidéos constituée par Bideau et Learned-Miller [1] est ainsi une bonne opportunité. Elle rassemble les bases FBMS-59 [22], Complex Background [21] et Camouflaged Animals [2], ré-annotées spécifiquement pour le problème de la segmentation d'objets mobiles. Cette base de vidéos comprend notamment des exemples adaptés à notre problème, en particulier les vidéos extraites de Camouflaged Animals, où visuellement seule l'information de mouvement permet de détecter l'animal saillant. Notre annotation de classe saillante pour une image résulte de la présence d'un élément saillant dans la vérité-terrain de cette base. Il faut noter que cette vérité-terrain n'est disponible que pour une partie des images des vidéos.

Toutefois, notre objectif n'étant pas de localiser spatialement les éléments mobiles saillants mais de détecter temporellement leur présence, des exemples de séquences d'images non saillantes sont nécessaires pour servir d'exemples négatifs. De tels exemples étant très minoritaires dans la base de [1], nous l'avons complétée avec 71 nouvelles vidéos exclusivement non saillantes, que nous avons nous-mêmes acquises.

La base de vidéos finale contient 144 vidéos, dont 94 dans l'ensemble d'apprentissage, 13 dans l'ensemble de validation et 37 dans l'ensemble de test, pour un total de 3451 paires d'images. Ces trois ensembles sont globalement équilibrés en terme d'images saillantes et non saillantes. Pendant l'apprentissage, nous procédons à une augmentation de données par redimensionnement, par « cropping », par inversion temporelle, ainsi que par symétrie autour d'un axe vertical (flip). De plus, la vérité-terrain étant disponible pour toutes les images des 71 nouvelles vidéos, systématiquement non saillantes, toutes les images de ces vidéos peuvent être utilisées pendant l'apprentissage. Les données pour la construction des batchs introduits dans l'apprentissage des réseaux seront par la suite choisies de façon à garantir un équilibre entre le nombre d'éléments saillants et non saillants.

4 Classification de la saillance dynamique

Dans cette section, nous allons décrire notre approche de classification image par image de la saillance dynamique, que nous pouvons aussi appeler approche de détection temporelle de la saillance dynamique. Elle s'appuie sur l'utilisation de réseaux de neurones convolutionnels (CNN en anglais, sigle que nous utiliserons à l'occasion pour simplifier l'écriture) qui ont montré toute leur efficacité ces dernières années sur des problèmes de classification d'image.

Nous en avons conçu plusieurs variantes. Tout d'abord, nous avons défini deux manières de poser ce problème de classification de saillance dynamique :

— La première compare l'image courante et l'image suivante recalée par le mouvement principal (ou dominant) estimé entre ces deux images ;
— La seconde se démarque des images et exploite la différence (ou flot résiduel) entre flot optique et mouvement dominant calculés entre l'image courante et sa suivante.

Nous avons comparé deux façons de calculer le modèle paramétrique de mouvement correspondant au mouvement dominant dans l'image. La première utilise la méthode robuste incrémentale Motion 2D [23]. Pour la seconde, nous avons élaboré une méthode par apprentissage profond.

4.1 Classification de la saillance après alignement des images successives

Dans la très grande majorité des cas, le mouvement induit par une caméra mobile correspond au mouvement dominant dans l'image. Ce dernier est exprimé sur les régions de l'image correspondant aux éléments statiques de la scène. Si les variations de profondeur ou d'orientation de ces éléments statiques sont faibles par rapport à la distance à la caméra, un seul modèle paramétrique 2D de mouvement, comme un modèle affine ou quadratique, pourrait raisonnablement appréhender ce mouvement dominant. Dans le cas contraire, il faudrait en introduire plusieurs, et procéder à une segmentation de la scène pour affecter un modèle de mouvement à chaque zone segmentée. Pour valider l'approche proposée, nous nous focaliserons à ce stade sur des situations où le recours à un seul modèle paramétrique de mouvement peut être suffisant. Notons par ailleurs que si la caméra effectue un plan rapproché sur un élément mobile de la scène, le mouvement dominant devient celui de cet objet en gros plan. Pour autant, notre détection de la saillance dynamique restera opérante, car il s'agira toujours de détecter une présence d'un élément se démarquant de ce mouvement dominant.

Une fois les images alignées à partir du modèle affine estimé, la classification est réalisée par un réseau convolutionnel. Les deux images pré-alignées étant des images en couleur, nous formons une entrée du réseau à 6 canaux. Le schéma de principe de cette première architecture de classification de la saillance dynamique est fourni à la figure 2. La structure du CNN pour la classification a été définie à l'aide d'expériences préliminaires menées sur la base de données synthétiques. Cette architecture est détaillée à la figure 3. Lors d'expérimentations préalables, nous avons observé qu'introduire une plus grande profondeur dans le réseau conduisait à du sur-apprentissage. L'entraînement de ce CNN est réalisé avec la fonction d'entropie croisée comme fonction de perte.

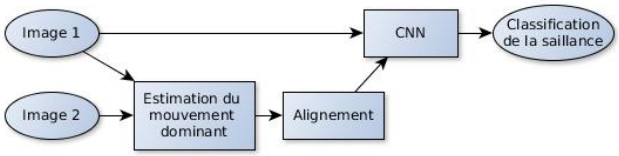

FIGURE 2 – Classification de la saillance dynamique basée sur l'alignement préalable des images à partir de l'estimation du mouvement dominant.

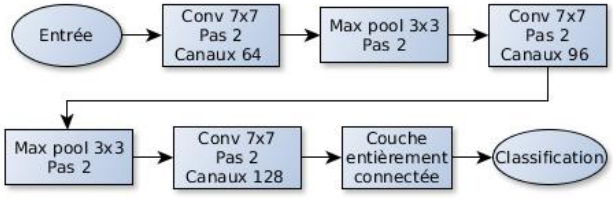

FIGURE 3 – Réseau convolutif pour la classification de la saillance dynamique après alignement des images. Les convolutions sont suivies d'une « batch normalisation » et de la non-linéarité ReLU.

La compensation du mouvement est réalisée avec un modèle paramétrique affine qui s'écrit pour chaque point $p \in \Omega$, où Ω est le domaine de l'image ;

$$\omega_\theta(p) = \begin{pmatrix} a_1 + a_2 x + a_3 y \\ a_4 + a_5 x + a_6 y \end{pmatrix} \quad (1)$$

où $p = (x, y)$, et $\theta = (a_1, ..., a_6)$ forme le vecteur des paramètres du modèle de mouvement.
Une première mise en œuvre de cette classification de saillance dynamique par alignement préalable des images exploitera l'algorithme robuste et incrémental Motion2D [23] pour l'estimation paramétrique du mouvement dominant. Cette méthode sera notée par la suite WS-Motion2D pour Warping Saliency with Motion2D.

4.2 Estimation du mouvement dominant par apprentissage profond

Dans la section précédente (méthode WS-Motion2D), l'estimation du modèle paramétrique de mouvement est obtenue par une méthode classique de régression robuste. Il est toutefois intéressant d'en étudier le remplacement par un module basé également sur l'apprentissage profond, pour envisager un schéma d'apprentissage de bout en bout.
Dans [25], Rocco *et al.* proposent un réseau convolutionnel pour le réalignement, entre deux images prises selon des points de vue éventuellement très différents, d'instances différentes d'objets d'une même classe, par exemple deux images de voitures. Il conduit à l'estimation paramétrique d'une transformation géométrique entre images.
Nous avons repris cette architecture et l'avons adaptée à l'estimation d'un modèle de mouvement 2D affine entre images successives d'une vidéo. Dans notre cas, un aspect important est la présence d'outliers qui ont un mouvement différent du mouvement dominant. De plus, les mouvements attendus sont nettement moins marqués. Aussi, l'amplitude maximale des déplacements générés dans la base synthétique a été fixée à 15% des dimensions de l'image. Cette architecture, schématisée à la figure 4, a été entraînée à l'aide des exemples synthétiques décrits au paragraphe 3.1. Dans cette architecture, le CNN est composé de deux couches convolutionnelles de supports respectifs 7x7 et 5x5, suivies d'une couche complètement connectée pour la classification.
La fonction de perte utilisée pour entraîner ce réseau est similaire à celle utilisée dans [25]. Une grille de points $\mathcal{G}$ est déformée par le mouvement dominant estimé de paramètres $\hat{\theta}$ et par la vérité-terrain θ_{GT}. La fonction de perte $\epsilon(\hat{\theta})$ compare ces deux ensembles de déplacements produits et s'écrit :

$$\epsilon(\hat{\theta}) = \frac{1}{N} \sum_{p \in \mathcal{G}} ||\omega_{\theta_{GT}}(p) - \omega_{\hat{\theta}}(p)||_2^2 \quad (2)$$

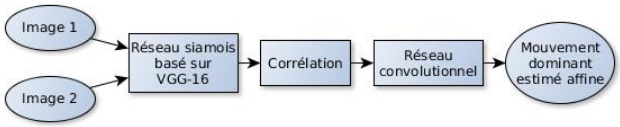

FIGURE 4 – Architecture inspirée de [25] pour l'estimation paramétrique du mouvement dominant.

En utilisant ce réseau, nous avons obtenu une erreur moyenne d'estimation du mouvement dominant sur l'ensemble de test synthétique de 0,20 pixel, avec un écart-type de 0,08. En ce qui concerne Motion2D, l'erreur moyenne est de 0,03 pixels pour un écart-type de 0,41.
Sur cet ensemble de test synthétique, Motion2D fournit une estimation du mouvement plus proche de la vérité-terrain, mais la précision obtenue par le réseau reste raisonnable. Nous pouvons penser que la précision diminuera sur les vidéos réelles et que le réseau aura *a priori* plus la capacité de s'adapter aux situations réelles. En remplaçant Motion2D par ce réseau, nous avons une nouvelle variante de la méthode de classification que nous nommerons par la suite WS-Fully Deep Learning ou WS-FD.
Il est à noter que pour la variante s'appuyant entièrement sur l'apprentissage profond, le réseau d'estimation du mouvement dominant a été pré-entraîné séparément puis figé par la suite. Il serait toutefois possible de réaliser un entraînement de bout en bout en adoptant une méthode de rétro-propagation de l'opération de réalignement, comme il est suggéré dans [12].

4.3 Classification de la saillance à partir du flot optique résiduel

La seconde méthode proposée part de la constatation que la saillance dynamique est par définition liée au mouvement. Au lieu de mettre en entrée du classifieur les images, l'idée est cette fois d'exploiter directement des informations liées au mouvement. Cela permet de s'affranchir complètement des aspects d'apparence. Nous calculons tout d'abord le flot optique résiduel, en faisant la différence du flot optique et du flot affine dû au mouvement dominant estimé $\hat{\theta}$:

$$\forall p \in \Omega, \qquad \omega_{res}(p) = \omega(p) - \omega_{\hat{\theta}}(p) \tag{3}$$

Ce flot résiduel est placé en entrée du classifieur.
Le schéma de principe de cette méthode est décrit à la figure 5. Les composantes du flot optique forment les deux canaux de l'entrée du classifieur.

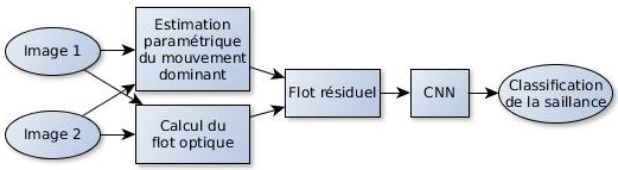

FIGURE 5 – Classification de la saillance basée sur le flot résiduel.

Le flot optique est calculé à l'aide de l'algorithme FlowNet2.0 [11]. FlowNet2.0 a été choisi en raison de ses bonnes performances, mais aussi en raison de son très faible temps de calcul. La même architecture de classifieur que celle décrite à la figure 3 est utilisée.
Une première version de cette méthode d'estimation de la saillance dynamique sera notée par la suite RFS-Motion2D pour « Residual Flow Saliency with Motion2D » dans le cas où Motion2D est utilisé pour l'estimation paramétrique du mouvement. Une seconde version exploitant le réseau présenté à la section précédente pour l'estimation du mouvement dominant sera dénommée RFS-FD pour RFS-Fully Deep Learning.

5 Évaluation expérimentale

5.1 Implémentation des méthodes

Les méthodes présentées dans la section 4 ont été implémentées à l'aide de la librairie caffe [13]. L'optimisation a été réalisée à l'aide de la méthode Adam [16] avec les paramètres proposés par les auteurs. Le taux d'apprentissage initial a été fixé à 10^{-3} pour l'apprentissage sur données synthétiques et à 10^{-5} pour les données réelles [1]. Le temps nécessaire pour le traitement d'un batch dans la phase d'apprentissage (prédiction et rétro-propagation), avec un GPU Tesla M40 et un processeur cadencé à 3,9 GHz, est respectivement de 1,4 sec, 1,8 sec, 0,7 sec et 1,2 sec pour WS-FD, WS-Motion2D, RFS-FD et RFS-Motion2D. La taille des batchs est de 32, 32, 8 et 12 éléments respectivement. La prédiction pour une image dans la phase de test se réalise respectivement en 20,0 fps, 15,2 fps, 10,4 fps et 9,5 fps.

5.2 Remarques préliminaires

Les quatre méthodes introduites ci-dessus ont d'abord été entraînées sur la base d'exemples synthétiques présentée au paragraphe 3.1. L'évaluation sur la base de validation synthétique a permis d'atteindre une précision dépassant les 98% dans tous les cas. Une performance de ce type s'explique par le caractère idéal des données synthétiques.

1. Des expériences ultérieures semblent indiquer que des performances équivalentes peuvent être obtenues plus rapidement avec un taux d'apprentissage à 10^{-3} pour les données réelles.

Il est un aspect du problème que nous n'avons pas encore abordé, le choix du pas de temps pour le test, c'est-à-dire l'écart temporel entre les deux images prises en compte. Un pas de temps de 1 signifie que nous prenons deux images consécutives de la vidéo, un pas de temps de 2, l'image t et l'image $t+2$, etc... En effet, il est plausible que prendre des images plus espacées dans le temps devrait faciliter la détection d'éléments saillants dont le mouvement est faible. Cependant, l'estimation paramétrique du mouvement dominant est censée être plus performante sur des images proches, moins susceptibles de comporter de grands mouvements de caméra. Un compromis sur le pas de temps sera donc à trouver comme expliqué plus loin.

5.3 Définition de deux méthodes de référence

À notre connaissance, le problème que nous étudions n'a pas été traité sous cette forme jusqu'à présent, ce qui rend impossible les comparaisons directes avec des méthodes existantes. Afin de cerner la difficulté du problème, deux méthodes de référence « naïves » sont proposées.
La première méthode considère une différence d'intensités recalées comme dans le paragraphe 4.1. Pour ce faire, un critère $\mathcal{C}_1$ est défini de la façon suivante :

$$\mathcal{C}_1 = \frac{1}{|\Omega|} \sum_{p \in \Omega} |I(p + \omega_{\hat{\theta}}(p), t+1) - I(p,t)| \tag{4}$$

où $\hat{\theta}$ correspond aux paramètres du mouvement dominant estimé, ω_θ représente le flot affine dominant, I est l'intensité de l'image, Ω est le domaine de l'image et p désigne un de ses points. De façon à éviter des effets de bord, les pixels sur les bords qui sortent de l'image ne sont pas inclus dans Ω. Motion2D est utilisé pour l'estimation du mouvement dominant.
La distribution empirique de ce critère $\mathcal{C}_1$ est calculée sur l'ensemble de validation de la base de données réelles. Cette distribution est identifiée à la loi exponentielle. Nous appliquons un test de p-valeur avec une probabilité de fausse alarme donnée (en pratique à 5%) pour fixer le seuil de décision sur $\mathcal{C}_1$.
La seconde méthode de référence exploite le flot résiduel comme dans le paragraphe 4.3. Nous évaluons alors le critère $\mathcal{C}_2$ suivant :

$$\mathcal{C}_2 = \frac{1}{|\Omega|} \sum_{p \in \Omega} \|\omega(p,t) - \omega_{\hat{\theta}}(p,t)\|_2 \tag{5}$$

où ω est le flot optique et $\omega_{\hat{\theta}}$ est le flot optique correspondant au mouvement dominant estimé. Motion2D est utilisé à nouveau pour estimer le mouvement dominant et FlowNet2.0 pour le calcul du flot optique.
De la même façon qu'avec $\mathcal{C}_1$, la distribution empirique des valeurs de $\mathcal{C}_2$ est identifiée à une distribution exponentielle. La même stratégie de seuillage fondée sur une probabilité de fausse alarme à 5% est appliquée.
La suite des expériences montrera que le choix d'un pas de temps plus grand que 1 permet d'améliorer les performances des méthodes. Afin que les méthodes de référence puissent tirer également parti de cette constatation, le pas

de temps est fixé à 4 pour ces deux méthodes.

Les performances de ces deux méthodes sur l'ensemble de test réel sont rassemblées dans le tableau 1. Les performances constatées ne dépassent pas les 55 % de prédictions correctes sur l'ensemble de test complet. Ces résultats confirment la non-trivialité du problème traité et la nécessité de développer des méthodes plus élaborées.

5.4 Évaluation des méthodes définies

Nous avons fait le choix d'architectures de classification fournissant la seule probabilité de saillance à laquelle nous appliquons un seuillage à 0,5. Nous avons en effet constaté par des expériences préliminaires, sachant que nous traitons un problème à deux classes, que la somme des probabilités de saillance et de non-saillance tend en pratique vers 1. L'alternative habituelle en classification consiste à calculer les probabilités pour toutes les classes et à choisir la classe de probabilité la plus élevée. Mais en l'occurrence, les deux stratégies sont pratiquement équivalentes. Les fonctions de répartition des probabilités de la figure 6 illustrent bien le fait que les probabilités calculées se démarquent bien du seuil à 0.5. La valeur du seuil n'est donc pas critique, et la politique de seuillage permet au passage de gagner un petit peu de temps de calcul.

Pour déterminer le pas de temps optimal pour chaque méthode, une évaluation a été d'abord réalisée sur l'ensemble de validation, comme illustré au tableau 2.

Les performances des différentes méthodes sont rassemblées dans le tableau 1. La méthode la plus efficace est RFS-Motion2D, basée sur le flot résiduel. Elle atteint une précision de 87,5% et démontre l'intérêt d'une approche portant sur la seule information du mouvement. La méthode WS-Motion2D s'appuyant sur les images recalées atteint un score qui reste élevé, de 85,2%. Nous constatons que les variantes des méthodes utilisant l'apprentissage profond pour l'estimation du mouvement dominant restent un peu moins performantes sur cette base de vidéos réelles. Des exemples de classification sont rassemblés et commentés en figure 7. Notons que, malgré le fait que le mouvement de la caméra dans ces scènes réelles est difficilement descriptible par un modèle paramétrique unique, les classifications proposées sont malgré tout pertinentes.

En comparant les résultats à ceux obtenus en n'utilisant que les données synthétiques pour l'entraînement (voir tableau 3), nous observons que l'apprentissage complémentaire sur vidéos réelles permet dans tous les cas d'améliorer la précision. Notons toutefois que la méthode WS-Motion2D a déjà de bonnes performances avec la seule utilisation des données synthétiques pour l'apprentissage, ce qui est moins vrai pour les autres méthodes. Une explication possible de ce comportement est que dans le cas de WS-Motion2D, les images sont réalignées avec une précision correcte, ce qui facilite la décision.

Pour les méthodes utilisant Motion2D, le choix du modèle paramétrique peut avoir un impact dans les performances des méthodes. Le tableau 4 compare les résultats obtenus pour WS-Motion2D en remplaçant le modèle affine par un modèle de mouvement quadratique (polynômes de degré 2 pour les deux composantes du vecteur de vitesse pour un total de 8 paramètres différents) dans le test, sans réentraîner le classifieur. Nous constatons que l'influence du choix du modèle est faible, du moins sur ces vidéos réelles.

FIGURE 7 – Exemples de classification par RFS-Motion2D. Les éléments de gauche sont bien classés et ceux de droite sont mal classés. Les éléments de la première ligne sont non saillants, les autres sont tous saillants. Des cas complexes comme le cours d'eau ou l'escargot sont bien classés. Le vent dans les feuillages (en haut à droite) et de petits objets mobiles partiellement masqués (entourés en rouge) posent par contre problème.

6 Conclusion

Dans cet article, nous avons formulé le problème de détection temporelle de la saillance dynamique dans des vidéos sous un angle nouveau. Nous avons présenté quatre méthodes pour résoudre ce problème, qui exploitent l'apprentissage profond. Une base de vidéos synthétiques a été constituée, en complément d'une base de vidéos réelles appropriée et annotée pour le problème traité. La meilleure méthode atteint une précision de 87,5% sur l'ensemble de test réel.

Des pistes de recherches pour de futurs travaux incluent une estimation plus complète du mouvement dominant dans le cas de scènes complexes, à l'aide d'une approche multi-modèle. Le CNN utilisé pour l'estimation du mouvement dominant pourrait également être amélioré, notamment par l'utilisation de stratégies multi-échelles.

Remerciements

Ces travaux ont été partiellement financés par la DGA et la Région Bretagne par des co-financements de la thèse de Léo Maczyta.

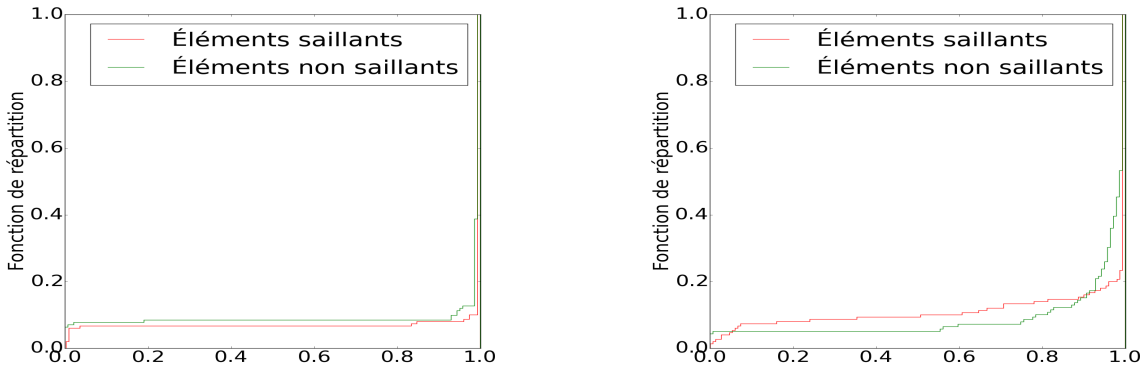

FIGURE 6 – Fonctions de répartition empiriques (probabilités cumulées) de la probabilité de saillance pour des images saillantes (en rouge) et de la probabilité de non-saillance pour des images non-saillantes (en vert), fournies par la méthode WS-Motion2D. Les deux graphiques correspondent à gauche au cas du réseau pré-entraîné sur les données synthétiques, et à droite, entraîné en outre sur les vidéos réelles. Dans les deux cas, l'ensemble de validation réel a été utilisé pour calculer les fonctions de répartition. Le pas de temps optimal a été utilisé pour chaque réseau.

Méthode	Pas de temps	Pourcentage d'images correctement classifiées	Pourcentage sur les images saillantes	Pourcentage sur les images non saillantes
$\mathcal{C}_1$	4	50,2	87,0	8,2
$\mathcal{C}_2$	4	54,4	64,6	42,8
WS-Motion2D	6	85,2	78,4	93,0
WS-FD	2	76,5	59,4	96,2
RFS-Motion2D	5	**87,5**	79,7	96,4
RFS-FD	3	84,6	73,0	98,0

TABLE 1 – Taux de bonne classification des différentes méthodes sur l'ensemble de test réel.

Pas de temps	1	2	3	4	5	6	7	8	9
Précision	84,9	89,0	88,3	88,9	**89,6**	88,9	88,9	89,2	88,5

TABLE 2 – Performance de RFS-Motion2D sur l'ensemble de validation réel pour différents pas de temps.

Méthode	Pas de temps	Pourcentage d'images correctement classifiées	Pourcentage sur les images saillantes	Pourcentage sur les images non saillantes
WS-Motion2D	2	**80,9**	76,8	85,7
WS-FD	3	67,3	62,3	73,1
RFS-Motion2D	5	76,0	62,2	91,8
RFS-FD	5	69,7	44,1	99,0

TABLE 3 – Taux de bonne classification des différentes méthodes sur l'ensemble de test réel, pour les méthodes entraînées uniquement sur données synthétiques.

Méthode	Pas de temps	Pourcentage d'images correctement classifiées	Pourcentage sur les images saillantes	Pourcentage sur les images non saillantes
WS-Motion2D (affine)	6	85,2	78,4	93,0
WS-Motion2D (quadratique)	6	85,2	77,9	93,5

TABLE 4 – Comparaison des modèle paramétriques affine et quadratique pour WS-Motion2D

Références

[1] Bideau, P. and Learned-Miller, E. A detailed rubric for motion segmentation. *arXiv :1610.10033v1*, Oct. 2016.

[2] Bideau, P. and Learned-Miller, E. It's moving ! A probabilistic model for causal motion segmentation in moving camera videos. In *ECCV* 2016.

[3] Chaabouni, S., Benois-Pineau, J. and Hadar, O. Prediction of visual saliency in video with deep CNNs. In *SPIE Applications of Digital Image Processing* 2016.

[4] Crivelli, T., Cernuschi-Frias, B., Bouthemy, P. and Yao, J.-F. Motion textures : modeling, classification and segmentation using mixed-state Markov random fields. *SIAM J. on Imaging Sciences*, 6(4) :2484-2520, 2013.

[5] Everingham, M., Van Gool, L., Williams, C. K. I., Winn, J. and Zisserman, A. The PASCAL visual object classes challenge 2012 (VOC2012) results. *http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html*

[6] Fang, Y., Wang, Z., Lin, W. and Fang, Z. Video saliency incorporating spatiotemporal cues and uncertainty weighting. *IEEE Trans. on Image Processing*, 23(9) :3910-3921, Sept. 2014.

[7] Feng, Y., Yuan, Y. and Lu, X. Learning deep event models for crowd anomaly detection. In *Neurocomputing* 2017.

[8] Fischer, P., Dosovitskiy, A, Ilg, E., Häusser, P., Hazırbaş, C., Golkov, V., Van der Smagt, P., Cremers, D., and Brox, T. FlowNet : Learning optical flow with convolutional networks. In *ICCV* 2015.

[9] He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *CVPR* 2016.

[10] Huang, C-R., Chang, Y-J., Yang, Z.-X., and Lin, Y.-Y. Video saliency map detection by dominant camera motion retrieval. *IEEE Trans. on Circuits and Systems for Video Technology*, 24(8) :1336-1349, Aug. 2014.

[11] Ilg, E., Mayer, N., Saikia, T., Keuper, M., Dosovitskiy, A., and Brox, T. FlowNet 2.0 : Evolution of optical flow estimation with deep networks. In *CVPR* 2017.

[12] Jaderberg M., Simonyan, K., Zisserman, A., and Kavukcuoglu, K. Spatial transformer networks. In *NIPS* 2015.

[13] Jia, Y. and Shelhamer, E. and Donahue, J. and Karayev, S. and Long, J. and Girshick, R. and Guadarrama, S. and Darrell, T. (2014) Caffe : Convolutional architecture for fast feature embedding. In *ACM Conf. on Multimedia*, 2014.

[14] Karimi, A H., Shafiee, M.J., Scharfenberger, C., BenDaya, I., Haider, S., Talukdar, N., Clausi, D.A., and Wong, A. Spatio-temporal saliency detection using abstracted fully-connected graphical models. In *ICIP* 2016.

[15] Kim, W. and Kim, C. Spatiotemporal saliency detection using textural contrast and its applications. *IEEE Trans. on Circuits and Systems for Video Technology*, 24(4) :646–659, May 2014.

[16] Kingma, D. and Ba, J. Adam : a method for stochastic optimization. In *ICLR* 2014.

[17] Krizhevsky, A. and Sutskever, I. and Hinton, G. E. ImageNet classification with deep convolutional neural networks. In *NIPS* 2012.

[18] Li, Z., Gavrilyuk, K., Gavves, E., Jain, M., Snoek, C.G.M. VideoLSTM convolves, attends and flows for action recognition. In *Computer Vision and Image Understanding*, 166 :41-50, 2018.

[19] Liu, Z., Zhang, X., Luo, S. and Le Meur, O. Superpixel-based spatiotemporal saliency detection. *IEEE Trans. on Circuits and Systems for Video Technology*, 24(9) :1522-1540, Sept. 2014.

[20] Mahapatra, D., Gilani, S.O. and Saini, M.K. Coherency based spatio-temporal saliency detection for video object segmentation. *IEEE J. of Selected Topics in Signal Processing*, 8(3) : 454–462, March 2014.

[21] Narayana, M., Hanson, A., and Learned-Miller, E. Coherent motion segmentation in moving camera videos using optical flow orientations. In *ICCV* 2013.

[22] Ochs P., Malik J., and Brox, T. Segmentation of moving objects by long term video analysis. *IEEE Trans. on Pattern Analysis and Machine Intel.*, 36(6) :1187–1200, 2014.

[23] Odobez, J.-M. and Bouthemy, P. Robust multiresolution estimation of parametric motion models. *J. of Visual Comm. and Image Repr.*, 6(4) :348-365, Dec. 1995.

[24] Pérez-Rúa, J.M., Basset, A., Bouthemy, P. Detection and localization of anomalous motion in video sequences from local histograms of labeled affine flows In *Frontiers in ICT, Computer Image Analysis*, May 2017.

[25] Rocco, I., Arandjelović, and R., Sivic, J. Convolutional neural network architecture for geometric matching. In *CVPR* 2017.

[26] Sharma, S., Kiros, R., and Salakhutdinov, R. Action recognition using visual attention. In *ICLR* 2016.

[27] Simonyan, K., Zisserman, A. Two-stream convolutional networks for action recognition in videos. In *NIPS*, 2014

[28] Sun, C., Shrivastava, A., Singh, S., and Gupta, A. Revisiting unreasonable effectiveness of data in deep learning era. In *ICCV* 2017.

[29] Szegedy, C., Liu, W., and Jia, Y. Going deeper with convolutions. In *CVPR* 2015.

[30] Tokmakov, P., Alahari, K., and Schmid, C. Learning motion patterns in videos. In *CVPR* 2017.

[31] Wang, W., Shen, J., and Shao, L. Consistent video saliency using local gradient flow optimization and global refinement. In *IEEE Trans. on Image Processing*, 24(11) :4185-4196, Nov. 2015.

[32] Wang, W., Shen, J., and Shao, L. Video salient object detection via fully convolutional networks. In *IEEE Trans on Image Processing*, 27(1) :38-49, Jan. 2018.

[33] Xu, D., Yan, Y. Ricci, E. and Sebe, N. Detecting anomalous events in videos by learning deep representations of appearance and motion. In *Computer Vision and Image Understanding*, 156 :117-127, 2017.

[34] Yao, L., Torabi, A., Cho, K., Ballas, N., Pal, C., Larochelle, H., and Courville, A. Describing videos by exploiting temporal structure. In *ICCV* 2015.

[35] Zhou, S., Shen, W. Zeng, D. Fang, M. Wei, Y. and Zhang, Z. Spatial–temporal convolutional neural networks for anomaly detection and localization in crowded scenes. In *Signal Proc. : Image Comm.*, 47 :358-368, 2016.