# Apprentissage profond non supervisé fondé sur l'algorithme EM pour la segmentation du mouvement

Etienne Meunier, Anaïs Badoual, Patrick Bouthemy

# Apprentissage profond non supervisé fondé sur l'algorithme EM pour la segmentation du mouvement

Etienne Meunier, Anaïs Badoual, et Patrick Bouthemy

Inria, Rennes, France

7 mars 2022

## Résumé

*Cet article présente une méthode entièrement non supervisée basée sur un réseau neuronal convolutionnel pour la segmentation du mouvement à partir du flot optique. Nous supposons que le flot optique d'entrée peut être représenté par un ensemble de modèles de mouvement paramétriques, typiquement des modèles de mouvement affines ou quadratiques. L'idée centrale de ce travail est d'exploiter le cadre de l'Espérance-Maximisation (EM). Il nous permet de concevoir d'une manière bien fondée la fonction de perte et la procédure d'apprentissage de notre réseau de neurones de segmentation du mouvement. Contrairement à la méthode EM itérative classique, une fois le réseau entraîné, nous pouvons fournir une segmentation pour tout nouveau champ de flot optique en une seule étape d'inférence et sans avoir avoir à estimer de modèles de mouvement. Notre méthode surpasse les méthodes non supervisées comparables et est très efficace en temps de calcul.*

## Mots Clés

Segmentation du mouvement, flot optique, réseau de neurones, algorithme EM.

## Abstract

*This paper presents a CNN-based fully unsupervised method for motion segmentation from optical flow. We assume that the input optical flow can be represented as a piecewise set of parametric motion models, typically, affine or quadratic motion models. The core idea of this work is to leverage the Expectation-Maximization (EM) framework. It enables us to design in a well-founded manner the loss function and the training procedure of our motion segmentation neural network. However, in contrast to the classical iterative EM, once the network is trained, we can provide a segmentation for any unseen optical flow field in a single inference step, with no dependence on the initialization of the motion model parameters since they are not estimated in the inference stage. Our method outperforms comparable unsupervised methods and is very efficient.*

## Keywords

Motion segmentation, optical flow, CNN, EM algorithm.

# 1 Introduction

La segmentation du mouvement est un problème important en vision par ordinateur, dont le but est de diviser une image en segments cohérents au sens du mouvement. Selon la formulation du problème, les segments peuvent être des couches, c'est-à-dire des sous-ensembles de points non nécessairement connectés, ou des régions, c'est-à-dire des segments connectés, formant une partition de la grille de l'image.

Pour aborder la segmentation du mouvement dans une séquence d'images de manière générale sans anticiper une application donnée, nous nous concentrons sur la segmentation du flot optique. En effet, le flot optique porte toute l'information relative au mouvement entre deux images successives d'une vidéo. Nous représentons le flot optique par un ensemble de modèles paramétriques de mouvement, typiquement affines ou quadratiques, chacun d'entre eux caractérisant le mouvement dans un segment. Ainsi, la recherche de supports (segments) et l'estimation des modèles de mouvement sont des questions étroitement liées.

Ce problème peut être abordé avec des variables latentes, ce qui impose généralement une stratégie d'optimisation alternée. L'algorithme d'Espérance-Maximisation (EM) est certainement la solution phare pour une approche statistique de ce problème [7]. Cependant, l'EM classique s'appuie sur des représentations des données d'entrée spécifiées manuellement et conduit à des algorithmes itératifs généralement très gourmands en temps de calcul. D'autre part, l'apprentissage profond, et notamment les réseaux de neurones convolutifs (CNN en anglais), sont maintenant devenus la solution-clé la plus performante pour la segmentation d'image ou de mouvement [1, 11, 27, 28].

Néanmoins, la conception d'un réseau nécessite toujours des choix, et l'étape d'apprentissage reste un problème important. L'apprentissage supervisé offre une grande précision, mais l'annotation manuelle de cartes de segmentation de mouvement pour construire une vérité terrain est très lourde. L'apprentissage non supervisé est donc préférable mais plus fragile. Cependant, une méthode non supervisée a certainement un pouvoir plus grand de généralisation et représente le meilleur moyen de traiter des contenus de vidéos absents de la phase d'apprentissage.

Dans cet article, nous réunissons les deux approches, EM et CNN, afin de concevoir une méthode de segmentation du mouvement non supervisée, efficace et mathématiquement bien fondée. Une fois entraîné, notre réseau est capable de segmenter chaque image de la vidéo sans aucune itération et sans estimation de modèle de mouvement. Par non supervisé, nous signifions que nous n'avons pas recours à la vérité terrain et à l'annotation manuelle, tant pour l'étape d'apprentissage (dans la fonction de perte et le critère d'arrêt) que pour la sélection d'hyper-paramètres du réseau. L'objectif principal de notre réseau est de découvrir la cohérence du mouvement. Pris seul, il peut segmenter le mouvement dans les vidéos. Il peut également être incorporé dans un cadre plus large d'analyse ou d'interprétation de vidéos qui nécessiterait un critère ou un indicateur de cohérence du mouvement.

Cet article est organisé comme suit. La section 2 fournit un bref état de l'art sur la segmentation du mouvement. Dans la section 3, nous formulons le problème de la segmentation du mouvement à travers le cadre EM. Dans la section 4, nous présentons comment nous utilisons l'algorithme EM pour concevoir la fonction de perte, l'architecture et la procédure d'entraînement du réseau de neurones. La section 5 présente des expériences approfondies incluant des comparaisons avec d'autres méthodes existantes sur quatre benchmarks. Enfin, la section 6 forme la conclusion.

## 2 État de l'art

Nous nous concentrons sur la période récente qui a notamment vu la segmentation du mouvement appréhendée via la segmentation des objets mobiles indépendants ou par ce qui est appelé la VOS (Video Object Segmentation). Rappelons néanmoins que par le passé des méthodes de segmentation du mouvement apparent avaient été développées à partir de deux images successives et la prise en compte de modèles de mouvement paramétriques par des méthodes markoviennes ou de clustering notamment [4, 31].

La segmentation d'objets dans des vidéos (VOS) se focalise sur la segmentation d'objets principaux (généralement un seul) se déplaçant au premier plan d'une scène et généralement suivis par la caméra. La VOS fournit une segmentation binaire, objet premier contre arrière-plan [32]. Néanmoins, il peut arriver que l'arrière-plan contienne également des objets en mouvement, comme dans certaines vidéos du jeu de données DAVIS2016 [26]. La disponibilité de grands ensembles de données VOS annotées rend possible l'utilisation de techniques d'apprentissage profond supervisé pour la VOS : [5], [6], [29], [12], [36] et [19].

Des méthodes de VOS non supervisées ont également été développées. Celle décrite dans [25] exploite les frontières de mouvement dans l'image et des modèles d'apparence pour reconnaître les objets en mouvement dans les vidéos. Dans [9], les auteurs supposent que l'objet en mouvement possède des caractéristiques d'apparence et de mouvement de bas niveau distinctives (comme l'orientation et l'amplitude des vecteurs de flot), par rapport à l'arrière-plan. Ils utilisent une mesure inspirée de la fonction de Tukey pour détecter les pixels aberrants dans les images et les étiqueter comme appartenant à des objets en mouvement. Dans [14], les auteurs exploitent la propriété de récurrence de l'objet mobile principal pour le segmenter à partir de la séquence d'images. Dans l'article [34], les auteurs mettent en place un cadre d'apprentissage adverse entre un réseau "générateur" produisant un masquage sur le flot optique, et un réseau "inpainter" essayant de reconstruire le flot à l'intérieur du masque. Le raisonnement est qu'un mouvement indépendant ne peut pas être prédit par le mouvement environnant. Cependant, cette méthode peut également être sensible aux objets statiques au premier plan générant un mouvement de parallaxe, comme le montre l'article [22]. Les auteurs de [35] font en sorte que deux modèles, dynamique et statique, se renforcent mutuellement pendant la phase d'apprentissage non supervisée, de sorte que le réseau puisse détecter précisément les objets d'intérêt dans les images traitées.

Dans un cadre différent, la méthode décrite dans [20] utilise le flot optique pour confirmer la validité d'une segmentation spatiale en vérifiant si le mouvement des pixels est cohérent dans les régions segmentées par le réseau. Comme pour notre méthode, ils estiment des modèles de mouvement paramétriques. Cependant, ils n'exploitent pas une perte de cohérence issue du cadre EM comme le fait notre méthode. En outre, leur schéma est limité à l'utilisation de la méthode des moindres carrés ordinaires pour conserver la differentiabilité de leur fonction de perte, ce qui ne permet pas d'introduire des fonctions robustes comme nous pouvons le faire. Dans [33], le réseau conçu effectue une segmentation du flot optique et comprend plusieurs composantes, le codage des caractéristiques, le regroupement itératif, le décodage en couches et la reconstruction du flot. La fonction de perte implique un terme d'entropie pour rendre les masques aussi binaires que possible, et un terme de cohérence temporelle. En outre, elle exploite le mécanisme d'attention introduit dans [18].

## 3 La segmentation de mouvement par l'algorithme EM

L'idée centrale de notre travail est d'exploiter le cadre de l'Espérance-Maximisation pour concevoir de manière mathématiquement fondée la fonction de perte et la procédure d'apprentissage de notre réseau de neurones de segmentation du mouvement. Dans cette section, nous décrivons d'abord une façon d'utiliser l'algorithme EM pour la segmentation du flot optique.

Le flot optique $f \in \mathbb{R}^{2\times W\times H}$ est un champ de vecteurs défini sur une grille d'image $\Omega$ de taille $W \times H$. Nous désignons par $f_i \in \mathbb{R}^2$ le vecteur de mouvement associé à chaque site $i \in \Omega$ de cette grille. Nous faisons l'hypothèse que tout champ de mouvement ou flot optique peut être décomposé en un ensemble de $K$ segments ou couches, chacun regroupant une partie (éventuellement non connectée) de la grille d'image et présentant un mouvement co-

hérent. Afin d'exprimer cette cohérence, nous choisissons de représenter le champ de mouvement au sein de chaque segment $k$ par un modèle paramétrique défini par des paramètres $\theta_k$. Nous notons $\theta = \{\theta_k, k = 1, \ldots, K\}$. En pratique, nous utilisons des modèles de mouvement polynomiaux, affine (polynômes du premier degré en les coordonnées des points dans l'image) ou quadratique (polynômes du second degré).

À partir de cette hypothèse, nous pouvons écrire la vraisemblance du flot optique $f$ étant donné l'ensemble des paramètres $\theta$ comme $p(f|\theta)$. Afin de rendre explicite la partition de $f$ en $k$ segments et les $\theta_k$ individuels associés, nous introduisons des variables latentes $z_i$ telles que $p(z_i = k|f_i, \theta_k)$ représente la probabilité que le site $i$ appartienne à la couche $k$. En supposant l'indépendance conditionnelle, le logarithme de la vraisemblance peut être écrit comme ci-dessous dans l'étape 1 de l'éq.(1). Ensuite, nous introduisons les variables $z_i$ (étapes 2 et 3 de l'éq.(1)), et nous faisons enfin apparaître directement toute distribution positive $q$ (étape 4 de l'éq.(1)). Nous avons :

$$\begin{aligned}
\log(p(f|\theta)) &= \log \prod_i p(f_i|\theta) \\
&= \log \prod_i \sum_k p(f_i, z_i^k|\theta_k) \\
&= \sum_i \log \sum_k p(f_i, z_i^k|\theta_k) \\
&= \sum_i \log \sum_k q(z_i^k) \frac{p(f_i, z_i^k|\theta_k)}{q(z_i^k)}, \qquad (1)
\end{aligned}$$

où $z_i^k \triangleq [z_i = k]$. Maximiser $\log(p(f|\theta))$ par rapport à $\theta$ est évidemment compliqué, même si cela se résume à $k$ maximisations par rapport aux $\theta_k$. En effet, les variables $z_i$ sont cachées. Pour maximiser l'éq.(1) par rapport à $\theta_k$, les variables $z_i$ doivent être disponibles. Par conséquent, nous devons également maximiser par rapport aux $z_i$.

Nous pouvons utiliser l'inégalité de Jensen ($h(\mathbb{E}[x]) \geq \mathbb{E}[h(x)]$ pour toute fonction concave $h$), comme cela est fait en EM classique [23], afin de construire une borne inférieure $ll(\theta)$ de la log-vraisemblance $\log(p(f|\theta))$ définie comme suit :

$$\begin{aligned}
ll(\theta) &= \sum_i \sum_k q(z_i^k) \log \frac{p(f_i, z_i^k|\theta_k)}{q(z_i^k)} \\
&= \sum_i \sum_k q(z_i^k) \log p(f_i, z_i^k|\theta_k) \\
&\quad - \sum_i \sum_k q(z_i^k) \log q(z_i^k), \qquad (2)
\end{aligned}$$

où le premier terme de l'éq.(2) est l'espérance sur $q(z_i)$ de $\log p(f_i, z_i|\theta)$ et le second est l'entropie que nous notons $\mathcal{H}$. L'expression résultante de la borne inférieure est :

$$ll(\theta) = \sum_i \mathbb{E}_{q(z_i)}[\log p(f_i, z_i|\theta)] + \sum_i \mathcal{H}(q(z_i)). \qquad (3)$$

Dans l'algorithme EM classique, on prend généralement $q(z_i^k) \triangleq p(z_i^k|f_i, \theta_k)$. Ensuite, on alterne entre une étape d'espérance où l'expressoin $q(z_i^k) \forall k$ est estimée, et une étape de maximisation où $ll(\theta)$ est maximisée par rapport aux $\theta_k$. L'alternance de ces deux étapes augmente de manière monotone la log-vraisemblance jusqu'à ce qu'elle atteigne un optimum local [23].

# 4 Segmentation de mouvement par un réseau convolutionnel

Dans notre cas, nous adoptons un modèle de réseau de neurones $g_\phi(f)$, paramétré par $\phi$ et prenant en entrée le flot optique $f$, pour produire la segmentation du mouvement qui est notre objectif principal. La motivation du choix d'un tel modèle est que nous avons ainsi une grande latitude de spécification de la fonction $q(z_i^k)$. Plus important encore, après l'étape d'apprentissage, notre réseau est capable de prédire la segmentation du mouvement sans itération et sans faire intervenir de modèles de mouvement, contrairement à l'algorithme EM classique. En outre, il est sans comparaison beaucoup plus rapide. Au stade de l'apprentissage, nous traitons deux ensembles de paramètres : les paramètres des modèles de mouvement $\theta$ et les paramètres du modèle de réseau $\phi$. Au stade de l'inférence, seuls les paramètres du réseau $\phi$ interviendront. Le diagramme général de notre méthode, avec les étapes d'apprentissage et d'inférence, est donné à la Fig.1.

## 4.1 Spécification du réseau

En revenant à l'équation (2) et en suivant le choix exprimé ci-dessus, nous prenons $g_\phi(f)_i^k$ comme $q(z_i^k)$, où $g_\phi(f)_i^k$ est la probabilité (prédiction) donnée par le réseau pour que le site $i$ appartienne au segment $k$ étant donné le flot optique d'entrée $f$. La borne inférieure $ll$ dépend maintenant de deux ensembles de paramètres, $\theta$ et $\phi$, et s'écrit :

$$\begin{aligned}
ll(\theta, \phi) &= \sum_i \sum_k g_\phi(f)_i^k \log p(f_i, z_i^k|\theta_k) \\
&\quad - \sum_i \sum_k g_\phi(f)_i^k \log g_\phi(f)_i^k \\
&= \sum_i \mathbb{E}_{g_\phi(f)_i}[\log p(f_i, z_i|\theta)] + \sum_i \mathcal{H}(g_\phi(f)_i), \qquad (4)
\end{aligned}$$

que nous optimisons alternativement par rapport à $\theta$ et $\phi$ pour l'étape d'entraînement comme suit :

$$\theta^* = \arg\max_\theta \sum_i \sum_k g_\phi(f)_i^k \log(p(f_i, z_i^k|\theta_k)) \qquad (5)$$

$$\phi^* = \arg\max_\phi \sum_i \sum_k g_\phi(f)_i^k \log(p(f_i, z_i^k|\theta_k^*)) + \sum_i \mathcal{H}(g_\phi(f)_i). \qquad (6)$$

Comme décrit dans [10], l'entropie de la segmentation prédite $\mathcal{H}(g_\phi(f)_i)$ apparaît naturellement en éq.(6). L'entropie mesure l'incertitude statistique et est maximisée pour

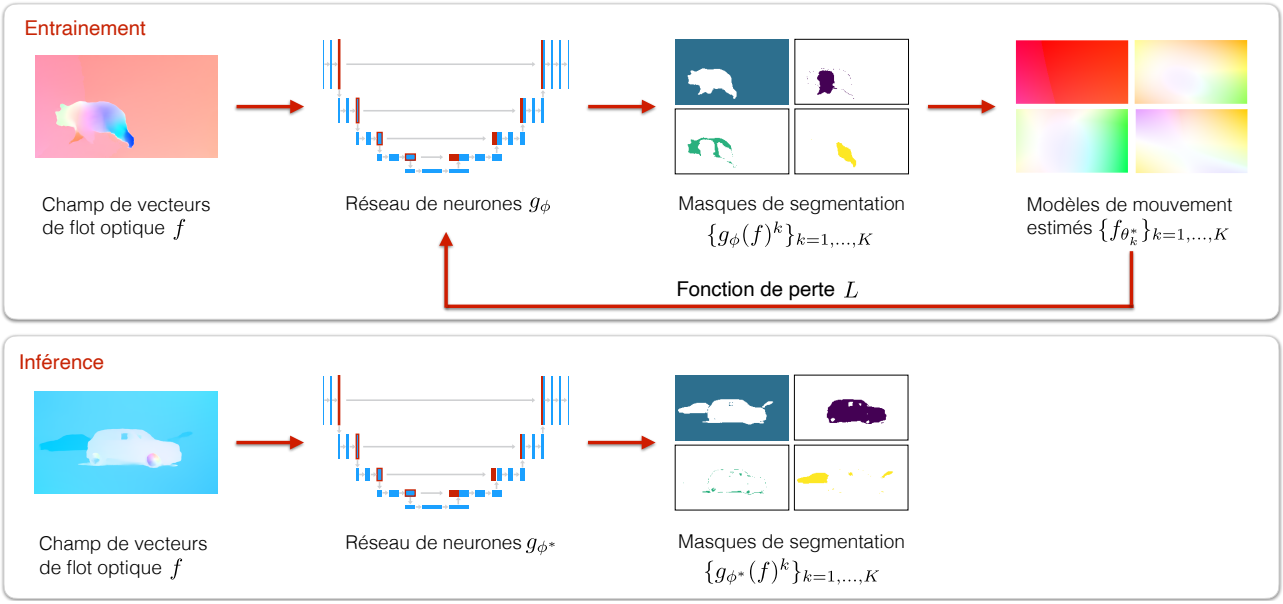


FIGURE 1 – Diagramme de la méthode d'apprentissage profond proposée pour les étapes d'apprentissage (en haut) et d'inférence (en bas). *Phase d'apprentissage* Tout d'abord, nous segmentons le champ de vecteurs de flot optique $f$ avec le réseau de neurones $g_\phi$. Ensuite, nous obtenons les modèles de mouvement paramétriques optimaux $\{f_{\theta_k^*}\}_{k=1,\dots,K}$ dans chaque masque de segmentation probabilisé $\{g_\phi(f)^k\}_{k=1,\dots,K}$ en utilisant (13). Enfin, nous mettons à jour les paramètres $\phi$ du réseau de neurones en utilisant (14), où la fonction de perte est définie dans (12). Cette étape d'apprentissage est exécutée de manière itérative sur chaque batch $\mathcal{B}$ (de taille 1 dans cette illustration pour des besoins de lisibilité). *Phase d'inférence* Nous appliquons directement le réseau entraîné $g_{\phi^*}$ à tout nouveau champ de flot optique $f$ pour obtenir les masques de segmentation probabilisés $\{g_{\phi^*}(f)^k\}_{k=1,\dots,K}$. Il n'y a pas d'estimation des modèles de mouvement $\{f_{\theta_k^*}\}_{k=1,\dots,K}$ dans l'étape d'inférence, contrairement à l'étape d'apprentissage. Pour des raisons de visualisation, les flots optiques et les modèles de mouvement polynomiaux sont représentés avec le code couleur HSV [2], mais en réalité, le flot $f$ utilisé comme entrée du réseau neuronal est considéré comme un champ de vecteurs 2D. Nous avons ainsi une entrée à deux canaux.

$g_\phi(f)_i^k = \frac{1}{K}, \forall i,k$. Elle agit comme un terme de régularisation équilibrant le terme de vraisemblance pour éviter de tomber trop rapidement dans des optima locaux.
En ce qui concerne l'optimisation sur $\theta$, nous pouvons atteindre un optimum local en utilisant un algorithme itératif standard. Cependant, nous ne pouvons effectuer qu'une étape de descente de gradient pour l'optimisation par rapport aux poids du réseau $\phi$.
Afin de comprendre comment le réseau apprend à produire la segmentation du mouvement, en suivant [23], nous pouvons réécrire la limite inférieure $ll$ ainsi :

$$\begin{aligned} ll(\theta,\phi) &= \sum_i \sum_k g_\phi(f)_i^k \log \frac{p(f_i, z_i^k|\theta_k)}{g_\phi(f)_i^k} \\ &= \sum_i \sum_k g_\phi(f)_i^k \log \frac{p(z_i^k|f_i,\theta_k)}{g_\phi(f)_i^k} \\ &+ \sum_i \log p(f_i|\theta) \sum_k g_\phi(f)_i^k \\ &= -\sum_i \mathbb{KL}[g_\phi(f)_i || p(z_i|f_i,\theta)] + \log(p(f|\theta)). \end{aligned} \tag{7}$$

Par conséquent, l'étape d'optimisation sur les poids du réseau est définie par :

$$\phi^* = \arg\min_\phi \sum_i \mathbb{KL}[g_\phi(f)_i || p(z_i|f_i,\theta^*)], \tag{8}$$

et cela revient à minimiser la KL-divergence entre la segmentation produite par le réseau et la segmentation liée aux paramètres optimaux $\theta^*$. Ainsi, le réseau est entraîné à produire une segmentation pour un ensemble donné de paramètres de mouvement. Au fur et à mesure que la qualité de la segmentation du réseau s'améliore, la qualité de l'estimation de $\theta^*$ s'améliore également, poussant en retour les poids du réseau à produire une meilleure segmentation.

## 4.2 Vraisemblance et fonction de perte

Dans la section précédente, nous avons décrit le principe du processus d'entraînement du réseau. Dans cette section, nous abordons la définition des différents termes de la fonction de perte.
Tout d'abord, nous décomposons la probabilité jointe dans l'eq.(4) en une vraisemblance conditionnelle et un *a priori* :

$$p(f_i, z_i^k|\theta_k) = p(f_i|z_i^k,\theta^k)p(z_i^k). \tag{9}$$

La vraisemblance conditionnelle $p(f_i, z_i^k|\theta_k)$ évalue comment le modèle de mouvement paramétrique estimé dans

une région donnée s'adapte au flot observé dans cette région. Dans ce travail, nous utilisons un *a priori* uniforme pour $p(z_i^k)$. Néanmoins, nous pourrions adopter un *a priori* plus complexe, si nous voulions par exemple influencer la taille des régions.

Un point important est le choix de la forme de la vraisemblance $p(f_i|z_i^k, \theta_k)$ qui est utilisée pour comparer le flot optique d'entrée avec le flot paramétrique pour un ensemble donné de paramètres $\theta$. En pratique, puisque nos modèles de mouvement paramétriques dépendent de la position des points sur l'espace de l'image 2D, nous introduisons une fonction déterministe $c(i)$ qui fait correspondre le site $i$ à une expansion polynomiale impliquant ses coordonnées. Plus précisément, pour un modèle de mouvement affine à 6 paramètres, nous définissons $c(i) = \left[1, x_i, y_i\right]$ ; pour un modèle quadratique complet à 12 paramètres, nous définissons $c(i) = \left[1, x_i, y_i, x_i^2, x_i y_i, y_i^2\right]$. La vraisemblance évalue la distance entre les vecteurs de flot d'entrée $f$ et les vecteurs de flot paramétrique $f_{\theta_{ki}} \triangleq \theta_k^T \cdot c(i), \forall k, i$. Sa forme générale est donnée par :

$$p(f_i|z_i^k, \theta_k) = \frac{1}{Z} \exp(-\delta(f_i, \theta_k^T \cdot c(i))), \tag{10}$$

où $\delta : \mathbb{R}^{2*2} \rightarrow \mathbb{R}$ est une fonction de distance à définir. Si $\delta$ vérifie $\delta(a + b, a) = \delta(b, 0)$, ce qui est vérifié pour toutes les fonctions de distance prises en compte, alors $Z$ ne dépend que de la fonction $\delta$ et non de son entrée. Cela nous permet d'effectuer une optimisation sans calculer explicitement $Z$.

Le choix de la fonction de distance $\delta$ est central, car elle est utilisée à la fois pour l'estimation des modèles de mouvement paramétriques et pour l'entraînement du réseau (voir eq.(5) et eq.(6)). Les fonctions de perte robustes peuvent être bénéfiques, comme le montre l'étude approfondie de [3]. Nous considérons les fonctions de distance suivantes :

- $L_2$ au carré : $\delta(f_i, \theta_k^T \cdot c(i)) = ||f_i - \theta_k^T \cdot c(i)||_2^2$,
- norme $L_2$ : $\delta(f_i, \theta_k^T \cdot c(i)) = ||f_i - \theta_k^T \cdot c(i)||_2$,
- norme $L_1$ : $\delta(f_i, \theta_k^T \cdot c(i)) = ||f_i - \theta_k^T \cdot c(i)||_1$.

$L_1$ (en raison de la valeur absolue impliquée) et $L_2$ (en raison de la racine carrée de la somme impliquée) apportent une robustesse aux éventuelles valeurs aberrantes dans le flot optique, contrairement à la distance quadratique.

Nous définissons la fonction de perte de notre réseau par :

$$L(f, \theta, \phi) = -ll(\theta, \phi), \tag{11}$$

où $ll(\theta, \phi)$ est donnée par eq.(4). En prenant en compte eq.(10), nous formulons la fonction de perte comme :

$$L(f, \theta, \phi) = \frac{1}{\alpha} \sum_i \sum_k g_\phi(f)_i^k \delta(f_i, \theta_k^T \cdot c(i)) + \sum_i \sum_k g_\phi(f)_i^k \log g_\phi(f)_i^k + I * \log(Z * K), \tag{12}$$

où $\alpha$ est lié à l'incertitude de la mesure du flot et à l'adéquation résultante du modèle paramétrique de mouvement. Elle nous permet d'équilibrer les termes de vraisemblance, d'*a priori* et d'entropie de la foncton de perte. Nous avons fixé $\alpha = 10^{-2}$ dans toutes nos expériences, mais en pratique, le modèle de réseau est assez robuste au choix de cet hyperparamètre.

### 4.3 Entraînement du réseau et augmentation des données

Pour chaque flot d'entrée $f$ de l'ensemble de données d'apprentissage, nous minimisons $L(f, \theta, \phi)$ en fonction de chaque groupe de paramètres. Cette optimisation alternée est effectuée sur chaque batch $\mathcal{B}$ comme suit :

$$\theta^* = \arg\min_\theta \sum_{f \in \mathcal{B}} L(f, \theta, \phi^t) \tag{13}$$

$$\phi^{t+1} = \phi^t - \gamma \nabla_\phi \sum_{f \in \mathcal{B}} L(f, \theta^*, \phi), \tag{14}$$

où $t$ est le nombre d'itérations, $f$ le champ de vecteurs du flot optique d'entrée et $\gamma$ le taux d'apprentissage.

En pratique, nous utilisons un algorithme d'optimisation pour estimer $\theta^*$ et une différentiation automatique pour calculer les gradients par rapport à $\phi$. Comme décrit dans la sous-section 4.1, nous considérons $\theta^*$ comme fixe dans l'étape du gradient par rapport à $\phi$, ce qui fait que $\nabla_\phi L(f, \theta^*, \phi)$ est triviale à calculer en utilisant la différentiation automatique. Les détails pratiques sont fournis dans la sous-section 5.1.

Comme cela s'est avéré bénéfique dans de nombreux problèmes de vision par ordinateur, nous procédons à l'augmentation de données pour entraîner le réseau de segmentation du mouvement. Cependant, les données d'entrée ne sont pas des images mais des flots optiques dans notre cas, ce qui nous a conduit à définir une procédure originale d'augmentation des données. Nous ajoutons, à chaque champ de vecteurs de flot optique du jeu de données, un modèle de mouvement paramétrique dont les paramètres sont tirés au hasard. Ceci a l'avantage de multiplier les configurations de flot, tout en gardant la même structure de flot que dans l'échantillon initial, c'est-à-dire la même segmentation cible à prédire. Ainsi, nous entraînons le réseau à être invariant au champ de mouvement global, ce que nous identifions comme étant l'un des défis les plus importants pour la généralisation.

## 5 Résultats expérimentaux

### 5.1 Implémentation

Le flot optique est calculé sur chaque paire de deux images successives des vidéos à l'aide de la méthode RAFT [30]. Ensuite, nous les sous-échantillonnons pour obtenir des champs de vecteurs de taille $128 \times 224$ qui sont fournis en entrée du réseau. La segmentation résultante est ensuite ré-échantillonnée à la taille de l'image originale pour être évaluée par rapport à la vérité terrain. Cela nous permet

| Méthode | Entraînement | Données d'entrée | Davis2016 $\mathcal{J}$ | Davis2016 $\mathcal{F}$ | SegTrack V2 $\mathcal{J}$ | FBMS59 $\mathcal{J}$ | MoCA $\mathcal{J}$ |
|---|---|---|---|---|---|---|---|
| **Notre méthode** | Non Supervisé | Flot | 69.3 | 70.7 | 55.5 | 57.8 | 61.8 |
| MoSeg [33] | | Flot | 68.3 | 66.1 | 58.6 | 53.1 | 63.4 |
| $TIS_0$ [9] | | Flot | 56.2 | 45.6 | - | - | - |
| $TIS_s$ [9] | | Image & Flot | 62.6 | 59.6 | - | - | - |
| CIS - Avec Post [34] | | Image & Flot | 71.5 | | 62.0 | 63.6 | 54.1 |
| CIS - Sans Post [34] | | Image & Flot | 59.2 | | 45.6 | 36.8 | 49.4 |
| FTS [25] | | Flot | 55.8 | | 47.8 | 47.7 | - |
| DyStab - Stat&Dyn [35] | Représentation Supervisée | Image & Flot | 80.0 | | 73.2 | 74.2 | - |
| DyStab - Dyn [35] | | Flot | 62.4 | | 40.0 | 49.1 | - |
| ARP [14] | | Image & Flot | 76.2 | 70.6 | 57.2 | 59.8 | - |
| MATNet [36] | Supervisé | Flot | 82.4 | 80.7 | | | 64.2 |
| COSNet [19] | | Image | 80.5 | 79.5 | - | 75.6 | 50.7 |

TABLE 1 – Résultats sur les ensembles de données DAVIS2016 (sous-ensemble de validation), SegTrackV2, FBMS59 (sous-ensemble de test), et MoCA, pour plusieurs méthodes non supervisées et supervisées (scores tirés de [9], [34], [33] et [35]). Par représentation supervisée, nosu entendons l'utilisation de descripteurs pré-appris par une méthode supervisée. $\mathcal{J}$ est l'indice de Jaccard (similarité des régions) et $\mathcal{F}$ tient compte de la précision des contours. Plus la valeur est élevée, meilleure est la performance. Pour plus d'explication sur les critères d'évaluation, nous renvoyons le lecteur au site web de DAVIS2016. Conformément aux processus d'évaluation standards, le score présenté est la moyenne des scores calculés sur chaque flots pour tous les datasets sauf pour DAVIS2016 ou c'est la moyenne des scores obtenus sur chaque séquence.

d'effectuer des étapes d'entraînement et d'inférence beaucoup plus économes en temps de calcul. Dans toutes les expériences, nous utilisons la fonction de perte définie avec la norme $L_1$, sauf indication contraire.

Nous choisissons le modèle de mouvement quadratique complet avec 12 paramètres pour représenter le flot optique dans chaque segment $k$. Nous prenons ce modèle de mouvement paramétrique car il peut mieux s'adapter aux mouvements complexes. Il est particulièrement utile pour le mouvement d'arrière-plan lorsque le mouvement de la caméra comprend à la fois une translation et une rotation dans une scène statique impliquant des objets à différentes profondeurs, ainsi que pour les mouvements articulés.

Notre méthode est entièrement non supervisée, ce qui signifie que nous n'avons recours à aucune annotation manuelle pour l'entraînement ou la sélection du modèle. En effet, nous fondons notre critère d'arrêt de l'entraînement sur la fonction de perte évaluée sur l'ensemble de validation. Ainsi, nous insistons sur le fait que nous n'utilisons aucun masque de vérité terrain pour l'entraînement ou la sélection des paramètres du réseau. Dans toutes les expériences, nous avons pris l'ensemble officiel d'entraînement de DAVIS2016 [26] comme ensemble de validation.

Notre réseau n'est jamais entraîné sur des vidéos d'entraînement appartenant au même ensemble de données que les vidéos de test. De cette façon, nous démontrons le pouvoir de généralisation de notre méthode sur des jeux de données non vus. Afin de renforcer ce principe, nous avons entraîné notre modèle sur le même jeu de données pour toutes les expériences. De plus, il s'agit d'un jeu de vidéos synthétiques, Flying Things 3D (FT3D) [21]. Par souci d'exhaustivité, nous avons également effectué des entraînements sur des jeux de données réels et avons obtenu des performances similaires ou seulement légèrement supérieures. Cependant, cela pose le problème du choix d'un jeu de données d'entraînement réel, alors que le jeu synthétique sera utilisable pour tout jeu réel de données de test.

Rappelons que l'optimisation sur $\theta$ n'intervient pas au stade de l'inférence. Les probabilités prédites par le réseau pour chaque point de l'image (ou site) d'appartenir à chaque segment $k$ sont directement utilisées pour produire la carte de segmentation du mouvement. Nous sélectionnons simplement pour chaque site le segment $\hat{k}$ ayant la probabilité la plus élevée.

Nous prenons en entrée le flot optique dans sa représentation par champ de vecteurs à valeurs réelles. Ainsi, nous disposons d'une entrée à deux canaux pour le réseau. Notre fonction de perte et notre procédure d'apprentissage pourraient être adaptées à n'importe quel réseau de neurones conçu pour la segmentation. Nous choisissons l'architecture convolutive bien connue U-Net [28] pour $g_\phi$. Nous utilisons une implémentation légèrement modifiée de celle disponible sous PyTorch Lightning [8]. Nous prenons sept couches de sous-échantillonnage et commençons avec une profondeur de caractéristiques de 64. De même que la sélection de l'epoch d'arrêt, cette structure a été choisie sans recourir à des annotations manuelles. Dans ce cas, la fonction de perte est prise en compte sur le jeu de données FT3D. Comme dans [22], nous utilisons InstanceNorm entre les blocs convolutifs, ce qui permet au réseau de gérer les variations d'amplitude du flot optique entre instants ou entre vidéos. Les amplitudes du flot ne peuvent être facilement bornées ou normalisées contrairement aux images codées sur 8 bits.

Nous utilisons l'optimiseur Adam [13] avec un taux d'apprentissage de $10^{-4}$ pour entraîner le réseau. L'optimisa-

tion sur $\theta$ suit l'implémentation Pytorch de L-BGFS [17]. Notre réseau est efficace en temps, étant un simple réseau convolutif, avec un temps de calcul moyen de $0,008s$ par champ de vecteurs de taille $128 \times 224$ sur un Tesla-V100, si l'on considère un batch de 32. Sans aucune parallélisation (batch de 1), il peut fonctionner à 36 images par seconde, ce qui le rend utilisable pour des applications en temps réel. En particulier, notre méthode est plus rapide que la méthode la plus rapide présentée dans [33], car nous n'utilisons pas de module d'attention itératif, ce qui réduit notre complexité de calcul. De plus, contrairement aux méthodes impliquant l'auto-attention, il existe une relation simplement linéaire entre la complexité de notre réseau et la taille du champ de vecteurs de flot optique.

## 5.2 Evaluation comparative

Nous allons évaluer quantitativement les performances de notre méthode de segmentation du mouvement. En raison du manque de benchmarks dédiés à la segmentation de flot optique, nous avons recours à des jeux de données VOS.

**DAVIS2016** [1] [26] comprend 50 vidéos (3455 images) réparties en 30 vidéos d'entraînement et 20 vidéos de validation représentant divers objets courants. Seul l'objet principal en mouvement est annoté dans la vérité terrain. Nous utilisons les critères officiels d'évaluation sur ce jeu de données, c'est-à-dire l'indice de Jaccard et le score de précision des contours.

**SegTrackV2** [16] et **FBMS59** [24] comprennent respectivement 14 vidéos (dont 1066 images annotées) et 59 vidéos (dont 720 images annotées), chacune impliquant un ou plusieurs objets en mouvement. Pour FBMS59, nous utilisons pour l'évaluation les 29 séquences de l'ensemble de test. Dans les cas où il y a plusieurs objets en mouvement, nous les regroupons en un seul masque de premier plan pour l'évaluation, comme cela est fait dans [33].

**Moving Camouflaged Animals (MoCA)** [15] présente des animaux camouflés dans des scènes naturelles. Pour une comparaison équitable, nous utilisons le sous-ensemble publié par [33] avec 88 vidéos et 4803 images. Des boîtes englobantes de vérité terrain sont fournies à la place des masques, pour l'évaluation. Par conséquent, nous convertissons notre résultat en une boîte englobante autour de la plus grande région connectée de notre masque de sortie, comme cela est fait dans [33].

À l'exception de quelques exemples dans les jeux de données SegTrackV2 et FBMS59, ces jeux de données se concentrent sur un objet mobile principal. En effet, les vidéos représentent en général un seul objet en mouvement indépendant au premier plan. Par conséquent, la vérité terrain ne comprend que deux segments : l'objet mobile principal au premier plan et l'arrière-plan. Pour être cohérent avec ce mode opératoire, nous appliquons notre méthode avec deux masques, c'est-à-dire $K = 2$. Afin de choisir le masque d'avant-plan, nous nous basons sur une heuristique simple où nous désignons le masque le plus grand comme étant celui d'arrière-plan.

Nous comparons notre méthode avec plusieurs autres méthodes supervisées et non supervisées : MoSeg [33], TIS (deux versions) [9], CIS [34], FTS [25], DyStab [35], ARP [14], MATNet [36] et COSNet [19]. Toutes ces méthodes ont été décrites dans la section 2. Les résultats sont rassemblés dans le tableau 1. Pour une comparaison équitable, nous soulignons plusieurs points dans ce tableau. Tout d'abord, nous distinguons les méthodes entraînées sur des masques de segmentation de vérité terrain tels que COSNet [19] ou MATnet [36], et les méthodes non supervisées. Nous indiquons également les méthodes qui utilisent des descripteurs ou caractéristiques pré-appris de manière supervisée, car cela présente un fort avantage, et qui ne s'intègrent donc pas dans un scénario non supervisé. Ainsi, DyStab [35] a recours à des poids issus d'un réseau de classification supervisée pour initialiser ses réseaux. ARP [14] nécessite un algorithme de détection des frontières de mouvement préalablement entraîné de manière supervisée.

De plus, comme nous évaluons ici la segmentation de mouvement à partir du flot optique, nous dissocions les méthodes qui prennent des images RVB en entrée des méthodes utilisant uniquement le flot optique comme la nôtre. Les méthodes TIS [9] et DyStab [35] proposent des versions avec et sans images RVB prises en entrée, respectivement $\text{TIS}_s$ et $\text{TIS}_0$ d'une part, DyStab-"Stat&Dyn" et DysStab-"Dyn" d'autre part, illustrant l'influence de la modalité d'image sur les résultats finaux. Enfin, nous distinguons pour la méthode CIS [34] la version CIS-"Avec Post" impliquant un post-traitement substantiel (en l'occurrence, un post-traitement basé sur les CRF -Conditional Random Field-), et la version CIS-"Sans Post" sans post-traitement, puisque le post-traitement augmente drastiquement le temps d'exécution rendant la méthode difficilement utilisable pour des applications pratiques. Il est indiqué dans [33] que CIS a un temps d'exécution de 11s par image avec post-traitement contre 0,1s sans.

Le tableau 1 montre que notre méthode surpasse toutes les méthodes comparables sur DAVIS2016 et FBMS59. Notre méthode est la seconde après MoSeg pour les deux autres jeux de données. Elle est même proche de MoSeg et loin devant les autres méthodes comparables pour MoCA. Par méthodes comparables, nous entendons des méthodes entièrement non supervisées, sans post-traitement coûteux. Soulignons que nous entraînons notre méthode sur un jeu de données externe, contrairement à MoSeg et CIS qui sont en outre entraînés sur des données de test [2] [3].

Dans la Fig.2, nous présentons des résultats visuels pour montrer comment notre méthode se comporte sur différents exemples typiques des données de DAVIS2016. Sont également présentés quatre "cas d'échec partiel" (scooter-black, kite-surf, blacksan et parkour) par rapport à la vérité terrain de DAVIS2016, même si les parties supplémentaires segmentées par notre méthode ont un sens par rapport à

1. https ://davischallenge.org/index.html

2. https ://github.com/charigyang/motiongrouping/
3. https ://github.com/antonilo/unsupervised_detection

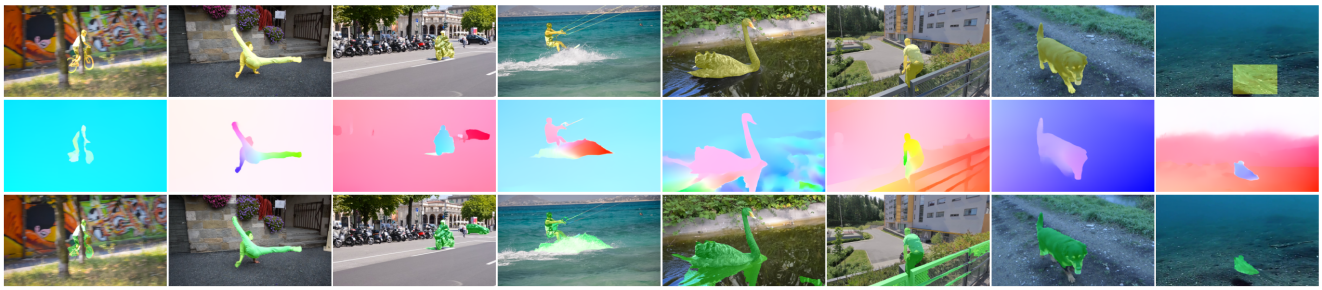

FIGURE 2 – Exemples de résultats de segmentation de mouvement obtenus par notre méthode avec deux masques, sur les vidéos bmx-trees, breakdance-flare, scooter-black, kite-surf, blackswan, parkour, dogs02 et cuttlefish1, des jeux de données DAVIS2016, FBMS et MoCA. Première ligne : une image de la vidéo avec la vérité terrain superposée en jaune. Deuxième rangée : le flot optique d'entrée affiché avec le code couleur HSV. Troisième ligne : la segmentation produite par notre méthode superposée en vert sur l'image correspondante.

l'objectif de segmentation du flot optique. Rappelons que la tâche VOS ne prend en compte par construction que l'objet mobile principal et non tous les objets mobiles. Dans Scooter-black, la voiture segmentée en arrière-plan est en mouvement ; dans Parkour, la clôture est segmentée car elle présente un mouvement de parallaxe important ; dans Kite-surf et Blackswan, les ondulations sur l'eau sont également segmentées. Ce type d'exemples complexes sera traité de manière plus appropriée avec la segmentation de mouvements multiples décrite ci-après.

### 5.3 Segmentation multi-masques

FIGURE 3 – Résultats obtenus avec notre méthode pour quatre masques ($K = 4$). Première ligne : une image de la vidéo. Deuxième ligne : flot optique affiché avec le code couleur HSV. Troisième ligne : cartes de segmentation du mouvement avec quatre masques, une couleur par masque (les quatre masques peuvent ne pas être présents s'ils ne sont pas nécessaires). Exemples tirés des jeux de données DAVIS2016, FBMS59 et SegTrackV2 (à savoir, libby, hummingbird, cars5, people2.

Notre méthode peut gérer la segmentation de mouvements multiples par conception. Il s'agit de choisir la valeur $K$ du nombre de segments. Dans la section 5.2, nous n'avons pris en compte que deux masques pour l'évaluation sur les différents jeux de données, car le challenge et la vérité terrain avaient été définis de cette manière. Dans cette sous-section, nous présentons des expériences supplémentaires avec quatre masques ($K = 4$). Les résultats visuels sont présentés à la Fig.3. Ils ont été obtenus sur des vidéos provenant des jeux de données, DAVIS2016, FBMS59 et SegTrackV2. Nous observons que notre méthode peut appréhender des mouvements multiples dans la vidéo et les segmenter correctement. Cette figure comprend un exemple de mouvements articulés (jambes dans la séquence people2), mais aussi des exemples où il y a plusieurs objets se déplaçant indépendamment (séquences hummingbird, cars5, people2). Ces résultats illustrent que nous pouvons traiter correctement avec notre méthode multi-masques des mouvements apparents supplémentaires y compris des mouvements de parallaxe dû à des objets statiques en avant plan avec une caméra en mouvement (séquence libby).

## 6 Conclusion

Nous avons défini une méthode originale non supervisée pour la segmentation du mouvement prenant le flot optique comme entrée. Nous avons exploité le paradigme EM pour définir une fonction de perte bien fondée et l'étape d'entraînement de notre réseau de neurones. Aucune annotation manuelle n'est nécessaire. Nous avons également conçu un schéma d'augmentation des données simple et efficace, adapté aux champs de vecteurs de flot optique. Contrairement à l'algorithme EM classique, notre méthode n'est pas itérative au stade de l'inférence et ne dépend donc pas de l'initialisation des paramètres des modèles de mouvement. En effet, l'estimation de ces derniers n'est plus nécessaire à l'inférence. En outre, notre méthode peut gérer par conception la segmentation de mouvements multiples. Notre méthode surpasse les méthodes non supervisées comparables sur plusieurs benchmarks et est également très rapide. Les travaux futurs porteront sur le traitement de la dimension temporelle du problème de la segmentation du mouvement.

# Références


[1] V. Badrinarayanan, A. Kendall, and R. Cipolla. SegNet : A deep convolutional encoder-decoder architecture for image segmentation. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 39(12) :2481-2495, December 2017. 1

[2] S. Baker, D. Scharstein, J. P. Lewis, S. Roth, M. J. Black, and R. Szeliski A Database and Evaluation Methodology for Optical Flow In *International Journal of Computer Vision*, 2011. 4

[3] J.T. Barron. A general and adaptive robust loss function. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, Long Beach, June 2019. 5

[4] P. Bouthemy and E. François. Motion segmentation and qualitative dynamic scene analysis from an image sequence. *Int. Journal of Computer Vision*, 10(2) :157-182, April 1993. 2

[5] J. Cheng, Y.-H. Tsai, S. Wang, and M.-H. Yang. Segflow : Joint learning for video object segmentation and optical flow. In *Int. Conf. on Computer Vision (ICCV)*, Venice, 2017. 2

[6] A. Dave, P. Tokmakov, and D. Ramanan. Towards segmenting anything that moves. In *Int. Conference on Computer Vision Workhops (ICCVW)*, Seoul, 2019. 2

[7] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society : Series B (Methodological)*, 39(1) :1–22, 1977. 1

[8] W. Falcon and K. Cho. A framework for contrastive self-supervised learning and designing a new approach. arXiv preprint arXiv :2009.00104, 2020. 6

[9] B. Griffin and J. Corso. Tukey-inspired video object segmentation. In *IEEE Winter Conf. on Applications of Computer Vision (WACV)*, Waikoloa Village, January 2019. 2, 6, 7

[10] R.J. Hathaway. Another interpretation of the EM algorithm for mixture distributions. *Statistics & Probability Letters* 4(2) :53-56, 1986. 3

[11] K. He, G. Gkioxari, P. Dollar, R. Girshick. Mask R-CNN. In *Int. Conf. on Computer Vision (ICCV)*, Venice, 2017. 1

[12] S.D. Jain, B. Xiong, and K. Grauman. FusionSeg : Learning to combine motion and appearance for fully automatic segmentation of generic objects in videos. In *Conf. on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, 2017. 2

[13] D. Kingma and B. Jimmy. Adam : A method for stochastic optimization. arXiv preprint arXiv :1412.6980, 2014. 6

[14] Y.J. Koh and C.-S. Kim. Primary object segmentation in videos based on region augmentation and reduction. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 2, 6, 7

[15] H. Lamdouar, C. Yang, W. Xie, and A. Zisserman. Betrayed by motion : Camouflaged object discovery via motion segmentation. In *Asian Conf. on Computer Vision (ACCV)*, Kyoto, 2020. 7

[16] F. Li, T. Kim, A. Humayun, D. Tsai, and J. M. Rehg. Video segmentation by tracking many figure-ground segments. In *International Conference on Comuter Vision (ICCV)*, Sydney, December 2013. 7

[17] D. Liu and J. Nocedal. On the limited memory BFGS method for large scale optimization. In *Mathematical Programming*, 45(1-3) :503-528, 1989. 7

[18] F. Locatello, D. Weissenborn, T. Unterthiner, A. Mahendran, G. Heigold, J. Uszkoreit, A. Dosovitskiy, and T. Kipf. Object-centric learning with slot attention. In *Conference on Neural Information Processing Systems (NeuRIPS)*, 2020. 2

[19] X. Lu, W. Wang, C. Ma, J. Shen, L. Shao, and F. Porikli. See more, know more : Unsupervised video object segmentation with co-attention Siamese networks. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 2, 6, 7

[20] A. Mahendran, J. Thewlis, and A. Vedaldi. Self-supervised segmentation by grouping optical flow. In *European Conf. on Computer Vision Workshops (ECCVW)*, Munich, 2018. 2

[21] N. Mayer, E. Ilg, P. Häusser, P. Fischer, D. Cremers, A. Dosovitskiy and T. Brox. A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 6

[22] E. Meunier and P. Bouthemy. Unsupervised computation of salient motion maps from the interpretation of a frame-based classification network. In *British Machine Vision Conference (BMVC)*, November 2021. 2, 6

[23] K.P. Murphy. *Machine Learning : a Probabilistic Perspective*, MIT Press, 2012. 3, 4

[24] P. Ochs, J. Malik, and T. Brox. Segmentation of moving objects by long term video analysis. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 36(6) :1187-1200, June 2014. 7

[25] A. Papazoglou and V. Ferrari. Fast object segmentation in unconstrained video. In *IEEE International Conference on Computer Vision (ICCV)*, Sydney, December 2013. 2, 6, 7

[26] F. Perazzi, J. Pont-Tuset, B. McWilliams, L. Van Gool, M. Gross, and A. Sorkine-Hornung. A benchmark dataset and evaluation methodology for video object segmentation. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, June 2016. 2, 6, 7

[27] A. Ranjan, V. Jampani, L. Balles, K. Kim, D. Sun, J. Wulff, and M.J. Black. Competitive collaboration : Joint unsupervised learning of depth, camera motion, optical flow and motion segmentation. In *Conf. on Computer Vision and Pattern Recognition (CVPR)*, Long Beach, 2019. 1

[28] O. Ronneberger, P. Fischer, and T. Brox. U-net : Convolutional networks for biomedical image segmentation. In *Int. Conf. on Medical Image Computing and Computer Assisted Intervention (MICCAI)*, Munich, October 2015. 1, 6

[29] H. Song, W. Wang, S. Zhao1, J. Shen, and K.-M. Lam. Pyramid dilated deeper ConvLSTM for video salient object detection. In *European Conference on Computer Vision (ECCV)*, Munich, 2018. 2

[30] Z. Teed and J. Deng. RAFT : Recurrent all-pairs field transforms for optical flow. In *European Conference on Computer Vision (ECCV)*, 2020. 5

[31] J.Y.A. Wang and E.H. Adelson. Representing moving images with layers. *IEEE Transactions on Image Processing*, 3(5) :625–638, Sept.1994. 2

[32] W. Wang, T. Zhou, F. Porikli, D. Crandall, and L. Van Gool. A survey on deep learning technique for video segmentation. arXiv :2107.01153v1, Dec. 2021. 2

[33] C. Yang, H. Lamdouar, E. Lu, A. Zisserman, and W. Xie. Self-supervised video object segmentation by motion grouping. In *International Conference on Computer Vision (ICCV)*, October 2021. 2, 6, 7

[34] Y. Yang, A. Loquercio, D. Scaramuzza, and S. Soatto. Unsupervised moving object detection via contextual information separation. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, Long Beach, 2019. 2, 6, 7

[35] Y. Yang, B. Lai, and S. Soatto, DyStaB : Unsupervised Object Segmentation via Dynamic-Static Bootstrapping In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 2, 6, 7

[36] Zhou, Tianfei, et al. Matnet : Motion-attentive transition network for zero-shot video object segmentation." In *IEEE Transactions on Image Processing 29*, 2020. 2, 6, 7