ÉCOLE DE TECHNOLOGIE SUPÉRIEURE
Université du Québec
Le génie pour l'industrie

Cours SYS843:

Réseau de neurones et systèmes flous

Étude expérimentale

Comparaison des méthodes de détection d'objets saillants par images RGB-D

Présenté à:

Soufiane Belharbi

Par:

Younes Driouache, DRIY

22/12/2021

Table des matières

1. Mise en situation

Domaine d'application

La perception visuelle est la capacité du cerveau à interpréter ce que les yeux voient. Elle est importante dans de nombreuses activités comme la lecture, l'écriture ainsi que la manipulation d'objets. En plus de la perception visuelle, la perception de la profondeur nous permet de voir le monde en trois dimensions, d'identifier facilement et efficacement des objets, d'estimer leurs tailles et leurs orientations dans l'espace. Pour les robots ou encore les véhicules autonomes, la perception visuelle permet d'extraire des informations fondamentales sur l'environnement dans lequel ils évoluent. En effet, elle contribue à augmenter leur autonomie pour la navigation et la localisation dans un environnement inconnu. Cependant, les environnements peuvent être plus ou moins complexes. L'implémentation d'un mécanisme d'attention visuelle est donc nécessaire. Un tel mécanisme permet de déterminer une carte de saillance qui représente l'importance d'un stimuli visuel par rapport à son environnement.

Une approche de la vision par ordinateur qui utilise un tel mécanisme est la détection d'objets saillants (SOD). L'avantage de cette approche, c'est qu'elle permet de focaliser l'attention sur une région de l'image afin de détecter de façon robuste un ou plusieurs objets. Par conséquent, cette approche peut être bénéfique pour des applications telles que les robots, les véhicules autonomes ainsi que la vidéo surveillance. En effet, pour les véhicules autonomes un des points importants est de garantir la robustesse de la reconnaissance des panneaux de signalisation. Étant généralement de couleurs assez vives, ils attirent facilement l'attention ainsi la SOD permet de les détecter de façon plus robuste, ce qui contribue à la sécurité routière. De la même façon, pour la vidéo surveillance, la SOD permet de localiser et de capter l'évolution des humains et des véhicules.

Finalement, grâce à l'émergence des capteurs de profondeur, de nombreuses techniques de détection d'objets saillants utilisant des images RGB-D ont été proposées. Toutes possédant leurs propres défis.

Problématique abordée

Dans la dernière décennie, plusieurs approches pour la détection d'objets saillants par image RGB-D ont été créées. La plupart des approches s'accordent sur l'utilisation de CNN pour l'extraction de caractéristiques, la prédiction des cartes de saillance et sur le fait de devoir fusionner efficacement une image RGB avec sa carte de profondeur. Néanmoins, il existe trois approches principales pour la fusion de ces entrées. La première est tout simplement de fusionner l'image RGB et la carte de profondeur pour former une entrée à quatre canaux qui sera ensuite traitée par un CNN. La deuxième est d'extraire parallèlement les caractéristiques de l'image RGB et ceux de la carte de profondeur puis de les fusionner. La troisième est de prédire les cartes de saillance pour l'image RGB et pour la carte de profondeur puis de les fusionner pour produire la carte de saillance finale.

De façon générale, ces approches ont des résultats prometteurs. Néanmoins, lors de la fusion des entrées, peu d'attention est portée sur le bruit présent sur la carte de profondeur ce qui peut affecter les performances. Ce manque de qualité peut être dû à des limitations du capteur de profondeur ou encore à de fortes variations de luminosités au sein de la scène. De plus, pour certaines applications comme les véhicules autonomes qui ont besoin de réaliser des détections d'objets en temps réel, le fait d'ajouter une dimension spatiale supplémentaire peut augmenter le temps de calcul et la quantité de mémoire à utiliser.

Objectifs du projet

Comme énoncé précédemment, plusieurs approches existent pour réaliser la détection d'objets saillants à l'aide d'images RGB-D. Dans ce projet, je me propose de comparer deux états de l'art récent qui possèdent chacun une approche de fusion différente, mais qui n'ont pas de module d'attention avec deux autres états de l'art qui ont également une approche de fusion différente, mais qui possède un module d'attention. Ces comparaisons permettront de déterminer à quel point les approches avec modules d'attention sont plus performantes et/ou efficaces que les approches sans modules d'attentions.

Structure du rapport

Ce rapport présente la synthèse des différentes approches pour réaliser la fusion d'une image RGB et de sa carte de profondeur pour la détection d'objets saillants ainsi que l'utilisation de l'attention pour aider la fusion. À la suite de cela, les méthodes DFM-Net, UC-Net, JL-DCF et DANet sont présentées en détail. S'en suit une présentation de la méthodologie du projet où seront présentées les bases de données utilisées, les métriques ainsi que le protocole qui a été mis en place. Ensuite, les résultats des expériences seront exposés et discutés. Pour finir ce rapport, une conclusion exposera les points clés de ce projet.

2. Synthèse des techniques

De nombreuses approches pour réaliser la détection d'objets saillants par images RGB-D existent dans la littérature. Toutes ces approches ont pour objectif de fusionner de la façon la plus efficace les caractéristiques de l'image RGB et de la carte de profondeur, car ces deux entrées permettent respectivement de déterminer les couleurs, les textures d'un objet ainsi que sa géométrie (taille, forme …). Il est nécessaire d'avoir une bonne stratégie de fusion afin de récupérer les informations pertinentes et complémentaires de ces entrées afin d'obtenir les cartes de saillance les plus fine pour déterminer l'objet qui attire le plus l'attention.

On distingue deux grandes catégories d'approches : les approches à flux unique et les approches à flux multiple. Les approches à flux unique vont essentiellement avoir un encodeur (CNN) à une seule branche pour extraire les caractéristiques de l'image RGB et de la carte de profondeur. Les méthodes de fusion qui se plient le mieux aux approches à flux unique sont les méthodes d'early fusion. Pour les approches à flux multiple, on peut retrouver des réseaux à deux ou trois branches qui vont traiter indépendamment l'image RGB et la carte de profondeur. Les méthodes de fusion faisant appellent à cette approche sont la late fusion ainsi que la multi-scale fusion. Dans cette section seront présentées les différentes approches de fusion répartie dans chacune des deux grandes catégories ainsi que les méthodes d'attention. Cette section comportera également une présentation plus détaillée pour les méthodes qui seront exploitées dans ce projet.

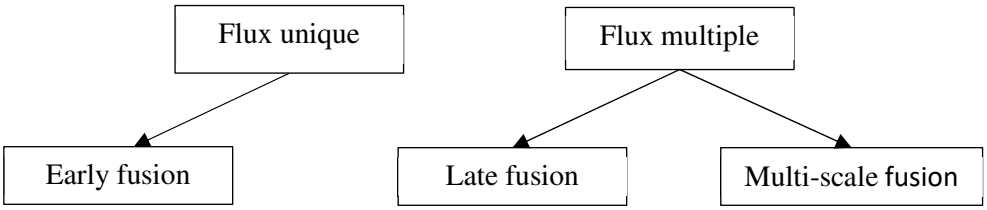

Fig. 1 : Taxonomie des approches de fusion pour la détection d'objets saillant par image RGB-D

Approches à flux unique

Early fusion

La méthode d'early fusion peut être séparée en deux catégories. La première consiste simplement à concaténer l'image RGB et de la carte de profondeur pour créer une entrée à quatre canaux (RGB-D). Par la suite, cette entrée est utilisée par un CNN pour extraire les caractéristiques et prédire la carte de saillance. Dans leur étude *Ziu L. et al.* proposent un réseau convolutionel récurrent où ils utilisent cette approche pour fusionner les entrées [2].

La seconde ne fusionne pas directement l'image RGB avec la carte de profondeur. En effet, ces deux entrées sont envoyées séparément dans un algorithme ou un CNN afin d'extraire leurs caractéristiques à un bas niveau puis ces caractéristiques sont concaténées pour obtenir une représentation croisée des entrées. Par la suite, cette combinaison de caractéristiques est fournie à un CNN afin de produire la carte de saillance. Cette approche a été utilisée par *Qu L. et al.*, ils ont extrait des vecteurs de caractéristiques de saillances (contraste global/local) à la main en segmentant leurs images (et carte de profondeur) en super pixel à l'aide de la méthode SLIC [4]. Ces caractéristiques sont ensuite utilisées en entrée d'un CNN afin de prédire les probabilités de saillances pour chaque super pixels. Finalement, une propagation laplacienne est utilisée pour générer les cartes de saillances finales [3].

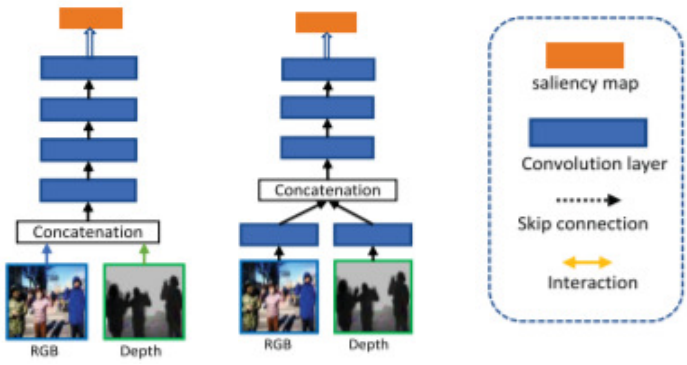

Fig. 2 : Schéma de l'approche Early fusion, source : [29]

Approches à flux multiple

Late fusion

Comme la méthode d'early fusion, la late fusion peut être divisée en deux catégories. La première consiste à faire passer l'image RGB et la carte de profondeur dans deux CNN séparés afin d'extraire leurs caractéristiques à un haut niveau. Par la suite, elles sont fusionnées pour générer la carte de saillance. Cette première catégorie peut être qualifiée de later feature fusion ou encore de middle fusion. Dans [5], cette approche est utilisée. Une fois que les vecteurs des caractéristiques de l'image et de la carte de profondeur sont extraits, un « fully connected layer » est utilisé pour les regrouper/fusionner afin d'obtenir les caractéristiques RGB-D finales et prédire la carte de saillance.

La seconde utilise également deux CNN parallèlement, un pour l'image RGB et un pour la carte de profondeur. La différence cette fois-ci, c'est que les deux réseaux vont prédire les cartes de saillances pour l'entrée RGB et pour la carte de profondeur. Par la suite, les deux cartes de saillances sont fusionnées pour générer la carte de saillance finale. Cette seconde catégorie de peut également être qualifiée de late result fusion. L'étude de *Ding Y. et al.* illustre bien cette approche. Elle permet également d'avoir une vue plus détaillée sur la façon dont les cartes de saillances sont fusionnées [11].

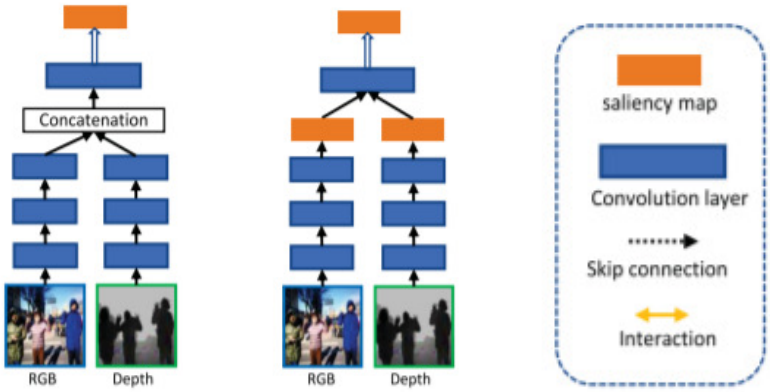

Fig. 3 : Schéma de l'approche Late fusion, source : [29]

Multi-scale fusion

Cette méthode utilise également deux branches parallèles de CNN afin de traiter l'image RGB et la carte de profondeur. À l'instar de la late fusion, cette approche permet de partager et de fusionner les caractéristiques qui sont extraites par les CNN tout au long de l'apprentissage afin d'exploiter efficacement la complémentarité entre les entrées RGB et la carte de profondeur. Tout comme les autres approches, celle-ci peut être séparée en deux catégories.

La première consiste à apprendre les interactions entre les caractéristiques de l'image RGB et la carte de profondeur puis de les fusionner dans un réseau « fully connected ». Le fait d'apprendre les interactions entre les caractéristiques sur plusieurs couches du réseau permet d'améliorer l'apprentissage sur la branche de la carte de profondeur et également de rendre possible une complémentarité entre les caractéristiques de bas et de haut niveau [6].

La seconde consiste à directement fusionner les caractéristiques de l'image RGB et de la carte de profondeur pour chacune des couches. Par la suite, à l'aide de « skip connection », les caractéristiques qui ont préalablement étaient fusionnées sont utilisées en entrée du décodeur afin de prédire la carte de saillance. *Li G. et al.* ont élaboré une méthode qui suit cette approche [12].

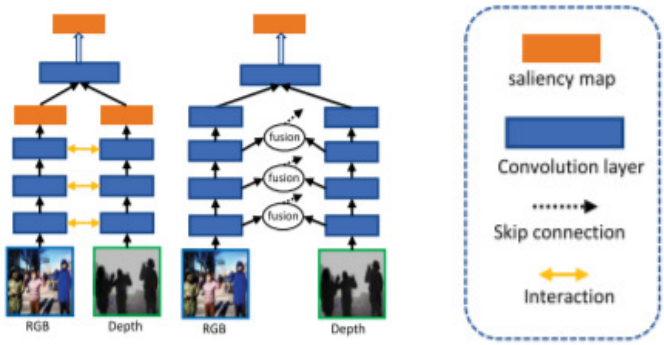

Fig. 4 : Schéma de l'approche Multi-scale fusion, source : [29]

Méthode d'attention

Dans la plupart des approches qui ont été énoncées précédemment, les caractéristiques des différentes régions de l'image qui ont été extraites sont toutes utilisées de la même façon lors de la fusion. De plus, certaines méthodes s'appuient fortement sur les cartes de profondeur qui peuvent être de faible qualité. Ce qui peut engendrer des prédictions de carte de saillance plus ou moins mauvaise. Chacune des régions de l'image peut donc avoir une influence différente sur la prédiction de la carte de saillance.

Pour régler ce problème, *Zhou X. et al.* ont introduit un mécanisme d'attention qui se base sur l'amélioration du contraste ([7], [8]) sur les cartes de profondeur afin de faire ressortir l'objet présent au premier plan. Une fois les cartes d'attention calculées, elles sont introduites entre les différents blocs de convolution pour la branche RGB et celle de la carte de profondeur afin de guider l'extraction de caractéristiques (encodeur) et la prédiction de la carte de saillance (décodeur) [9].

Dans leur étude, *Zhang Z. et al.* ont présenté un mécanisme d'attention bilatéral qui permet grâce aux caractéristiques qui ont été extraites de mettre en évidence le premier plan et l'arrière-plan de l'image. Leur module d'attention traite donc ces deux plans de façon séparée afin d'éliminer les surplus d'informations qui pourraient être présents dans ces plans [10].

Présentation détaillée des méthodes du projet

L'acquisition des cartes de profondeur n'étant pas parfaite, elles peuvent parfois contenir une quantité importante de bruit ce qui peut réduire considérablement leur qualité et par conséquent impacter la détection de l'objet saillant sur l'image.

Ce projet a pour objectif de comparer quatre méthodes avec des approches de fusion différente et la présence ou non de module d'attention afin de comparer leurs comportements face à des situations où la détection peut être compliquée tels que l'exemple énoncé plus haut.

JL-DCF ou « Joint Learning and Densely-Cooperative Fusion »

Cette méthode est divisée en deux parties, « Joint Learning » et « Densely-Cooperative Fusion » (Fig. 6). Elle propose une nouvelle façon d'extraire les caractéristiques et de réaliser la middle fusion (ou late feature fusion). Contrairement aux méthodes qui ont pu être citées dans les paragraphes précédents qui font l'usage de deux encodeurs indépendants pour traiter l'image RGB et la carte de profondeur, JL-DCF utilise un réseau siamois qui permet de partager les poids entre les deux parties du réseau afin de faire ressortir leurs similarités (Fig. 5). Le réseau siamois se trouve dans la partie grise de la Fig. 6, c'est le « Joint Learning ».

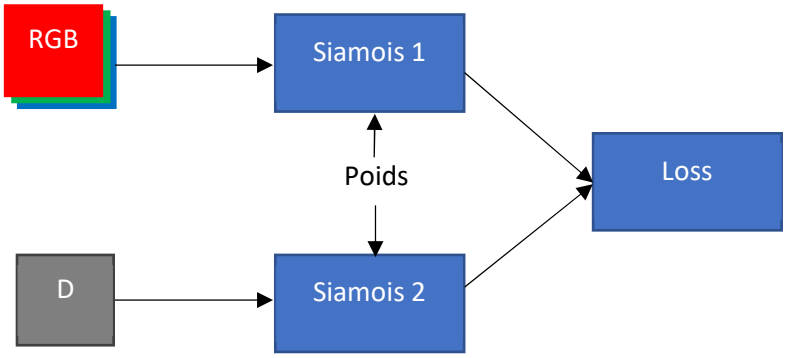

Fig. 5 : Schéma d'un réseau de neurones siamois

Pour ce qui est de la fusion des caractéristiques, c'est la partie verte (Fig. 6) nommée « Densely-Cooperative Fusion » qui remplit ce rôle. Les couches présentent dans la partie DCF sont disposées de façon à ce que la complémentarité entre les modalités RGB et de profondeur puissent être explorée grâce à leurs caractéristiques. La fusion des caractéristiques croisées est réalisée grâce à un module de fusion intermodal qui sera détaillé plus tard. Les deux parties de cette approche vont être détaillées dans les paragraphes suivants.

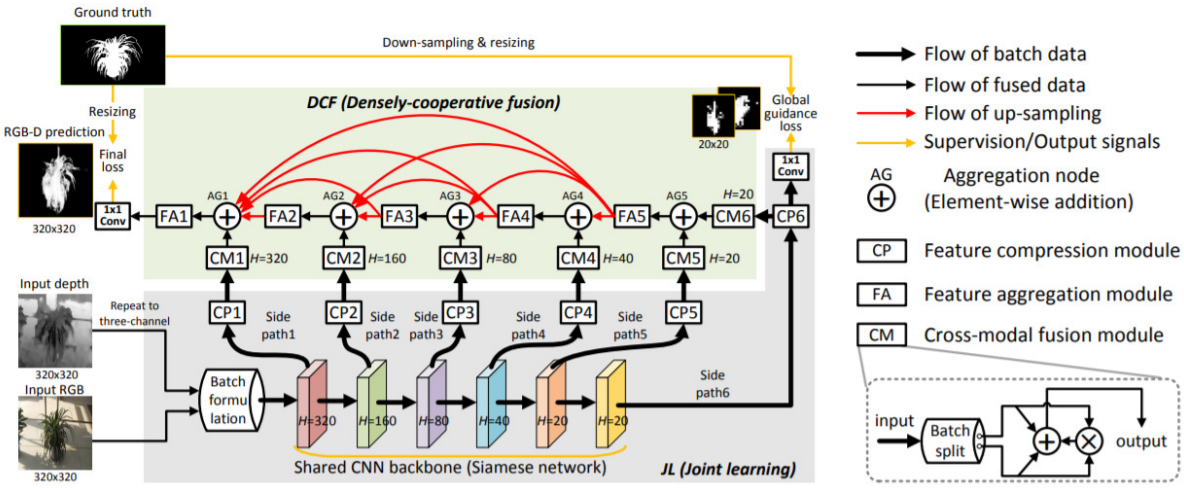

Fig. 6 : Diagramme de la méthode JL-DCF, source : [13]

La partie « Joint Learning » prend en entrée l'image RGB ainsi que la carte de profondeur. L'utilisation d'un réseau siamois implique que les entrées soient de la même dimension. Par conséquent, la carte de profondeur (nuance de gris) est normalisée dans l'intervalle [0, 255] puis elle est empilée trois fois afin d'obtenir une carte de dimension 320x320x3 (Fig.7).

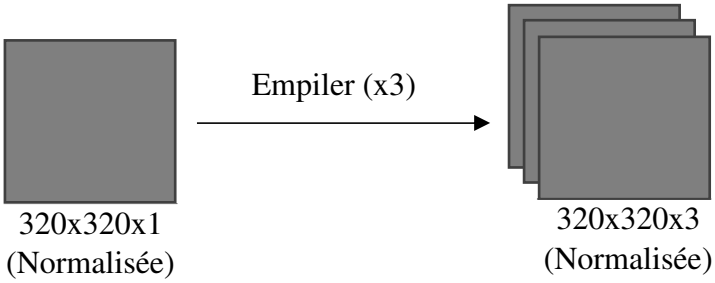

Fig. 7 : Schéma du prétraitement de la carte de profondeur

Par la suite, l'image RGB et la carte de profondeur sont concaténées de telle sorte à créer un lot afin que le réseau puisse réaliser l'extraction des caractéristiques des deux entrées. Contrairement aux autres méthodes d'early fusion, la concaténation se fait sur le quatrième canal et non le troisième. Ainsi, les dimensions d'un lot sont 320x320x3x2 au lieu de 320x320x6.

Les caractéristiques hiérarchiques extraites par l'encodeur siamois ont toute une résolution différente. Un module de compression (CP) est donc utilisé pour uniformiser la taille de toutes les caractéristiques. L'utilisation d'un tel module a pour avantage de diminuer le temps de calcul ainsi que la mémoire utilisée pour décoder les caractéristiques.
Selon *Fu K. et al.*, la localisation grossière de la région saillante permet de guider l'encodeur dans l'extraction simultanée des caractéristiques hiérarchique indépendante de l'image RGB et de la carte de profondeur. Par conséquent, une couche de convolution est placée à la sortie du module CP6 (Fig. 6) afin d'obtenir la carte de localisation grossière de l'objet. La sortie de cette couche (la carte de localisation grossière) est supervisée par la groundtruth sous-échantillonnée afin guider l'encodeur dans son extraction de caractéristiques grâce au calcul de la « global guidance loss ».

La partie DCF réalise la fusion intermodale des caractéristiques à plusieurs échelles. Elle prend en entrée les caractéristiques issues des modules de compression CP à différentes échelles. Pour chaque module de compression, un module de fusion intermodal a été mis en place. Son rôle est de séparer le lot en deux, caractéristiques RGB d'un côté et caractéristiques de profondeur de l'autre pour ensuite les fusionner en les additionnant et en les multipliant (Fig. 6 : légende CM). L'addition exprime la complémentarité des caractéristiques et la multiplication met leur point commun en évidence. *Fu K. et al.* nomme cette fusion la « cooperative fusion ». Les caractéristiques qui ont été fusionnées dans les modules CM1-CM6 sont utilisées en entrée du décodeur qui possède des connexions denses ce qui permet un meilleur mélange des caractéristiques RGB et de profondeur à différentes échelles [14]. Finalement, un module d'agrégation de caractéristiques (FA) est utilisé entre chaque niveau afin de regrouper les différentes caractéristiques fusionnées de FA5 à FA1 (Fig. 6).

UC-Net ou Uncertainty Inspired RGB-D Saliency Detection via Conditional Variational Autoencoder

La méthode UC-Net a pour objectif de représenter l'incertitude humaine qui peut être présente lors de la labélisation des cartes de saillance. Cette méthode probabiliste de détection d'objets saillants se base sur un « conditional variational autoencoder » ou CVAE qui apprend la distribution des cartes de saillance plutôt qu'une simple prédiction. UC-Net est composé de quatre principaux modules :

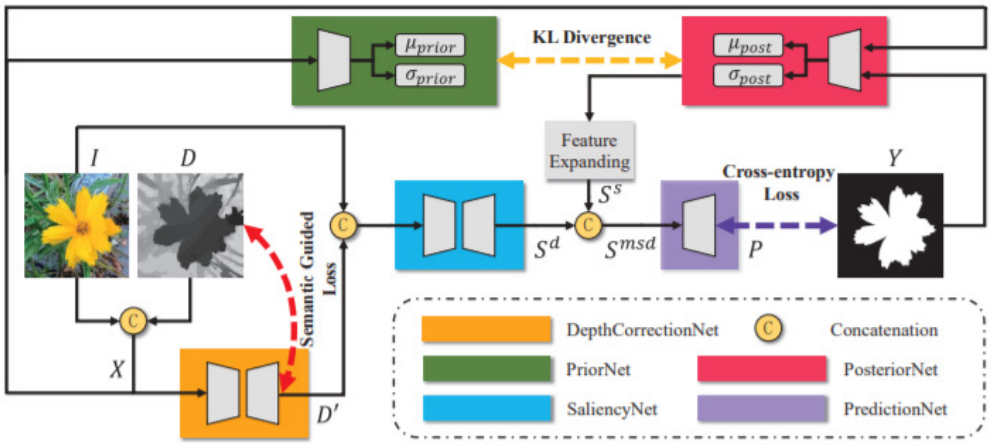

Fig. 8 : Diagramme de la méthode UC-Net, source : [15]

1) LatentNet : Ce module est le CVAE. Il est composé de deux parties nommées PriorNet et PosteriorNet. Le PriorNet cartographie la paire d'entrées RGB-D notées X dans un espace latent à faible dimension. Le PosteriorNet cartographie la sortie Y et l'entrée X dans l'espace latent à faible dimension.

   La structure du PriorNet et du PosteriorNet est identique, elle est composée de 5 couches de convolutions (Fig. 9) afin de cartographier les X dans l'espace latent gaussien où μ et σ représente l'écart type et la moyenne de la distribution produite dans l'espace latent.

Fig. 9 : Structure PriorNet/PosteriorNet, source : [15]

Pour permettre au PriorNet d'encoder des variants de label pour une entrée X pendant l'entrainement, il est nécessaire d'utiliser plusieurs versions de groundtruth Y afin d'obtenir des labels multiples pour un même exemple dans l'espace latent. Or, les

datasets de détection d'objets saillant avec image RGB-D ne possèdent qu'une unique groundtruth par image. *Zhang J. et al.* ont donc créé leur propre datasets. Pour se faire, ils ont caché l'objet saillant dans l'image RGB puis ils ont utilisé un modèle de détection d'objets saillant sur image RGB [16] afin de produire une nouvelle carte de saillance. Ce procédé est répété trois fois pour chaque exemple d'entrainement (Fig. 10).

Fig. 10 : Génération des nouveaux label en cachant itérativement les régions saillantes, source : [15]

2) SaliencyNet : Ce module produit une carte de saillance déterministe (S^d) grâce à l'image RGB et la carte de profondeur qui a été corrigée/affinée par le module DepthCorrectionNet. L'extraction des caractéristiques se fait par un encodeur basé sur VGG16 [17]. En segmentation comme en détection d'objet saillant, la taille des objets peut varier grandement, ce qui peut poser des problèmes pour les caractéristiques de haut niveau, car elles portent les informations sur de multiples échelles. Ainsi, *Zhang J. et al.* ont supprimé toutes les couches après la cinquième couche de pooling pour y ajouter le module DenseASPP [18] afin de générer des caractéristiques avec des champs visuels plus larges tout en conservant la résolution spatiale (Fig. 11).

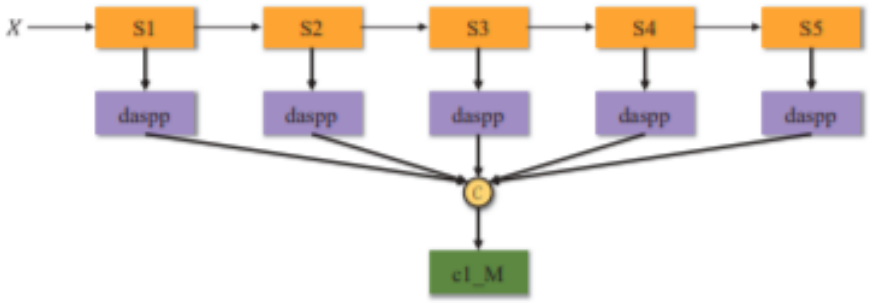

Fig. 11 : Structure de l'encodeur du module SaliencyNet, les blocs en jaune correspondent aux différents blocs du réseau VGG16, source : [15]

3) DepthCorrectionNet : Comme énoncé un peu plus haut, les données de profondeur ne sont généralement pas parfaitement acquises. En effet, elles comportent du bruit. Le module DepthCorrectionNet a pour objectif de débruiter la carte de profondeur. Pour se faire, un réseau composé d'un encodeur similaire à celui du module SaliencyNet et d'un décodeur a été mis en place. Selon *Zhang J. et al.*, pour corriger efficacement la carte de profondeur il faut que les contours des données de profondeur ainsi que celle de l'image RGB soient alignés. Ils ont donc implémenté la boundary IOU loss [19] pour régulariser DepthCorrectionNet afin d'obtenir une carte de profondeur affinée.

4) PredictionNet : Les modules LatentNet et SaliencyNet qui ont été présentés précédemment produisent respectivement les caractéristiques stochastiques (S^S) et déterministe (S^d). Le rôle du PredictionNet est de fusionner ces caractéristiques au niveau de leurs canaux de tel sorte que le réseau ne puisse pas faire la distinction entre les caractéristiques stochastiques et déterministes. Cette fusion produit une nouvelle carte de caractéristiques S^{sd} qui a pour dimension K+M où K correspond à la taille de l'espace latent (LatentNet) et M correspond à la taille des canaux de S^d. Par la suite, ces canaux sont mélangés en fonction d'une variable de classement défini par *Zhang J. et al.* pour obtenir un carte de caractéristiques mixée S^{msd} . Finalement, S^{msd} passe par trois couches de convolution afin de prédire la carte de saillance finale P.

Il existe un cinquième module qui est présent uniquement pendant la phase de test. Durant cette phase, plusieurs prédictions sont réalisées pour l'exemple à tester afin d'obtenir la distribution de saillance. Ainsi, le Saliency consensus module a pour objectif de déterminer de façon probabiliste la prédiction la plus représentée dans la distribution de saillance.

DANet

La méthode DANet explore l'utilisation de la carte de profondeur pour guider l'early fusion et la middle fusion. La carte de profondeur a comme particularité de mettre en évidence les différents niveaux de contraste assez aisément. Ainsi, la segmentation entre le premier et l'arrière-plan d'une image RGB est plus facile lorsqu'elle est guidée par la carte de profondeur correspondante. DANet introduit donc un mécanisme de filtre spatial qui utilise la carte de

profondeur afin de discriminer les caractéristiques qui appartiennent au premier plan et celles qui appartiennent à l'arrière-plan.

La structure de la méthode est basée sur le « Feature Pyramid Network » [20] qui est utilisée dans le fast-RCNN pour la détection d'objets.

Contrairement aux méthodes détaillées plus haut, DANet utilise un encodeur à flux unique, c'est-à-dire que les entrées RGB et de profondeur sont traitées dans le même du réseau. Ces entrées sont donc fusionnées en suivant l'approche d'early fusion pour former une entrée à quatre canaux (RGB-D). L'encodeur est basé sur VGG-16 et utilise les poids préentrainés sur ImageNet. Or le fait d'avoir une entrée à quatre canaux n'est pas compatible avec les poids ImageNet. Zhao X. et al. ont donc initialisé une première couche de convolution pour prendre en entrée du RGBD et renvoyé une carte de caractéristiques à 64 canaux qui va pouvoir recevoir les poids ImageNet (Fig. 12). Afin de conserver le plus possible les informations issues des couches les moins profondes, la dernière couche de pooling ainsi que les couches fully connected sont retirées.

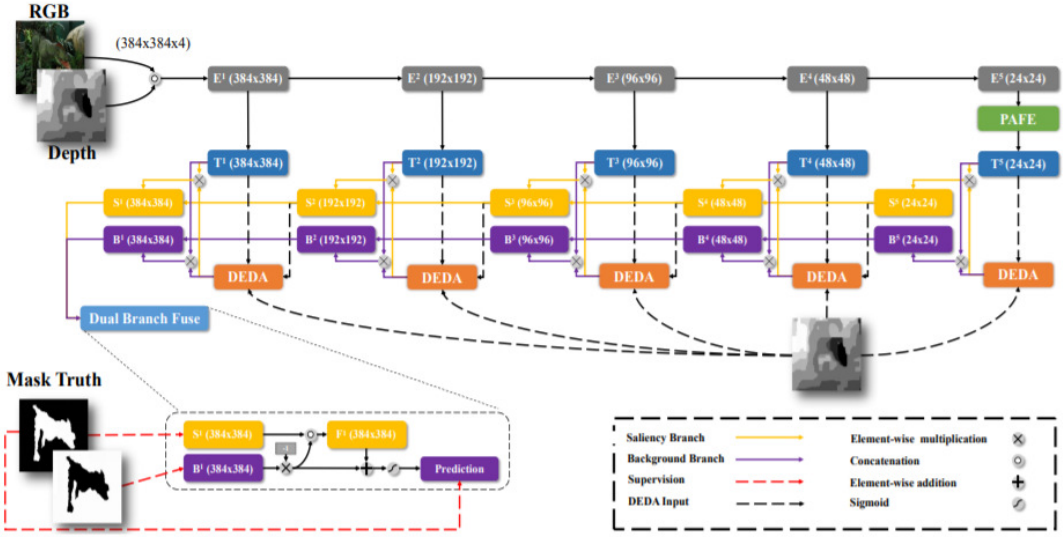

Fig. 12 : Diagramme de la méthode DANet, source : [21]

Zhao X. et al. ont élaboré un module d'attention pour profiter du fait que la carte de profondeur met en évidence les niveaux de contrastes, « Depth-enhanced Dual Attention Module ». Ce module d'attention est utilisé pour produire des caractéristiques qui mettent en avant les différents niveaux de contrastes. Néanmoins, il se peut que la portée de la profondeur sur la carte soit trop large ou bien que le premier plan et l'arrière-plan soient à la même profondeur. Pour pallier à ce problème, *Zhao X. et al.* ont fait le choix de superviser ce module avec la groundtruth afin de filtrer les informations qui peuvent induire en erreur le réseau. La Fig. 13

permet de visualiser les différentes opérations qui sont réalisées pour obtenir les cartes d'attention de premier et de l'arrière-plan.

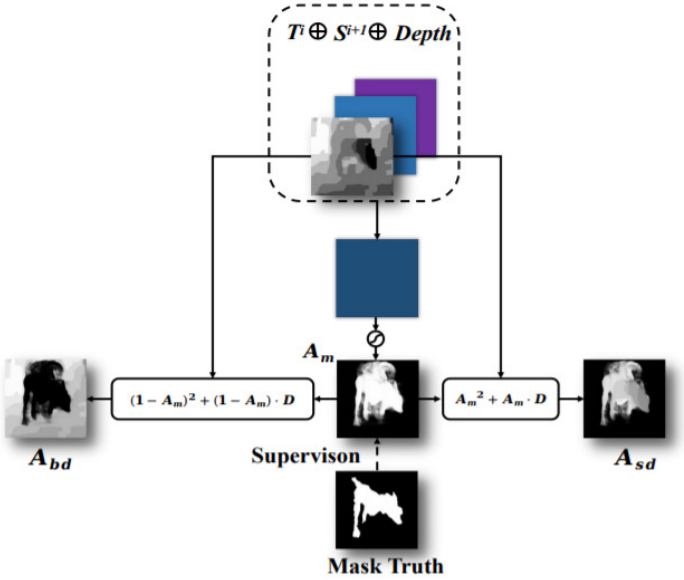

Fig. 13 : Opérations réalisées dans le module d'attention (DEDA), source : [21]

En segmentation tout comme en détection d'objets saillants, l'échelle des objets peut varier. Il est donc nécessaire d'utiliser des caractéristiques multiéchelles afin d'extraire des informations sur le contexte des objets à différentes échelles. Pour traiter efficacement ces caractéristiques, *Zhao X. et al.* leur ont appliqué un module d'attention spatiale à chaque échelle dans le but de se concentrer sur les régions de l'image les plus importantes. Ce module se nomme : « Pyramidally Attended Feature Extraction module » ou PAFE (Fig. 14).

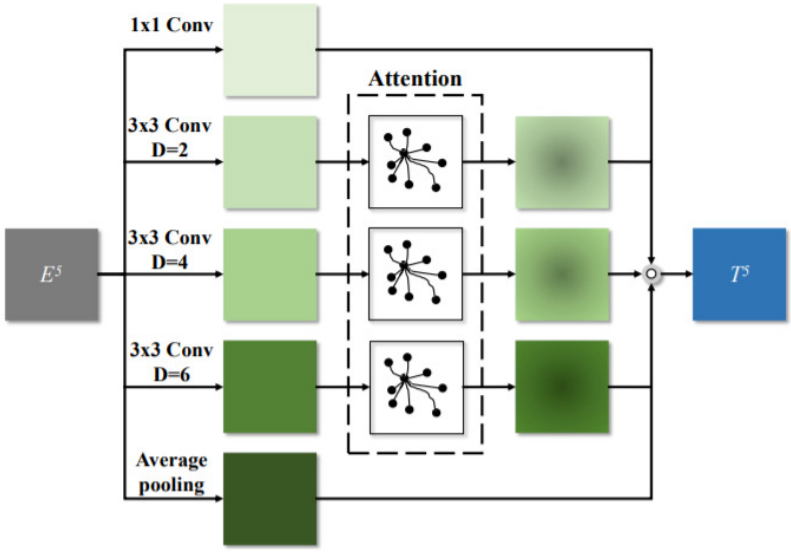

Fig. 14 : Module d'attention spatiale PAFE, source : [21]

DFM-Net ou « Depth Feature Manipulation Network »

La technique DFM-Net permet de discriminer les cartes de profondeur en fonction de leur qualité afin de les traiter différemment dans le but de limiter l'impact des cartes de profondeur de mauvaise qualité sur les performances.

Pour déterminer leur qualité, *Zhang W. et al.* se sont basés sur le fait que le niveau de qualité d'une carte de profondeur dépend principalement de son alignement avec l'image RGB correspondante. Ils ont donc appliqué un détecteur de contours sur l'image RGB et sur la carte de profondeur. Pour quantifier l'alignement entre ces deux cartes, ils ont calculé le « Dice coefficient » [23]. Cette quantification leur permet de déterminer si la profondeur est de bonne qualité ou non et de la manipuler en fonction.

DFM-Net est composé d'un encodeur comportant deux branches, une pour extraire les caractéristiques de l'image RGB et l'autre pour les caractéristiques de profondeur. La branche qui traite les données RGB est basée sur MobileNet-v2. Pour ce qui est de la branche qui traite les données de profondeur, *Zhang W. et al.* se sont basés sur une version allégée du « inverted residual bottleneck » de MobileNet-v2 [24], car les données de profondeur possèdent moins d'informations que les données RGB, ce qui permet de diminuer les calculs. Entre la branche RBG et celle de la profondeur se trouve un module qui permet de manipuler les caractéristiques de profondeur en lui appliquant des coefficients calculés grâce à deux opérations : « Depth Quality-inspired Weighting » et « Depth Holistic Attention » (Fig. 15, partie orange).

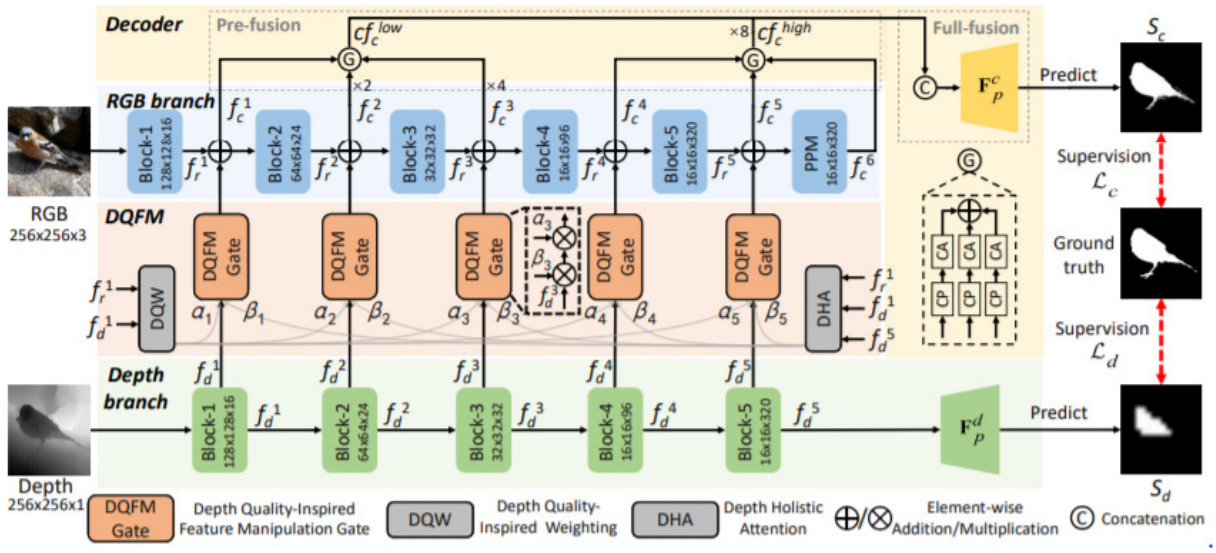

Fig. 15 : Diagramme de la méthode DFM-Net, source : [22]

Le module « Depth Quality-inspired Weighting » prend en compte uniquement les caractéristiques de bas niveau, car c'est elles qui représentent le mieux les contours. En se basant sur le processus énoncé au paragraphe précédent, DQW est en mesure de prédire les coefficients α à appliquer pour chacun des caractéristiques issues des cinq blocs de convolution de la branche de profondeur (Fig. 16).

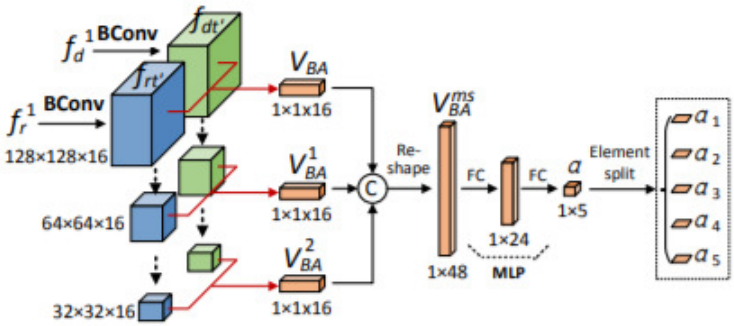

Fig. 16 : Architecture module DQW, source : [22]

Le module « Depth Holistic Attention » produit des cartes d'attention β pour mettre en valeur les caractéristiques de profondeur. La détermination de la carte d'attention commence par une localisation grossière de la région saillante. Ce sont les caractéristiques de haut niveau qui fournissent cette information (f_d^5). La multiplication des caractéristiques de bas niveau f_r^1 et f_d^1 permet de mettre en évidence les contours communs entre l'image RGB et la profondeur. La carte de caractéristique résultant de cette multiplication permet de recalibrer les cartes de caractéristiques issues de la branche qui traite f_d^5 (Fig. 17).

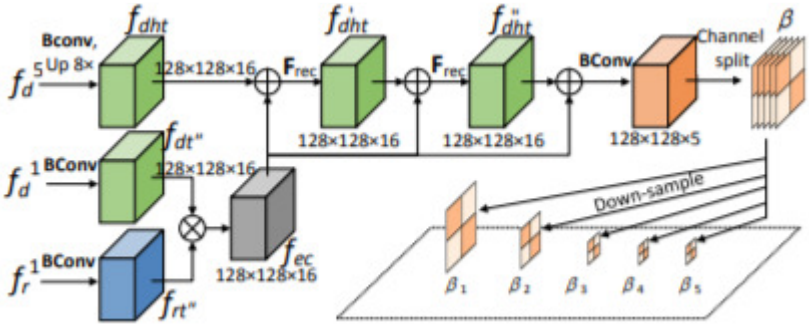

Fig. 17 : Architecture module DHA, source : [22]

Les cartes d'attention sont obtenues pour les cinq caractéristiques hiérarchiques du réseau sont obtenues en downsamplant β (Fig. 18).

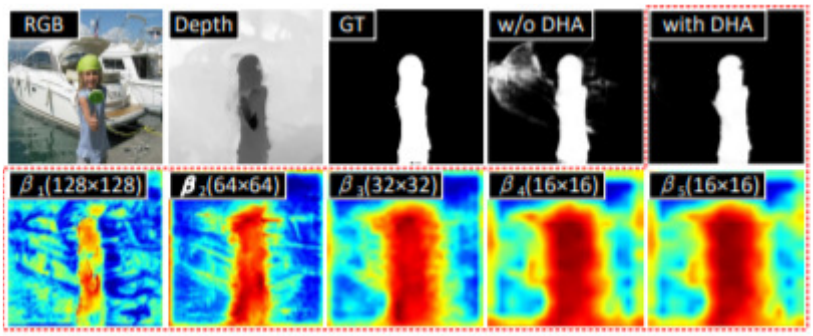

Fig. 18 : Exemple des cartes d'attention et de leur carte de saillance, source : [22]

Une fois que les coefficients α et β sont calculés, ils sont appliqués aux caractéristiques de profondeur grâce au DQFM Gate par multiplication puis sont fusionnés dans la branche RGB avec les caractéristiques f_r par addition (Fig. 15).

À la fin de la branche RGB de l'encodeur, il y a un « Pyramid Pooling Module » [25] qui a pour objectif de récolter les informations sémantiques à différentes échelles. Finalement, les différentes caractéristiques f_c et la caractéristique en sortie du PPM est envoyée au décodeur qui les fusionne en deux étapes :

- Pré-fusion :

  Dans cette première étape de fusion, les canaux des trois caractéristiques de bas niveau sont compressés puis subissent une opération d'attention qui va pondérer les différents canaux des caractéristiques pour les mettre en valeur. Finalement, les caractéristiques issues de ces deux opérations sont fusionnées par addition. Les mêmes opérations sont appliquées aux trois caractéristiques de haut niveau (Fig. 15, bloc jaune decodeur). La pré-fusion permet de passer de six cartes caractéristiques hiérarchiques à deux cartes caractéristiques (bas niveau et haut).
- Fusion complète :

  La fusion complète a le simple rôle de concaténer les cartes de caractéristiques établies lors de la pré-fusion pour finalement prédire la carte de saillance.

3. Méthodologie expérimentale

Cette partie présentera la méthodologie utilisée pour mener les différentes expériences. Les bases de données, les métriques utilisées ainsi que le protocole expérimental seront mis en avant. Ces informations permettront d'assurer la reproductibilité des expériences.

Bases de données

Pour comparer les différentes méthodes, la première étape est de choisir les bases de données à utiliser pour les expérimentations.

Dans la littérature, les principales bases de données utilisées sont :

Base de données	Nb exemples	Résolution
STERE	1000	[251-1200]x[222-900]
DES	135	640x480
NLPR	1000	640x480, 480x640
LFSD	100	360x360
NJU2K	1985	[213-1213]x[274-828]
SIP	929	922x774

Tab. 1 : Présentation des bases de données de la littérature utilisées pour ce projet

Ces bases de données sont composées des images RGB, des cartes de profondeur ainsi que des vérités terrain. En s'inspirant de la méthodologie expérimentale de chacune des approches étudiées. L'ensemble d'entrainement a été établi en utilisant 1500 exemples de la base de données NJU2K et 700 exemples de NLPR. L'ensemble de test est composé des données restantes de NJU2K et NLPR ainsi que des données de DES, LFSD, STERE et SIP.

Néanmoins, l'ensemble de test est organisé de sorte à pouvoir investiguer l'influence de différents facteurs challengeant tels que la taille des objets à détecter, la multiplicité des objets, la variation de luminosité ainsi que la complexité de l'arrière-plan.

Taille des objets saillants (Objectscale) :

Pour déterminer la taille des objets saillants, un ratio entre l'aire de l'objet saillant et la taille de l'image a été calculé sur la vérité terrain. Les différentes tailles ont été définies de la façon suivante :

- Lorsque le ratio est inférieur à 0.1, l'objet est petit (small) ;
- Lorsque le ratio est compris entre 0.1 et 0.4 l'objet est de taille moyenne (medium) ;
- Lorsque le ratio est supérieur à 0.4, l'objet est grand (large).

Cette base de données de test utilise les exemples de STERE, LFSD, DES et SIP.

Multiplicité des objets (Multipleobjects) :

Cette base de données de test a été construite avec les exemples restants de NLPR et les exemples de SIP en utilisant l'image de vérité terrain pour déterminer s'il y avait un ou plusieurs objets saillants. Les sous-ensembles de donnée de test sont donc « Single » et « Multi ».

Variation de luminosité (Illumination) :

Pour former cette base de données, uniquement les exemples de la base SIP ont été utilisés, car ils étaient déjà organisés en exemple à forte luminosité « High » et faible luminosité « Low ».

Complexité de l'arrière-plan (ComplexBackground) :

C'est grâce aux auteurs de l'article [29] que nous avons pu avoir accès à cette base donnée. Pour élaborer cette base de données, ils ont utilisé des méthodes classiques de détection d'objets saillants par image RGB, car elles ont tendance à utiliser le contraste des couleurs ou des informations préalables pour localiser l'objet saillant. De façon général, ces méthodes à se ratent lorsque l'arrière-plan est complexe. Ils se sont donc basé sur la métrique S-measure [27] pour catégoriser la complexité d'une image :

- Lorsque la S-measure est supérieure à 0.9, l'arrière-plan de l'image est « simple » ;
- Lorsque la S-measure est inférieure à 0.6, l'arrière-plan de l'image est « complex » ;
- Lorsque la S-measure est comprise entre 0.6 et 0.9, l'arrière-plan de l'image est « uncertain ».

Cette base de données de test utilise les exemples restants de NLPR ainsi que les exemples de STERE et LFSD.

Métriques

Pour ce projet, deux types de métriques seront utilisées. Les métriques permettant de quantifier les performances d'une méthode et les métriques permettant de quantifier leur efficacité.

Métrique de performance :

La métrique Précision-Rappel sera utilisée pour mettre en évidence la capacité des différentes méthodes à bien classifier les pixels appartenant ou non aux objets saillants.
Pour une carte de saillance donnée S, on peut obtenir un masque binaire M en convertissant S. Le calcul de la précision est du rappel se base donc sur M (masque binaire) et G (vérité terrain). [29]

$$Précision = \frac{|M \cap G|}{|M|}, \qquad Rappel = \frac{|M \cap G|}{|G|}$$

La métrique F_β-measure sera utilisée pour exprimer l'accuracy de la méthode. En effet, étant donné que sur une carte de saillance nous avons un déséquilibre entre les pixels saillants (=1) et non saillants (=0), il est donc nécessaire d'utiliser la F_β -measure. [26]

$$F_\beta = (1 + \beta^2) \frac{précision * rappel}{\beta^2 précision + rappel}$$

Ici, $\beta^2 = 0.3$ afin de donner plus de poids à la précision. Les valeurs de F_β-measure qui seront présentés dans les résultats seront les max F_β-measure.

La métrique S-measure sera utilisée pour quantifier la capacité d'une méthode à capturer la structure des objets saillants à détecter. [27]

Métrique d'efficacité :

La vitesse d'entrainement qui est quantifié en mesurant le temps d'entrainement de chaque méthode.

La vitesse d'inférence mesurée en image par seconde. Cette métrique est obtenue en divisant le nombre d'images pour lesquels on fait des prédictions par le temps qui s'est écoulé pour réaliser ces prédictions :

$$vitesse\ inférence\ (FPS) = \frac{nombre\ d'images\ dans\ le\ dataset\ de\ test}{temps\ pour\ une\ prédiction}$$

La vitesse d'inférence permet de se rendre compte de si un modèle est potentiellement utilisable pour des applications qui ont besoin de prédictions en temps réel.

La taille des poids des modèles en Mbit. Similairement à la vitesse d'inférence, cette métrique nous permet de déterminer si la méthode peut potentiellement être utilisée dans des systèmes embarqués où il y a généralement peu de capacité de mémoire à allouer pour toute les applications dont le système peut avoir besoin.

Ces métriques nous permettront de comparer les méthodes avec attention et les méthodes sans attention afin de déterminer les avantages et/ou les inconvénients d'utiliser des modules d'attention pour la détection d'objets saillants par image RGB-D.

Protocole expérimental

Le grand nombre de paramètres des modèles utilisés pour ce projet (jusqu'à 143M), nous pousse à utiliser un environnement cloud pour réaliser les calculs, car un simple ordinateur ne peut pas supporter de réaliser autant de calculs ou prendrait énormément de temps. L'environnement Google Colab avec un GPU a donc était utilisé. Le GPU mis à notre disposition était une NVIDIA Tesla P100. Cet environnement m'était également à disposition 25 GB de RAM ainsi que 166 GB d'espace disque. Toutes les expériences ont été réalisées dans cet environnement.

Chacune des méthodes utilisées dans ce projet utilise comme backbone un CNN de tel que VGG16, ResNet ou encore Mobile-Netv2. Les poids pré entrainés sur ImageNet sont donc

utilisés pour initialiser les poids. En initialisant les backbones avec les poids pré entrainés sur Image-Net, on facilite l'extraction des caractéristiques sur nos données.

Pour entrainer chacun des modèles, la base de donnée utilisée est celle qui a été décrite dans la partie « Base de données », elle correspond à l'ensemble comportant 1500 exemples de NJU2K et 700 exemples de NLPR.

Pour ces expérimentations, nous n'avons pas considéré de set de validation, car les données d'entrainement qui ont été utilisées sont identiques à celles que les auteurs ont utilisées pour déterminer les meilleurs hyperparamètres ainsi que l'epoch qui fournit les meilleurs poids.

Le processus d'entrainement pour la méthode UC-Net est le suivant, nous utilisons l'optimizer Adam avec un momentum de 0.9, le taux d'apprentissage est initialisé à 5e-5. La dimension de l'espace latent est fixée à 3. Les caractéristiques de saillance en sortie du générateur sont réduites à 32 channels. L'entrainement nécessite 100 epochs avec un batch size de 10 et les images sont redimensionnées pour faire une taille de 352x352.

Le processus d'entrainement pour la méthode DANet est le suivant, nous utilisons l'optimizer Stochastic Gradient Descent (SGD) avec un momentum de 0.9 et un weight decay de 5e-4, le taux d'apprentissage est initialisé à 1e-3 puis est ajusté avec un learning rate decay de 0.9. L'entrainement nécessite 40 epochs avec un batch size de 4 et les images sont redimensionnées pour faire une taille de 384x384. Pour éviter l'overfitting et donc permettre une meilleure généralisation, des techniques d'augmentation telles que le flip horizontal, la rotation et les changements de luminosité/saturation/contraste sont utilisées.

Le processus d'entrainement pour la méthode DFM-Net est le suivant, nous utilisons l'optimizer Adam avec un taux d'apprentissage de 1e-4. L'entrainement nécessite 300 epochs avec un batch size de 10 et les images sont redimensionnées pour faire une taille de 256x256. Toujours dans une démarche pour éviter l'overfitting et permettre une meilleure généralisation, des techniques de data augmentation telle que le flip horizontal, le crop aléatoire, la rotation, l'ajout de bruit gaussien et les changements de luminosité/saturation/contraste sont utilisées.

Le processus d'entrainement pour la méthode JL-DCF est le suivant, contrairement aux autres méthodes qui sont initialisées avec des poids pré entrainés sur ImageNet, cette méthode est initialisée par des poids qui ont été pré entrainés sur la méthode DSS [30] qui est un modèle de

détection d'objets saillants par image RGB. Nous utilisons l'optimizer Stochastic Gradient Descent (SGD) avec un taux d'apprentissage à 5e-5, un momentum de 0.99 et un weight decay de 5e-4. La technique de flip horizontal est utilisée pour augmenter les données.

À noter que les modèles JL-DCF et DFM-Net n'ont pas pu être entrainés par nous. Malgré le GPU fourni par Google Colab (NVIDIA Tesla P100), le modèle JL-DCF prend environ 30 heures d'entrainement. Nous avons donc téléchargé les poids du modèle déjà entrainé (fourni par les auteurs de la méthode) pour réaliser nos tests sur les différents ensembles de test. Il est possible d'opérer de cette façon pour notre projet, car nous utilisons la même base d'entrainement que les auteurs. Les poids obtenus à l'issue de nos entrainements sont donc les mêmes que les leurs à l'issue de leurs entrainements. Ainsi la phase d'entrainement peut être sautée pour ce modèle. Pour le modèle DFM-Net, les auteurs ont bien fourni les codes pour initialiser le modèle, la préparation des données, le test, etc… mais ils n'ont pas fourni le code d'entrainement. Ainsi, nous avons opéré de la même façon que pour le modèle JL-DCF.

Une fois que tous les entrainements ont été faits, nous pouvons commencer à tester chacun des modèles sur les différentes bases de test énoncés précédemment. Les résultats de ces tests seront analysés dans la section suivante.

4. Résultats de simulation

Dans cette partie, les résultats obtenus à la suite des expérimentations seront présentés puis analysés.

Efficacité des méthodes

Méthodes	Temps d'entrainement	Vitesse d'inférence (FPS)	Taille des poids (Mbit)
JL-DCF	~ 30 h	2	561
UCNet	4 h 8 mins 40 s	29	123
DANet	1 h 25 mins 37 s	41	104
DFMNet	5 h 45 mins 15 s	60	9

Tab. 2 : Métriques d'efficacité calculées pour chaque méthode

Les valeurs pour le temps d'entrainement présentées dans le Tableau.2 sont mesurées à l'issue de l'entrainement complet.

Le Tab.2 met en évidence le fait que les méthodes ne possédant pas de module d'attention sont plus massives que les méthodes avec attention. En effet, pour les méthodes sans attention comme JL-DCF, il est nécessaire d'utiliser des backbones assez massifs comme ResNet101 tandis que pour les méthodes avec attention des backbones plus légers comme Mobile-Netv2 ou encore VGG16 peuvent être utilisés. Le fait que les méthodes avec attention n'aient pas forcément besoin de backbone aussi dense est dû à la capacité des modules d'attention à prioriser certaines régions de la carte de caractéristique pour la prédiction de la carte de saillance.

Par conséquent, on peut voir que la méthode JL-DCF est très lente lors de l'inférence. En effet, elle est 30 fois plus lente que la méthode d'attention DFMNet et 20 fois plus lente que la méthode DANet. Malgré le fait que la méthode UCNet n'ait pas de module d'attention, sa vitesse d'inférence reste correcte. De plus, tout comme la méthode DANet, UCNet est une méthode à flux unique, elle emploie l'Early « input » fusion. Ainsi, il est normal que sa vitesse d'inférence soit sensiblement similaire à celle de DANet. Néanmoins, DANet reste 1.5 fois plus rapide grâce à ses modules d'attention.

Ainsi, il est plus adéquat d'utiliser les méthodes avec attention pour des applications qui ont besoin de résultats en temps réel.

Performances des méthodes

Expérimentation sur la base de données « Objectscale »

Cette expérience a été conduite afin d'évaluer la capacité des différentes méthodes à identifier des objets de petite, moyenne ou grande taille dans une image. Pour obtenir les valeurs de la catégorie « global », l'expérience a été faite sur la base de données au complet sans faire la distinction entre « small », « medium » et « large ». Les métriques considérées ont été énoncées précédemment. Les résultats obtenus sont les suivants :

	Catégorie	DANet	DFMNet	JL-DCF	UCNet
$S_\alpha \uparrow$	small	0,859	**0,864**	**0,871**	0,856
	medium	**0,902**	**0,902**	**0,903**	0,893
	large	**0,875**	**0,876**	0,867	0,862
	global	0,891	**0,892**	**0,894**	0,883
$F_\beta^{max} \uparrow$	small	0,790	**0,805**	**0,819**	0,794
	medium	0,908	**0,909**	**0,915**	0,904
	large	**0,935**	**0,933**	0,931	0,924
	global	0,886	**0,890**	**0,896**	0,882

Tab. 3 : Comparaison quantitative des différentes méthodes sur la base de données « Objectscale ». Vert et Rouge indiquent respectivement la meilleure et la seconde performance. ↑ indique que plus la valeur est grande mieux c'est.

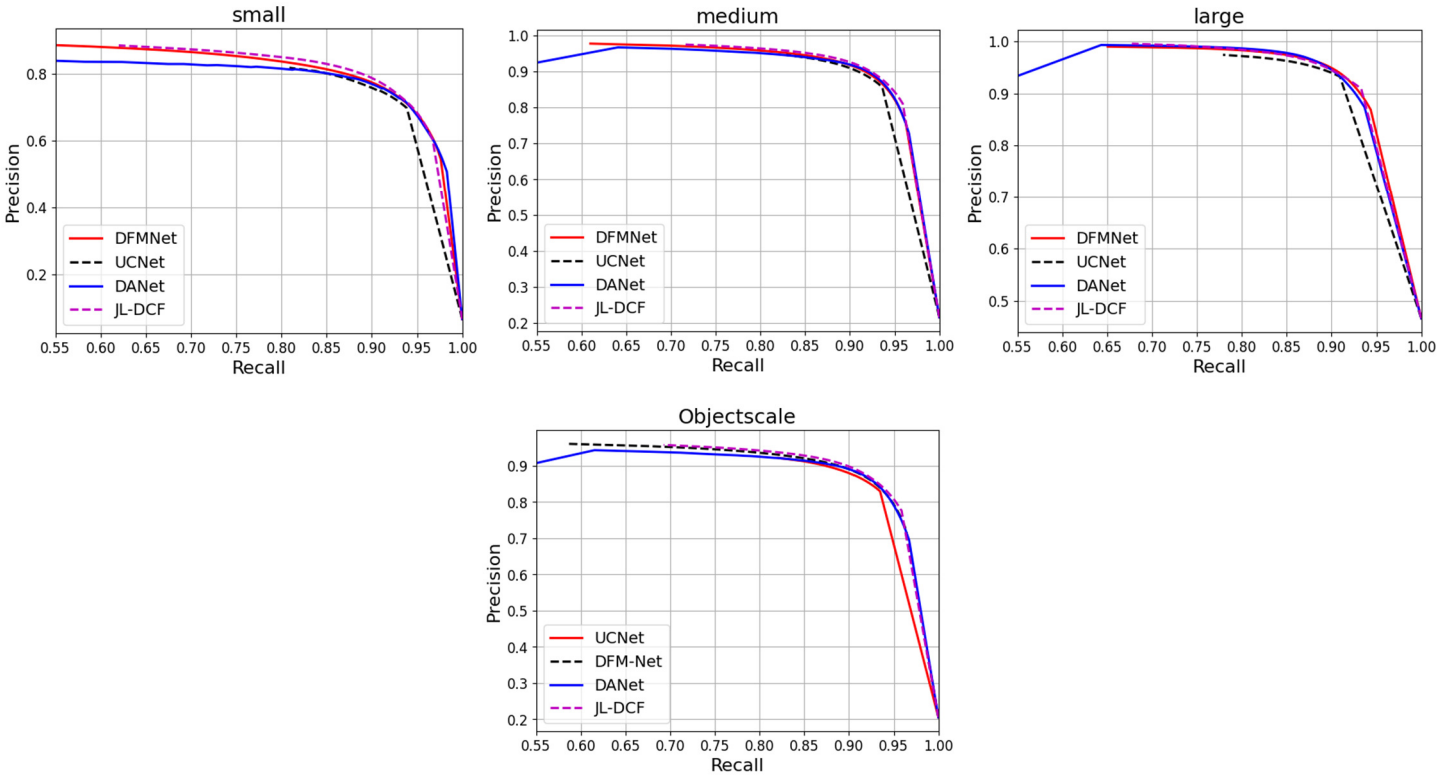

Fig. 19 : Courbes de Précision-Rappel sous différents seuils pour la base de données « Objectscale ». **Attention :** Pour le graphe avec le titre « Objectscale », les courbes ont des couleurs différentes.

La Tab. 3 ainsi que la Fig. 19 nous permettent de visualiser les performances des différentes méthodes. On remarque la méthode JL-DCF est très souvent plus performante que les autres méthodes. Ceci est en partie dû au fait que cette méthode utilise un backbone qui a été initialisé par des paramètres pré entrainés sur DSS [30] qui est une méthode de détection d'objets saillants par image RGB et qu'elle possède bien plus de paramètres que les autres méthodes. Néanmoins, on remarque que les méthodes avec attention restent les méthodes qui performent

le mieux après JL-DCF. De plus, pour les objets de grosse taille ces méthodes sont les plus performantes.

En mettant F_β^{max} qui représente l'accuracy de la méthode en fonction de S_α qui correspond au respect de la structure des objets saillants à détecter, il est plus facile de visualiser à quel point une méthode est performante par rapport aux autres :

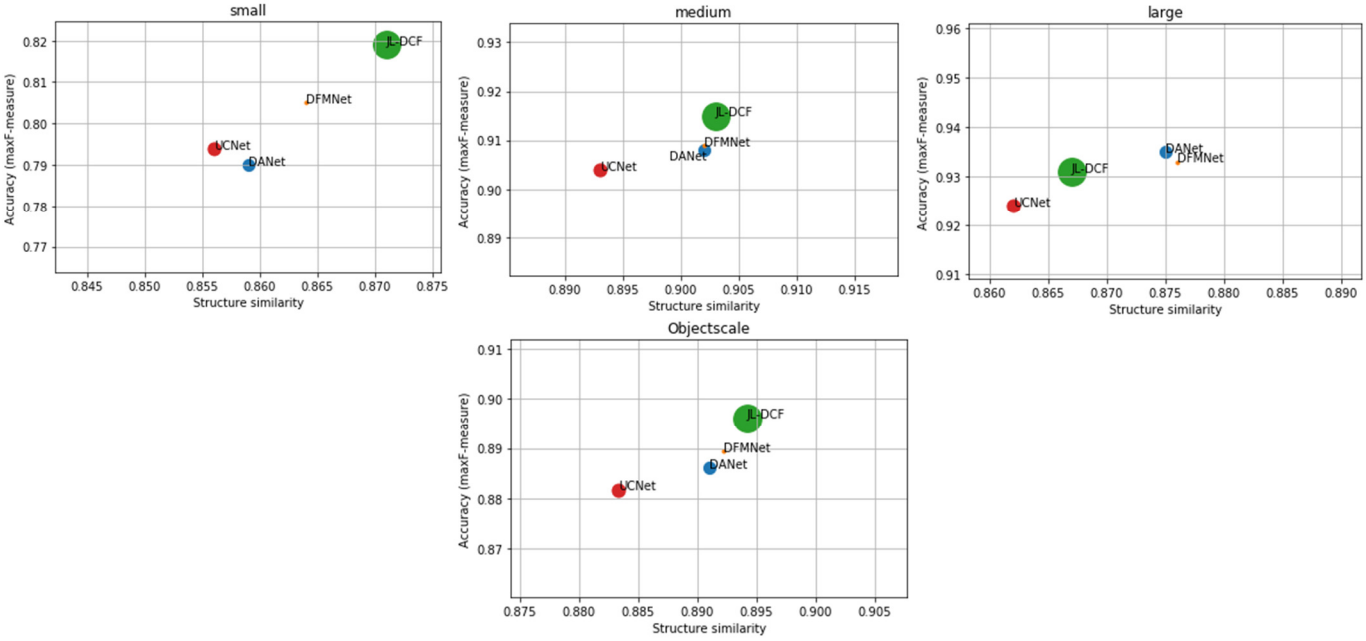

Fig. 20 : Comparaison entre F_β^{max} et S_α sur le dataset « Objectscale ». La grosseur des points correspond à la taille des poids des méthodes en Mbit. (Le petit point orange correspond à DFMNet)

La Fig.20 permet de mieux visualiser le fait que les méthodes d'attention sont plus légères, et qu'elles possèdent de bonnes performances. Néanmoins, on remarque que sur le sous dataset « small », la méthode DANet n'est pas aussi performante que son homologue DFMNet. Ceci est principalement dû au fait que DANet a un module d'attention qui discrimine le premier plan de l'arrière-plan ce qui peut être une tâche ardue pour les petits objets.

Finalement, les méthodes avec attention sont plutôt robustes aux différentes tailles d'objets et décrivent relativement bien la structure de ces objets.

Expérimentation sur la base de données « Multipleobjects »

Cette expérience a été conduite afin d'évaluer la capacité des différentes méthodes à identifier un ou plusieurs objets saillants. Pour obtenir les valeurs de la catégorie « global », l'expérience a été faite sur la base de données au complet sans faire la distinction entre « Single » et « Multi ». Les métriques considérées ont été énoncées précédemment. Les résultats obtenus sont les suivants :

	categorie	DANet	DFMNet	JL-DCF	UCNet
$S_\alpha \uparrow$	Multi	**0,841**	**0,831**	0,818	0,829
	Single	0,912	**0,919**	**0,921**	0,900
	global	**0,884**	**0,885**	0,881	0,872
$F_\beta^{max} \uparrow$	Multi	**0,876**	**0,866**	0,864	**0,866**
	Single	0,898	**0,907**	**0,913**	0,888
	global	**0,888**	**0,888**	**0,890**	0,877

Tab. 4 : Comparaison quantitative des différentes méthodes sur la base de données « Multipleobjects ». Vert et Rouge indiquent respectivement la meilleure et la seconde performance. ↑ indique que plus la valeur est grande mieux c'est.

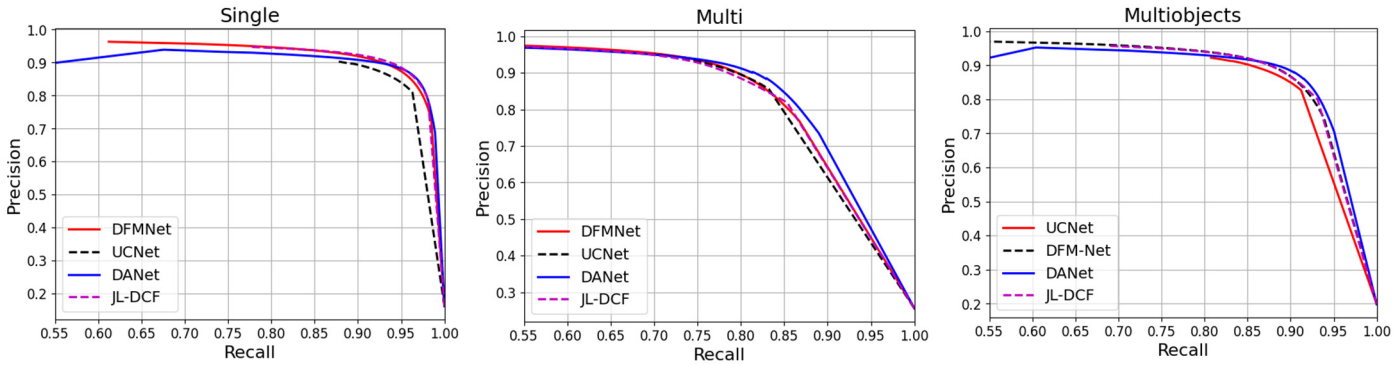

Fig. 21 : Courbes de Précision-Rappel sous différents seuils pour la base de données « Multipleobjects ».
Attention : Pour le graphe avec le titre « Multipleobjects », les courbes ont des couleurs différentes.

D'après les résultats présentés à la Tab. 4 ainsi que sur la Fig. 21, on peut voir que les méthodes avec attention semblent être plus performantes que les méthodes sans attention. En effet, sur la courbe de précision-rappel pour le dataset « Single » la classification des pixels appartenant à un objet saillant pour les méthodes DANet, DFMNet et JL-DCF est très bonne. Pour la courbe de précision-rappel sur le dataset « Multi », cette fois on peut voir que la méthode avec attention DANet performe mieux que les autres. Finalement la courbe de précision-rappel sur le dataset au global « Multipleobjects » nous montre que DANet reste supérieure aux autres méthodes suivies de très près par DFMNet et JL-DCF. La Tab. 4 montre également que ce sont les

méthodes avec attention qui ont en général la meilleure ainsi que la seconde meilleure performance pour le dataset « Multipleobjects ».

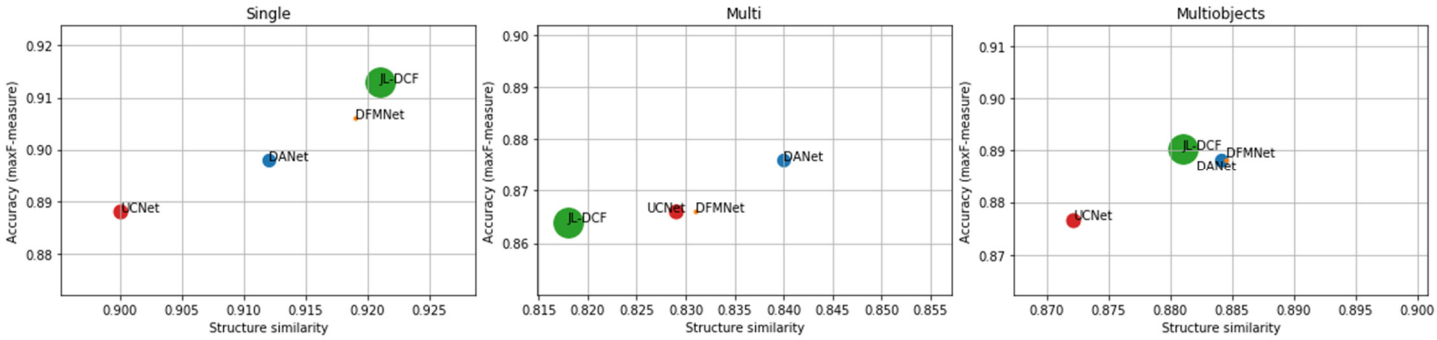

Fig. 22 : Comparaison entre F_β^{max} et S_α sur le dataset « Multipleobjects ». La grosseur des points correspond à la taille des poids des méthodes en Mbit. (Le petit point orange correspond à DFMNet)

Le graphe comparant la F_β^{max} et S_α sur le dataset « Multi » (Fig. 22) met en évidence le fait que les méthodes avec attention arrivent à mieux performer que les méthodes sans attention tout en gardant l'intégrité structurelle des objets saillants à détecter.

Finalement, le graphe sur le dataset global « Multiobjects » (Fig. 22) montre que les méthodes DANet et DFMNet ont une F_β^{max} légèrement inférieur à celle de JL-DCF, mais elles conservent mieux la structure des objets saillants à détecter, peu importe s'il y a un ou plusieurs objets.

Expérimentation sur la base de données « Illumination »

Cette expérience a été conduite afin d'évaluer la capacité des différentes méthodes à identifier des objets saillants dans des configurations de luminosité différentes. Pour obtenir les valeurs de la catégorie « global », l'expérience a été réalisée sur la base de données au complet sans faire la distinction entre « Low » et « High ». Les métriques considérées ont été énoncées précédemment. Les résultats obtenus sont les suivants :

	catégorie	DANet	DFMNet	JL-DCF	UCNet
$S_\alpha \uparrow$	High	**0,888**	**0,888**	**0,881**	0,876
	Low	0,859	**0,861**	**0,872**	0,838
	global	**0,883**	**0,883**	**0,879**	0,869
$F_\beta^{max} \uparrow$	High	**0,888**	**0,890**	**0,888**	0,878
	Low	**0,881**	0,873	**0,892**	0,849
	global	0,886	**0,887**	**0,889**	0,872

Tab. 5 : Comparaison quantitative des différentes méthodes sur la base de données « Illumination ». Vert et Rouge indiquent respectivement la meilleure et la seconde performance. ↑ indique que plus la valeur est grande mieux c'est.

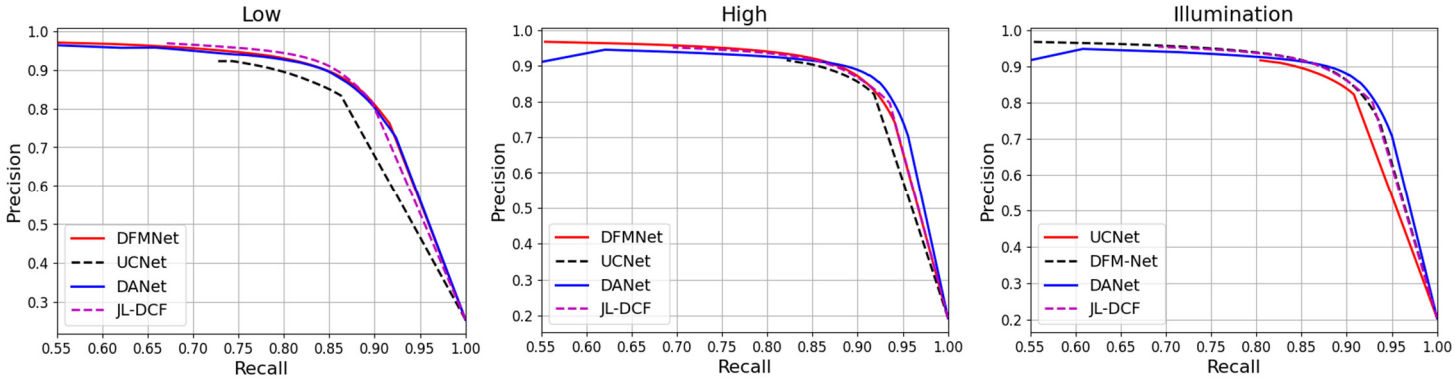

Fig. 23 : Courbes de Précision-Rappel sous différents seuils pour la base de données « Illumination ». **Attention :** Pour le graphe avec le titre « Illumination », les courbes ont des couleurs différentes.

Grâce à la Tab. 5 ainsi qu'à la Fig. 23, on peut voir que les méthodes avec attention semblent être plus performantes que les méthodes sans attention. En effet, sur la courbe de précision-rappel pour le dataset « High » la classification des pixels appartenant à un objet saillant pour la méthode DANet est supérieure à celle des autres méthodes, suivi de près par DFMNet. La courbe de précision-rappel sur le dataset « Low » ne permet pas de mettre évidence si DANet, DFMNet ou JL-DCF est plus performant. Finalement la courbe de précision-rappel sur le dataset au global « Illumination » nous montre que DANet reste supérieure aux autres méthodes suivies de très près par DFMNet et JL-DCF. La Tab. 5 nous permet également de mettre en évidence le fait que ce sont les méthodes avec attention qui ont en général la meilleure ainsi que la seconde meilleure performance pour le dataset « Illumination ».

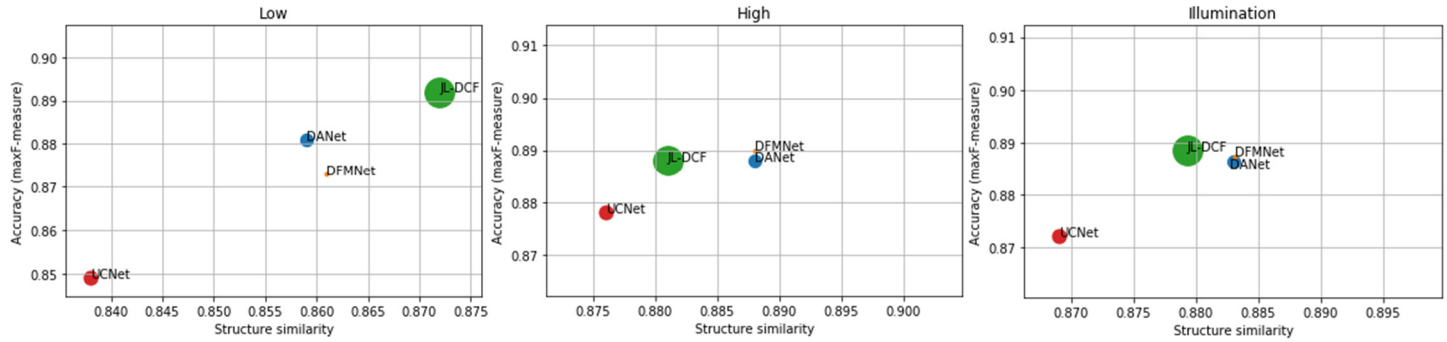

Fig. 24 : Comparaison entre F_β^{max} et S_α sur le dataset « Illumination ». La grosseur des points correspond à la taille des poids des méthodes en Mbit. (Le petit point orange correspond à DFMNet)

En observant la Fig . 24, on remarque que pour un scénario où la luminosité est correct, les méthodes avec attention seront toujours plus robustes que les méthodes sans attention pour préserver la structure des objets saillants à détecter. La F_β^{max} est sensiblement la même entre JL-DCF, DANet et DFMNet. Lorsque la luminosité est plus faible, c'est JL-DCF qui reste la meilleure méthode, mais les méthodes avec attention arrivent à conserver un $S_\alpha > 0.85$.

Globalement pour des scénarios où la luminosité varie, DANet et DFMNet ont une F_β^{max} légèrement inférieur à celle de JL-DCF, mais elles conservent mieux la structure des objets saillants à détecter.

Expérimentation sur la base de données « Complexbackground »

Cette expérience a été conduite afin d'évaluer la capacité des différentes méthodes à identifier des objets saillants dans des scénarios où l'encombrement de l'arrière-plan de l'image est variant. Pour obtenir les valeurs de la catégorie « global », l'expérience a été réalisée sur la base de données au complet sans faire la distinction entre « simple », « uncertain » et « complex ». Les métriques considérées ont été énoncées précédemment. Les résultats obtenus sont les suivants :

	catégorie	DANet	DFMNet	JL-DCF	UCNet
$S_\alpha \uparrow$	simple	0,960	**0,967**	**0,969**	0,957
	uncertain	**0,905**	**0,905**	**0,913**	0,898
	complex	**0,821**	0,819	**0,826**	**0,826**
	global	**0,901**	**0,901**	**0,908**	0,895
$F_\beta^{max} \uparrow$	simple	0,970	**0,979**	**0,981**	0,972
	uncertain	0,899	**0,902**	**0,911**	0,895
	complex	0,757	0,760	**0,777**	**0,774**
	global	0,891	**0,893**	**0,903**	0,889

Tab. 6 : Comparaison quantitative des différentes méthodes sur la base de données « Complexbackground ». Vert et Rouge indiquent respectivement la meilleure et la seconde performance. ↑ indique que plus la valeur est grande mieux c'est.

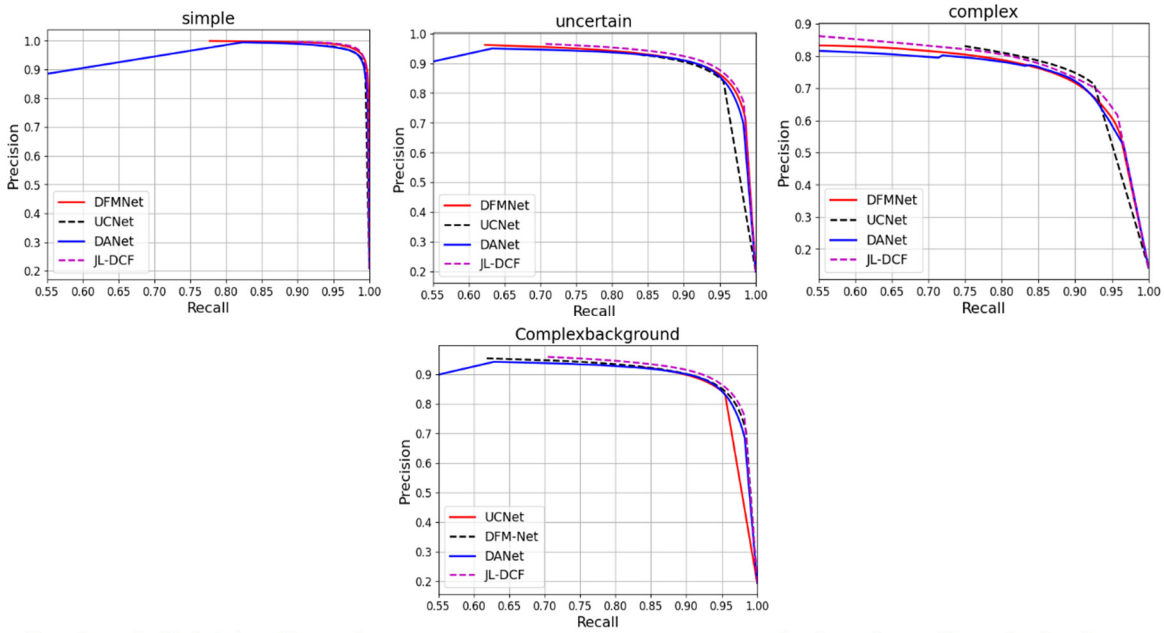

Fig. 25 : Courbes de Précision-Rappel sous différents seuils pour la base de données « Complexbackground ».
Attention : Pour le graphe avec le titre « Complexbackground », les courbes ont des couleurs différentes.

Les résultats de la Fig. 25 montrent que pour des scènes où l'arrière-plan est « simple », toutes les méthodes classifient presque parfaitement les pixels appartenant à un objet saillant. Pour les scènes où l'arrière-plan est considéré comme « complex », on remarque que les méthodes sans attention classifient mieux les pixels appartenant aux objets saillants contrairement aux méthodes avec attention. De plus, avec les résultats de la Tab. 6, on remarque que les méthodes avec attention possèdent généralement les secondes meilleures performances sauf pour le dataset « complex ».

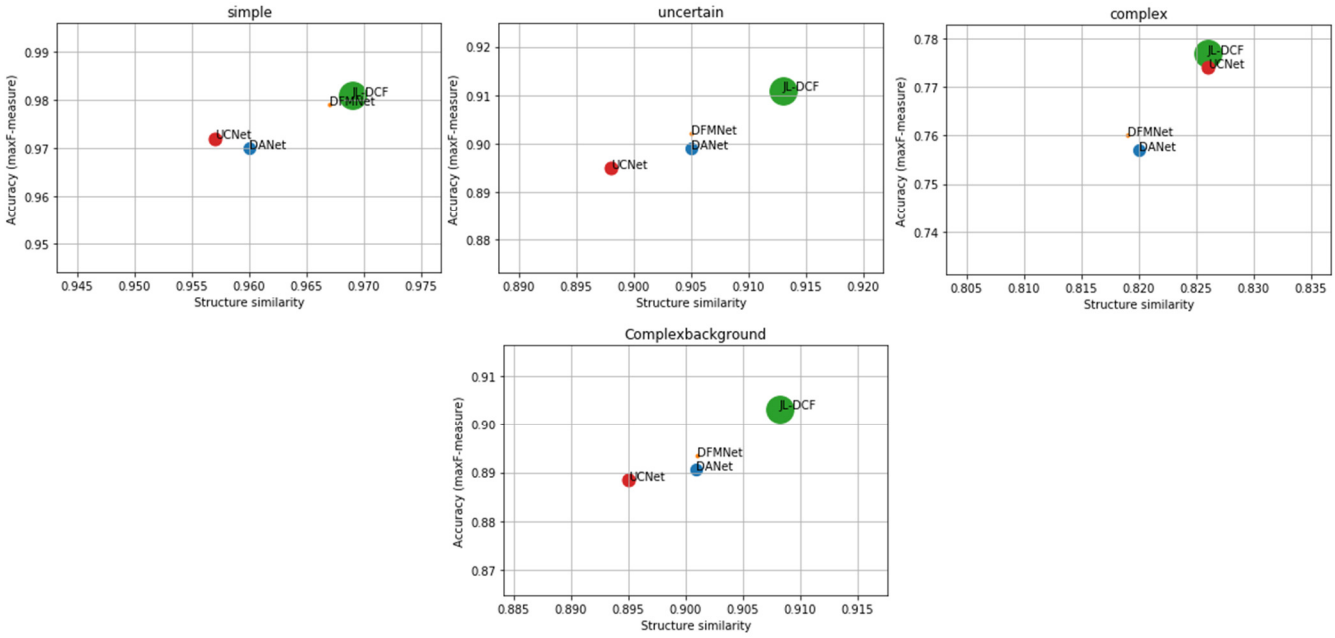

Fig. 26 : Comparaison entre F_β^{max} et S_α sur le dataset « Complexbackground ». La grosseur des points correspond à la taille des poids des méthodes en Mbit. (Le petit point orange correspond à DFMNet)

Finalement, la Fig. 26 nous permet de mieux visualiser le fait que les méthodes avec attention sont peu performantes pour les images où l'arrière-plan est « complex » comparé aux méthodes sans attention. De plus dans la globalité du dataset « Complexbackground », mis à part le fait que DFMNet et DANet sont plus légères il ne semble pas y avoir plus d'avantages que cela d'utiliser une méthode avec attention, car la différence entre le S_α de UCNet et de DANet est relativement faible.

Analyse qualitative

Dans cette partie, quelques cartes de saillances prédites par les différentes méthodes seront présentées dans la Fig. 27. La première ligne correspond à un objet de petite taille, la seconde ligne correspond à un objet de taille moyenne, la troisième ligne correspond à un objet de grande

taille, la quatrième ligne correspond à une image avec plusieurs objets saillants, la cinquième ligne correspond à un objet dans un environnement lumineux, la sixième ligne correspond à un objet dans un environnement à faible luminosité, la septième ligne correspond à une image où l'arrière-plan est simple et la huitième ligne correspond à une image où l'arrière-plan est complexe :

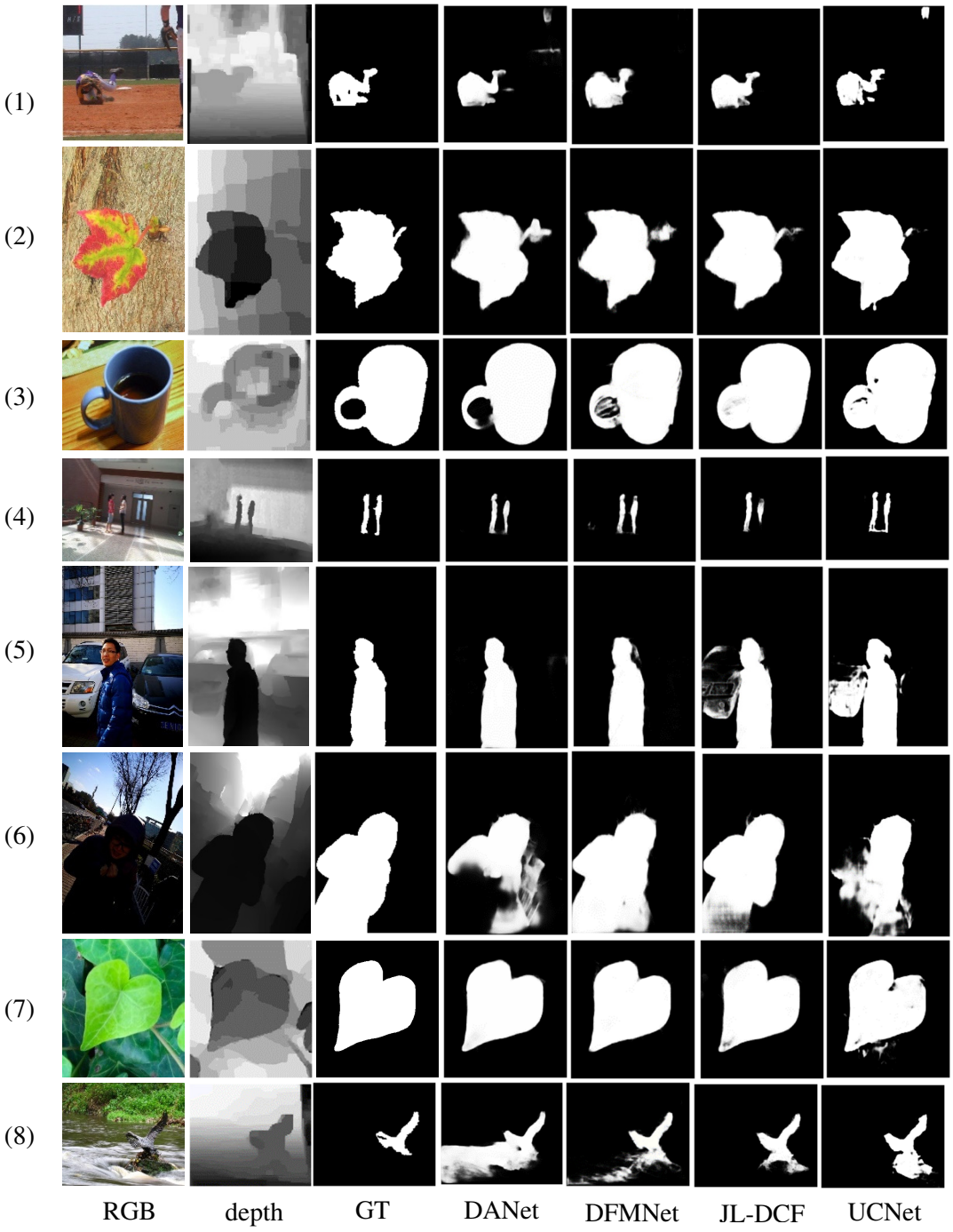

Fig. 27 : Comparaison qualitative des prédictions des différentes méthodes de détection d'objets saillants par image RGB-D sur des datasets qui propose des scénarios challengeant.

Pour la ligne 1, on remarque que la plupart des méthodes arrivent à détecter l'objet de petite taille. Toutefois, on voit également que les méthodes DANet et UCNet présentent quelques petits artefacts en haut à droite de l'image. Pour les lignes 2 et 7, les méthodes arrivent globalement à détecter les objets de taille moyenne et ceux qui sont sur un arrière-plan simple. Néanmoins, on remarque que pour les objets de taille moyenne les méthodes avec attention ont tendance à détecter une plus grosse partie de la tige.

La ligne 3 met évidence que les méthodes avec attention ont plus de facilité à bien définir les objets de taille large. Dans une image, il peut y avoir plusieurs objets saillants à détecter, la ligne 4 montre que les méthodes sans attention ont tendance à détecter qu'une partie de l'objet saillant (JL-DCF) ou encore à prendre en compte des parties de l'image qui ne font pas partie de l'objet à détecter (UCNet) tandis que les méthodes avec attention arrivent à détecter tout l'objet saillant. La luminosité peut également varier sur une image, la ligne 5 montre que lorsque la scène est soumise à un éclairage plus ou moins fort, les méthodes sans attention ont tendance à détecter les objets environnants qui sont affectés par cet éclairage. À contrario, les méthodes avec attention détectent finement l'objet saillant.

En observant les cartes de caractéristiques des branches RGB et de profondeur de JL-DCF (Joint Learning), on peut voir que la lumière qui tape sur la voiture blanche induit le modèle en erreur.

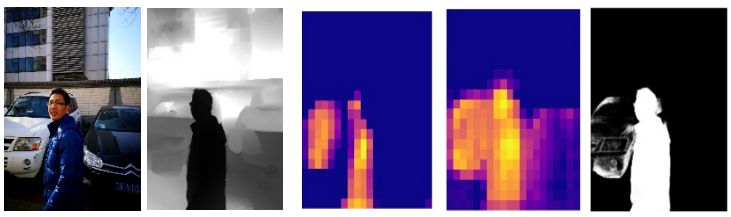

Fig. 28 : Carte de caractéristique pour la branche rgb et de profondeur générée par la partie Joint Learning de la méthode JL-DCF.

Similairement à la ligne 5, pour la ligne 6 on observe une scène où la luminosité est faible. On remarque que les méthodes DANet et UCNet n'arrivent pas à conserver la structure de l'objet saillant à détecter contrairement à JL-DCF et DFMNet. Pour UCNet, ce manque de robustesse est dû au caractère stochastique de la méthode.

Dans un scénario où l'arrière-plan de l'image est complexe (ligne 8), on remarque que les méthodes d'attention ont plus de mal à dissocier le premier plan de l'arrière-plan pour guider la détection de l'objet saillant.

Investiguons les méthodes DFMNet et DANet pour comprendre pourquoi elles font de moins bonnes prédictions que les méthodes JL-DCF et UCNet sur la ligne 8 de la Fig. 27 :

Pour DFM-Net, on va extraire les 5 cartes d'attention qui ont pour but d'identifier les zones saillantes :

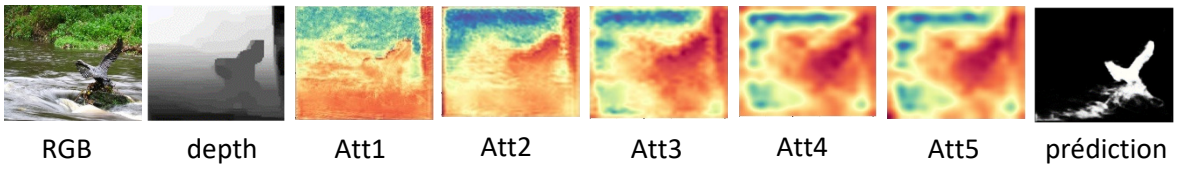

Fig. 29 : Carte d'attention générée par le module d'attention DHA de la méthode DFM-Net.

La couleur rouge sur les cartes d'attention correspond à la présence potentielle d'objets saillants. Les cartes d'attention de 1 à 3 sont associées aux caractéristiques de bas niveau et les cartes d'attention de 4 et 5 correspondent aux caractéristiques de haut niveau. Sur la Fig. 29, on remarque que la carte d'attention 5 conserve pas mal de zones n'appartenant pas à l'objet saillant : Bas, gauche et droite de l'image. Les zones gauche et droite n'ont pas été prises en compte lors de la prédiction grâce aux coefficients générés par le module DQW qui permet de déterminer l'importance des caractéristiques de profondeur. Il semblerait donc que le manque de contraste sur la carte de profondeur ait induit en erreur le module d'attention DHA dans son identification des zones saillantes.

Pour DANet, nous allons d'abord observer les cartes de caractéristiques en sortie de l'encodeur puis nous observerons les cartes d'attention pour le premier plan et pour l'arrière-plan de l'image. Rappelons que pour cette méthode, l'image RGB et la carte de profondeur sont fusionnées pour former une image à 4 canaux (Early « input » fusion).

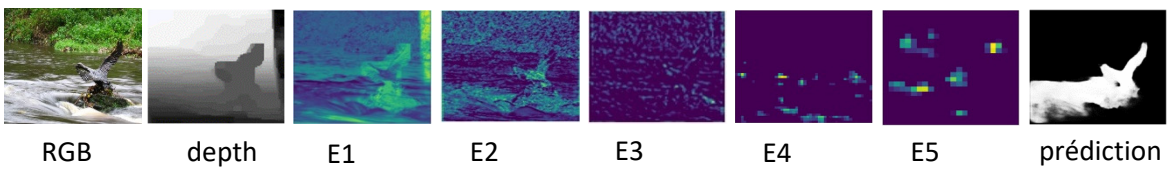

Fig. 30 : Cartes de caractéristiques générées par l'encodeur de la méthode DANet. Les différents blocs de l'encodeur sont identifiés par E1 – E5.

Dès le premier bloc E1, on remarque que la distinction entre ce qui appartient au premier plan et à l'arrière-plan n'est pas évidente. Dans les plus hauts niveaux E5, peu d'informations sont extraites pour identifier l'objet saillant. Ce manque d'information a donc impacté le module d'attention permettant de séparer le premier plan de l'arrière-plan.

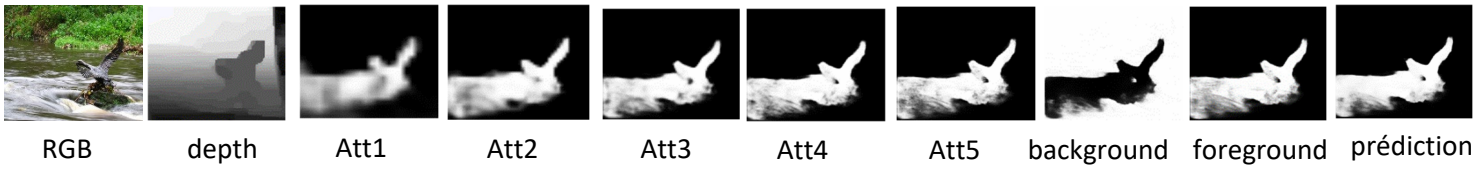

Fig. 31 : Carte d'attention générée par le module d'attention DEDA de la méthode DANet. (+ carte d'attention du premier et de l'arrière-plan).

Finalement, si on observe la Fig. 30 et la Fig. 31 on comprend que c'est les caractéristiques de E1 et E2 qui ont induit le modèle en erreur et le peu d'information dans les blocs suivants n'a pas permis de mieux séparer le premier plan de l'arrière-plan ce qui mené à une prédiction bruitée. À noter que la qualité moyenne de la carte de profondeur a eu un effet certain sur cette prédiction.

5. Conclusion

Dans ce projet, nous avons utilisé différentes méthodes de détection d'objets saillants par image RGB-D. Quatre méthodes ont été évaluées expérimentalement : deux méthodes avec un module d'attention (DANet [21], DFM-Net [22]) et deux méthodes sans module d'attention (UCNet [15], JL-DCF [13]). Nous avons utilisé six bases de données de la littérature : LFSD, DES, NLPR, NJU2K, SIP et STERE pour créer quatre bases de données qui simulent différents scénarios challengeant : Object scale, Multiple objects, Illumination, Complex background.

Les résultats obtenus peuvent être décomposés en deux parties. Dans un premier temps, il y a les résultats concernant l'efficacité des différentes méthodes. À l'issue de nos expérimentations, on a mis en évidence le fait que les méthodes avec attention prennent moins d'espace mémoire comparé aux méthodes sans attention. De plus, la vitesse d'inférence des méthodes avec attention est supérieure à celle des méthodes sans attention. Nous pouvons donc en conclure que pour des applications nécessitant des prédictions en temps réel les méthodes avec attentions sont plus adaptées.

Ensuite il y a les résultats concernant les performances des différentes méthodes. À l'issue des expérimentations, on a pu calculer les courbes de précision-rappel qui permettent de mettre en évidence la capacité des méthodes à bien classifier les pixels appartenant à l'objet saillant, la F_β^{max} qui définit l'accuracy des méthodes et la S_α qui défini la capacité des méthodes à conserver la structure des objets saillant à détecter qui nous ont permis de mettre en évidence le fait que la méthode JL-DCF était la méthode la plus performante, mais suivie de très près par la méthode DFM-Net. De plus, nous avons mis en évidence que les méthodes avec attention ont la capacité de conserver de façon efficace l'intégrité structurelle des objets à détecter, peu importe les scénarios challengeant, ces méthodes ont également démontré des performances supérieures à la méthode JL-DCF pour les datasets composé d'objets de grande taille, de scène bien éclairées et de plusieurs objets à détecter. Malheureusement, les méthodes avec attention restent peu performantes pour les images qui ont des arrière-plans complexes. Toutefois, globalement les méthodes d'attention permettent d'obtenir de bonne performance tout en étant efficace. Le code python qui a été utilisé pour nos expérimentations est disponible sur le GitHub suivant : https://github.com/y0un0/SYS843_EtudeExp_RGBD_SOD

Pour un futur projet, il serait intéressant d'utiliser un backbone plus dense tel que ResNet pour la méthode DFM-Net afin de voir si ses performances peuvent surpasser celles de JL-DCF dans les différents datasets défini précédemment. Il serait également intéressant de faire un dataset de test avec différent niveau de qualité de la carte de profondeur afin de voir la robustesse des méthodes face à des cartes de profondeur de mauvaise qualité. De plus, il pourrait être intéressant d'implémenter un module d'attention sur la méthode JL-DCF afin de voir si on peut le rendre plus efficace et plus performant.

Références

[1] Wang, N., & Gong, X. (2019). Adaptive Fusion for RGB-D Salient Object Detection. Repéré à http://arxiv.org/abs/1901.01369

[2] Liu, Z., Shi, S., Duan, Q., Zhang, W., & Zhao, P. (2019). Salient object detection for RGB-D image by single stream recurrent convolution neural network. *Neurocomputing*, *363*, 46-57. https://doi.org/10.1016/j.neucom.2019.07.012

[3] Qu, L., He, S., Zhang, J., Tian, J., Tang, Y., & Yang, Q. (2016). RGBD Salient Object Detection via Deep Fusion. https://doi.org/10.1109/TIP.2017.2682981

[4] Achanta, R., Shaji, A., Smith, K., Lucchi, A., Fua, P., & Süsstrunk, S. (2012). SLIC Superpixels Compared to State-of-the-Art Superpixel Methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *34*(11), 2274-2282. https://doi.org/10.1109/TPAMI.2012.120

[5] Han, J., Chen, H., Liu, N., Yan, C., & Li, X. (2018). CNNs-Based RGB-D Saliency Detection via Cross-View Transfer and Multiview Fusion. *IEEE Transactions on Cybernetics*, *48*(11), 3171-3183. https://doi.org/10.1109/TCYB.2017.2761775

[6] Chen, H., Li, Y., & Su, D. (2019). Multi-modal fusion network with multi-scale multi-path and cross-modal interactions for RGB-D salient object detection. *Pattern Recognition*, *86*. https://doi.org/10.1016/j.patcog.2018.08.007

[7] Fan, D.-P., Cheng, M.-M., Liu, Y., Li, T., & Borji, A. (2017). Structure-measure: A New Way to Evaluate Foreground Maps. Repéré à http://arxiv.org/abs/1708.00786

[8] Zhao, J.-X., Cao, Y., Fan, D.-P., Cheng, M.-M., Li, X.-Y., & Zhang, L. (2019). Contrast Prior and Fluid Pyramid Integration for RGBD Salient Object Detection. Dans *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 3922-3931). https://doi.org/10.1109/CVPR.2019.00405

[9] Zhou, X., Li, G., Gong, C., Liu, Z., & Zhang, J. (2020). Attention-guided RGBD saliency detection using appearance information. *Image and Vision Computing*, *95*. https://doi.org/10.1016/j.imavis.2020.103888

[10] Zhang, Z., Lin, Z., Xu, J., Jin, W., Lu, S.-P., & Fan, D.-P. (2020). Bilateral Attention Network for RGB-D Salient Object Detection. https://doi.org/10.1109/tip.2021.3049959

[11] Ding, Y., Liu, Z., Huang, M., Shi, R., & Wang, X. (2019). Depth-aware saliency detection using convolutional neural networks. *Journal of Visual Communication and Image Representation*, *61*. https://doi.org/10.1016/j.jvcir.2019.03.019

[12] Li, G., Liu, Z., & Ling, H. (2020). ICNet: Information Conversion Network for RGB-D Based Salient Object Detection. *IEEE Transactions on Image Processing*, *29*, 4873-4884. https://doi.org/10.1109/TIP.2020.2976689

[13] Fu, K., Fan, D.-P., Ji, G.-P., & Zhao, Q. (2020). JL-DCF: Joint Learning and Densely-Cooperative Fusion Framework for RGB-D Salient Object Detection. Repéré à https://arxiv.org/abs/2004.08515

[14] Huang, G., Liu, Z., van der Maaten, L., & Weinberger, K. Q. (2016). Densely Connected Convolutional Networks. Repéré à https://arxiv.org/abs/1608.06993

[15] Zhang, J., Fan, D.-P., Dai, Y., Anwar, S., Saleh, F. S., Zhang, T., & Barnes, N. (2020). UC-Net: Uncertainty Inspired RGB-D Saliency Detection via Conditional Variational Autoencoders. Dans *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 8579-8588). https://doi.org/10.1109/CVPR42600.2020.00861

[16] Qin, X., Zhang, Z., Huang, C., Gao, C., Dehghan, M., & Jagersand, M. (2019). BASNet: Boundary-Aware Salient Object Detection. Dans *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 7471-7481). https://doi.org/10.1109/CVPR.2019.00766

[17] Simonyan, K., & Zisserman, A. (2014). Very Deep Convolutional Networks for Large-Scale Image Recognition. Repéré à http://arxiv.org/abs/1409.1556

[18] Yang, M., Yu, K., Zhang, C., Li, Z., & Yang, K. (2018). DenseASPP for Semantic Segmentation in Street Scenes. Dans *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 3684-3692). https://doi.org/10.1109/CVPR.2018.00388

[19] Luo, Z., Mishra, A., Achkar, A., Eichel, J., Li, S., & Jodoin, P.-M. (2017). Non-local Deep Features for Salient Object Detection. Dans *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 6593-6601). https://doi.org/10.1109/CVPR.2017.698

[20] Lin, T.-Y., Dollár, P., Girshick, R., He, K., Hariharan, B., & Belongie, S. (2016). Feature Pyramid Networks for Object Detection. Repéré à https://arxiv.org/abs/1612.03144

[21] Zhao, X., Zhang, L., Pang, Y., Lu, H., & Zhang, L. (2020). A Single Stream Network for Robust and Real-time RGB-D Salient Object Detection. Repéré à http://arxiv.org/abs/2007.06811

[22] Zhang, W., Ji, G.-P., Wang, Z., Fu, K., & Zhao, Q. (2021). Depth Quality-Inspired Feature Manipulation for Efficient RGB-D Salient Object Detection. Repéré à https://arxiv.org/abs/2107.01779

[23] Milletari, F., Navab, N., & Ahmadi, S.-A. (2016). V-Net: Fully Convolutional Neural Networks for Volumetric Medical Image Segmentation. Repéré à https://arxiv.org/abs/1606.04797

[24] Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., & Chen, L.-C. (2018). MobileNetV2: Inverted Residuals and Linear Bottlenecks. Repéré à https://arxiv.org/abs/1801.04381

[25] Zhao, H., Shi, J., Qi, X., Wang, X., & Jia, J. (2016). Pyramid Scene Parsing Network. Repéré à http://arxiv.org/abs/1612.01105

[26] Achanta, R., Hemami, S., Estrada, F., & Susstrunk, S. (2009). Frequency-tuned salient region detection. Dans *2009 IEEE Conference on Computer Vision and Pattern Recognition* (pp. 1597-1604). https://doi.org/10.1109/CVPR.2009.5206596

[27] Fan, D.-P., Cheng, M.-M., Liu, Y., Li, T., & Borji, A. (2017). Structure-measure: A New Way to Evaluate Foreground Maps. Repéré à http://arxiv.org/abs/1708.00786

[28] Fan, D.-P., Gong, C., Cao, Y., Ren, B., Cheng, M.-M., & Borji, A. (2018). Enhanced-alignment Measure for Binary Foreground Map Evaluation. Repéré à http://arxiv.org/abs/1805.10421

[29] Zhou, T., Fan, D.-P., Cheng, M.-M., Shen, J., & Shao, L. (2020). RGB-D Salient Object Detection: A Survey. Repéré à https://arxiv.org/abs/2008.00230

[30] Hou, Q., Cheng, M. M., Hu, X., Borji, A., Tu, Z., & Torr, P. H. S. (2019). Deeply Supervised Salient Object Detection with Short Connections. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *41*(4), 815-828. https://doi.org/10.1109/TPAMI.2018.2815688